

Software y estándares para la Web

P4. COMPUTACIÓN EN EL SERVIDOR WEB

VERSIÓN 1.0 24/NOVIEMBRE/2021

Contenido

Objetivos	2
Ejercicio 1	2
Guía para resolver el ejercicio 1	3
Ejercicio 2	3
Guía para resolver el ejercicio 2	4
Ejercicio 3	5
Guía para resolver el ejercicio 3	5
Ejercicio 4	9
Guía para resolver el ejercicio 4	9
Ejercicio 5	10
Guía para resolver el ejercicio 5	11
Ejercicio 6	11
Guía para resolver el ejercicio 6	13
Ejercicio 7	13
Guía para resolver el ejercicio 7	14
Criterios de evaluación	14
Entrega de la práctica	15

RECUERDA: Revisar los criterios de evaluación y la forma de entrega de la práctica

Objetivos

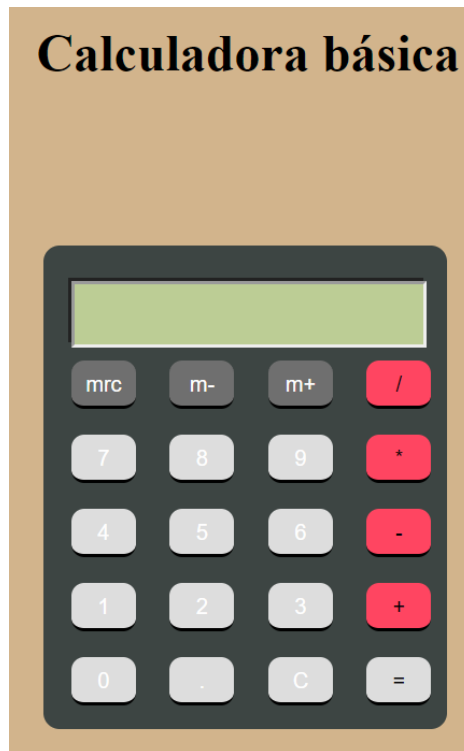
En esta práctica el objetivo es hacer computación en el servidor Web usando el lenguaje PHP.

Esta práctica se corresponde con los temas de teoría:

- Computación en el Servidor Web
- El lenguaje PHP
- PHP: Archivos y MySQL

Ejercicio 1

Tarea 1. Escribir un archivo en PHP denominado **CalculadoraBasica** (con extensión **.php**) que referencie a un archivo CSS denominado **CalculadoraBasica** (con extensión **.css**). El archivo en HTML5 deberá usar la metáfora de una calculadora “física” para contener la interfaz web de una calculadora básica. La interfaz puede ser de la forma siguiente:



Tarea 2. Escribir un archivo CSS denominado **CalculadoraBasica** (con extensión **.css**) con la hoja de estilo de la interfaz web de la calculadora básica.

Tarea 3. El archivo PHP debe contener:

- Una clase denominada **CalculadoraBasica** con los atributos y métodos necesarios para realizar las operaciones de la calculadora.
- Se deberá usar la función **eval()** para evaluar las expresiones de la pantalla de la calculadora
- Se deberá **proteger la pantalla de la calculadora** para que no sea editable, evitando así la posible inyección de código PHP a través de **eval()**.

Epílogo. Todas las tareas se presentan en la misma sub-carpeta denominada **Ejercicio-1** que debe contener como mínimo, los archivos:

- CalculadoraBasica.php
- CalculadoraBasica.css

Guía para resolver el ejercicio 1

El objetivo es crear objetos a partir de clases en PHP. Para ver como se crea una clase en PHP puede consultarse el ejercicio:

<http://di002.edv.uniovi.es/~cueva/php/015-Clases.php>

Para entender la forma en la que se usan objetos y clases en PHP de forma interactiva consultar:

<http://di002.edv.uniovi.es/~cueva/php/025-CalculadoraDiabetica.php>

Para entender el uso de eval() y manejo de excepciones en PHP consultar:

<http://di002.edv.uniovi.es/~cueva/php/022-Calculadora.php>

Para entender el uso de botones como teclas consultar:

<http://di002.edv.uniovi.es/~cueva/php/026-ClaseBotones.php>

<http://di002.edv.uniovi.es/~cueva/php/027-Botonera.php>

Para que la calculadora recuerde las teclas pulsadas utilizar SESSION, por ejemplo:

<http://di002.edv.uniovi.es/~cueva/php/023-Contador.php>

<http://di002.edv.uniovi.es/~cueva/php/028-RegistroAcontecimientos.php>

El uso de eval() es una solución simple para la evaluación de expresiones algebraicas, aunque puede **comprometer la seguridad**, si se deja una puerta abierta a la ejecución de código malicioso. Por eso es necesario proteger la pantalla de la calculadora con “**disabled**”, para que los usuarios obligatoriamente introduzcan las expresiones a evaluar por medio de pulsación de teclas de la calculadora y no escribiendo directamente en la pantalla.

Debe usarse obligatoriamente manejo de excepciones.

Si no se usase eval() se nos complicarían un poco el ejercicio y necesitaríamos definir una gramática, para usar técnicas de procesamiento de lenguajes de programación, que estudiareis en la asignatura "Diseño de lenguajes de programación".

Una breve introducción a las técnicas de procesamiento de lenguajes puede consultarse en:

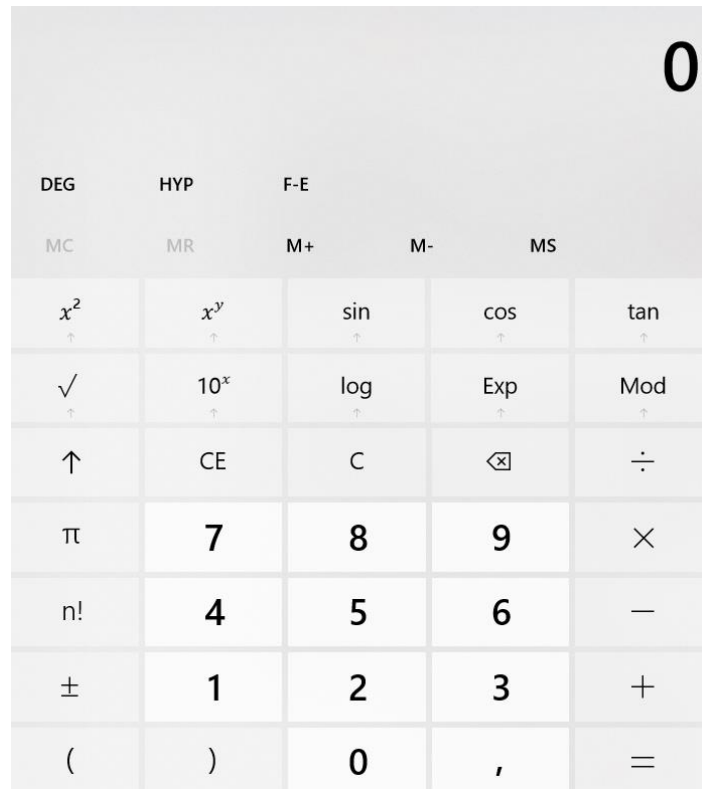
http://di002.edv.uniovi.es/~cueva/publicaciones/monografias/10_Conceptos_Basicos_Procesadores_Lenguaje.pdf

Ejercicio 2

Tarea 1. Escribir un archivo en PHP denominado **CalculadoraCientífica** (con extensión **.php**) que referencie a un archivo CSS denominado **CalculadoraCientífica** (con extensión **.css**).

Debe utilizarse la herencia de la clase calculadora básica del ejercicio anterior.

El archivo en PHP deberá usar la metáfora de una calculadora para contener la interfaz web de una calculadora científica. La interfaz puede emular a la calculadora científica de Windows 10. No es obligatorio implementar todas las opciones de la calculadora de Windows 10.



Tarea 2. Escribir un archivo CSS denominado **CalculadoraCientífica** (con extensión **.css**) con la hoja de estilo de la interfaz web de la calculadora científica.

Tarea 3. El código PHP debe contener:

- Una clase denominada **CalculadoraCientífica** que hereda de la clase **CalculadoraBásica** con los atributos y métodos necesarios para realizar las operaciones de la calculadora científica.
- Se deberá usar la función **eval()** para evaluar las expresiones de la pantalla de la calculadora
- Se deberá **proteger la pantalla de la calculadora** para que no sea editable, evitando así la posible inyección de código a través de eval()).

Epílogo. Todas las tareas se presentan en la misma sub-carpeta denominada **Ejercicio-2** que debe contener como mínimo, los archivos:

- CalculadoraCientifica.php
- CalculadoraCientifica.css

Guía para resolver el ejercicio 2

Para el manejo de funciones matemáticas en PHP se puede consultar el manual de PHP:

<https://www.php.net/manual/es/ref.math.php>

Se debe utilizar la herencia de clases. Se puede crear una clase **CalculadoraCientífica** que hereda de la clase **CalculadoraBásica**.

Sobre el uso de la herencia consultar el manual de PHP y el ejercicio siguiente:

<https://www.php.net/manual/es/language.oop5.php>

<http://di002.edv.uniovi.es/~cris/PHP/13-Herencia.php>

Ejercicio 3

Tarea 1. Escribir un archivo en PHP denominado **CalculadoraRPN** (con extensión. php) que referencie a un archivo CSS denominado **CalculadoraRPN** (con extensión .css).

Las calculadoras RPN:

- Utilizan la notación postfija o notación polaca inversa (*reverse polish notation*).
- Evaluará las expresiones utilizando una pila LIFO.
- No tienen el botón igual.

No se permite usar la función eval(), por su peligrosidad potencial y por no ser necesaria.

La calculadora debe tener además de las funciones básicas (+,-,*,/), funciones científicas (sin, cos, tan,...). Se deja libre al estudiante el diseño de las funciones incorporadas.

Epílogo. Todas las tareas se presentan en la misma sub-carpeta denominada **Ejercicio-3** que debe contener como mínimo, los archivos:

- CalculadoraRPN.php
- CalculadoraRPN.css

Guía para resolver el ejercicio 3

En este ejercicio **no se puede usar eval()**, que puede dar problemas de seguridad

Una calculadora RPN **no tiene la tecla “igual”**, pero si tiene una tecla **“Enter”**..

En la pantalla de una calculadora RPN se muestra los valores que hay en la pila LIFO (Last In First Out, último en entrar primero en salir). A continuación, se muestra la pantalla de una calculadora RPN:

5:	27,50
4:	0,50
3:	15,00
2:	32,30
1:	9,55

En este caso se han introducido los valores de una lista de la compra con 5 artículos de precios: 27,5 €, 0,50 €, 15,0 €, 32,3 € y 9,55€

A continuación, se muestra el aspecto de una calculadora RPN, observad que no tiene la tecla "=", pero si una tecla "Enter" (el igual que aparece debajo de la teclas "." Y "Enter" tienen otras funciones).

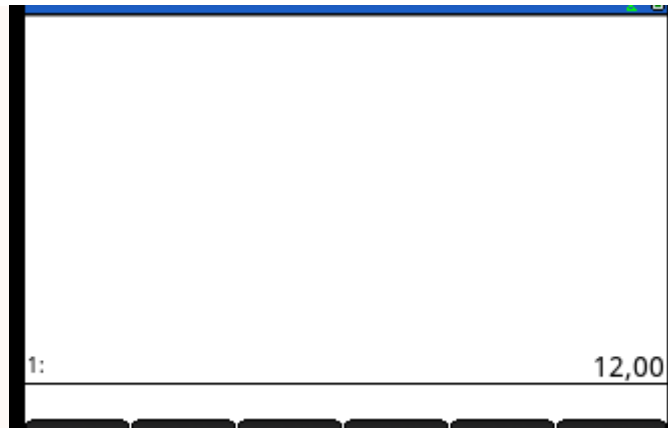


Las calculadoras RPN utilizan notación postfija o RPN, es decir primero van los operandos y después los operadores. En la pila solamente se introducen los números (enteros y float).

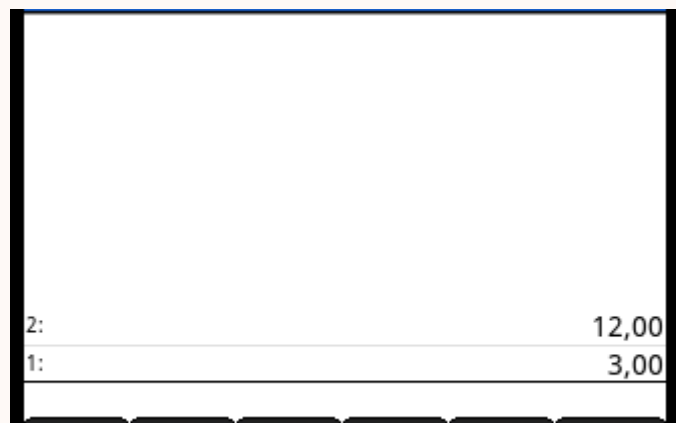
Por ejemplo, para calcular $12+3$ se debe escribir **"12 Enter 3 Enter +"**

La evaluación se realiza utilizando una pila, así para evaluar **"12 Enter 3 Enter +"** el proceso es el siguiente:

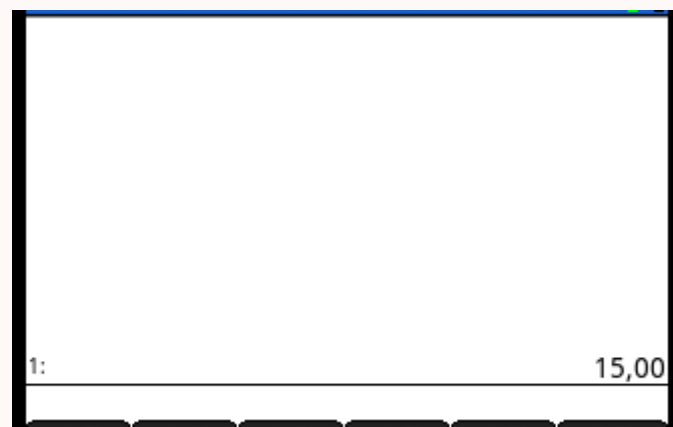
- se teclea 12 (se inserta 12 en la pila pulsando la tecla "Enter")



- se teclea 3 (se inserta 3 en la pila pulsando la tecla "Enter")



- se teclea + (al ser un operador binario se sacan los dos últimos elementos de la pila y el resultado se mete en la pila)



- En la pila solamente queda el valor 15



La pila puede ser "infinita"

Con RPN no hacen falta paréntesis, así para calcular " $(7+5)/(2+1)$ " en RPN sería:

7 Enter 5 Enter + 2 Enter 1 Enter + /

Hay calculadoras modernas que trabajan con RPN, por ejemplo, **HP Prime**. Hay versión "física" y virtual (para Win y Mac)

<https://support.hp.com/es-es/product/hp-prime-graphing-calculator/5367459>

<https://support.hp.com/es-es/product/hp-prime-graphing-calculator/5367459/model/5367460/manuals>

<https://support.hp.com/es-es/drivers/selfservice/hp-prime-graphing-calculator/5367459/model/5367460>

La notación postfija o RPN se puede estudiar en

https://es.wikipedia.org/wiki/Notaci%C3%B3n_polaca_inversa

Para implementar la calculadora es necesario usar una estructura de datos pila LIFO.

Puede verse una implementación de una pila en el ejercicio:

<http://di002.edv.uniovi.es/~cueva/php/021-Clase-PilaLIFO.php>

Ejercicio 4

Tarea 1. Realización de una aplicación web que consuma un tipo servicios Web en función del de la terminación del UOXXXXXX de cada estudiante.

- UOXXXXXX terminados en 0 o en 5: Consumo de servicios Web de precios del gas natural
- UOXXXXXX terminados en 1 o en 6: Consumo de servicios Web de precios del petróleo
- UOXXXXXX terminados en 2 o en 7: Consumo de servicios Web de precios del oro
- UOXXXXXX terminados en 3 o en 8: Consumo de servicios Web de precios del cobre
- UOXXXXXX terminados en 4 o en 9: Consumo de servicios Web de precios de la plata

Se utilizará HTML5, CSS y PHP.

Se valorará la presentación, la complejidad de la aplicación, la originalidad, la creatividad y los elementos usados de **PHP** y de **consumo de servicios Web**.

Epílogo. Todas las tareas se presentan en la misma sub-carpeta denominada **Ejercicio-4** que debe contener como mínimo los archivos:

- Ejercicio4.php
- Ejercicio4.css

Guía para resolver el ejercicio 4

El estudiante deberá buscar un proveedor de servicios Web de la temática que le ha tocado. Es importante elegir un proveedor con una buena documentación de la API y si es posible con buenos ejemplos.

Ejemplos:

<http://di002.edv.uniovi.es/~cueva/php/100-JSON-OpenWeatherMap.php>

<http://di002.edv.uniovi.es/~cueva/php/101-XML-OpenWeatherMap.php>

<http://di002.edv.uniovi.es/~cueva/php/102-Objetos-PHP-Flickr%20.php>

<http://di002.edv.uniovi.es/~cueva/php/103-JSON-Flickr%20.php>

Ejercicio 5

Se debe crear el usuario y la password específico en MySQL para poder utilizarlo en el resto de los ejercicios. Es obligatorio usar los siguientes datos de usuario, host y password.

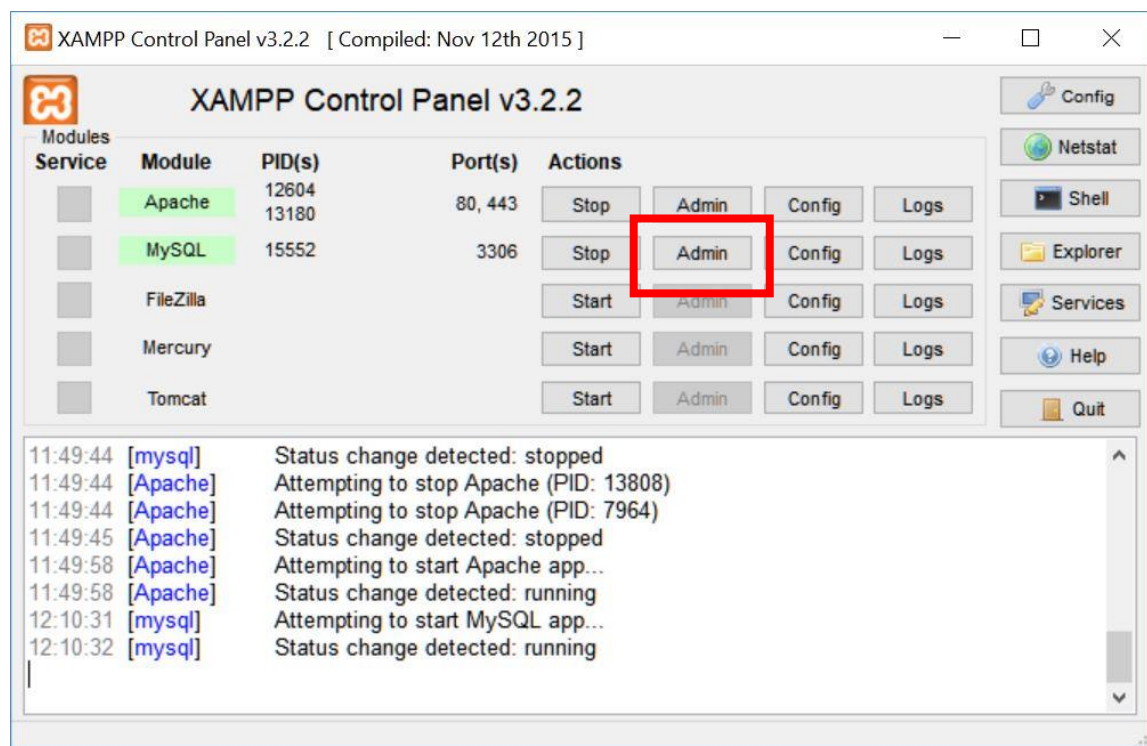
DATOS DEL USUARIO, HOST Y PASSWORD – OBLIGATORIAMENTE DEBEN SER:

USUARIO: **DBUSER2021**

Nombre del HOST: **Local** (localhost)

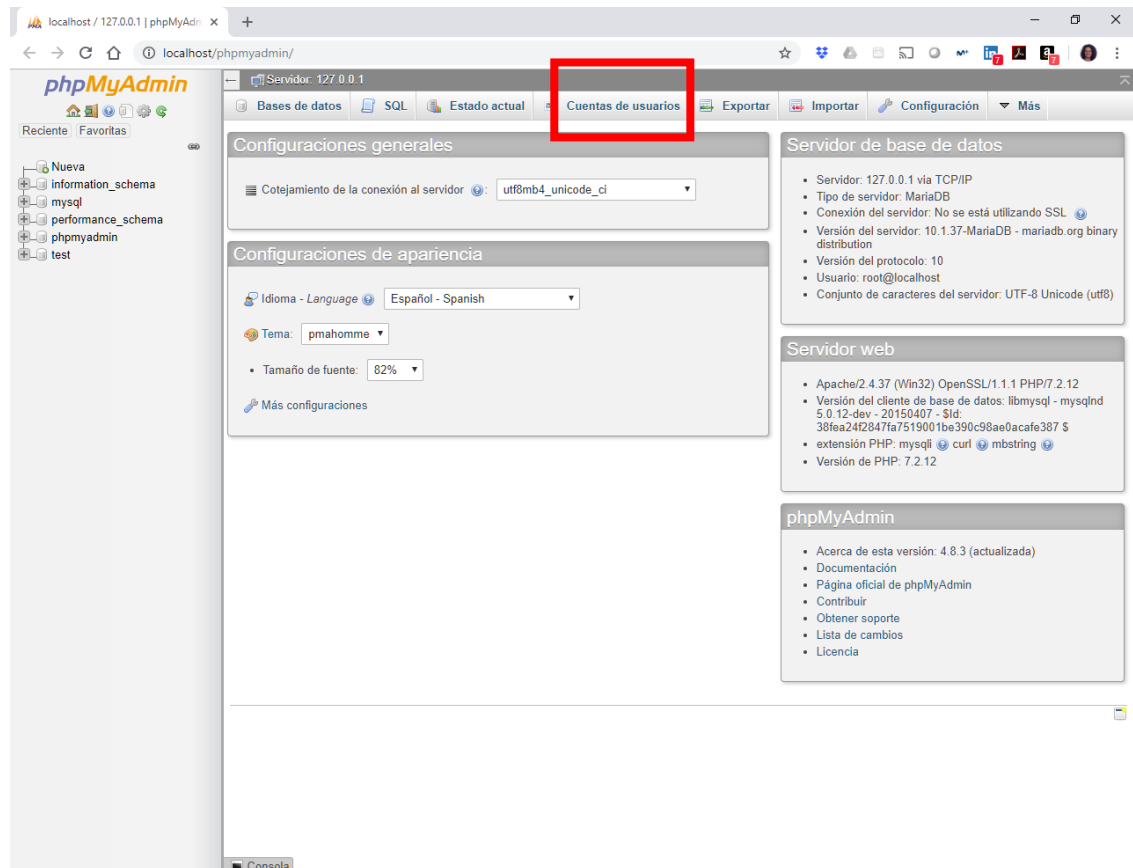
PASSWORD: **DBPSWD2021**

Una vez arrancado MySQL en XAMPP, pulsamos en Admin para entrar en la administración del MySQL

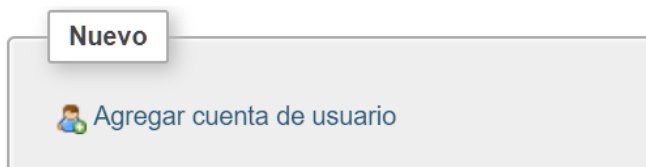


En el navegador web se abrirá la web <http://localhost/phpmyadmin/> desde donde se pueden administrar las bases de datos y los usuarios.

Seleccionar “**Cuentas de Usuarios**” en el menú, aparece la lista de los usuarios definidos, el tipo y sus privilegios.



Pulsar en “Agregar cuenta de usuario”.



Introducir el nuevo usuario con los datos especificado y con Privilegios globales. El resto de opciones se dejan por defecto. Pulsar en “Continuar” y el usuario ya estará creado.

Guía para resolver el ejercicio 5

Seguir el enunciado y consultar el tema de teoría “3 - PHP: Archivos y MySQL”

Ejercicio 6

DATOS DEL USUARIO, HOST Y PASSWORD – OBLIGATORIAMENTE DEBEN SER:

USUARIO: **DBUSER2021**

Nombre del HOST: **Local** (localhost)

PASSWORD: **DBPSWD2021**

Tarea 1. Escribir un archivo en HTML5 denominado **Ejercicio6** (con extensión **.php**) que referencie a un archivo CSS denominado **Ejercicio6** (con extensión **.css**) cuyo objetivo será la gestión de una base de datos MySQL.

Tarea 2. Escribir un **menú** en HTML5 con las siguientes opciones:

- Crear Base de Datos
- Crear una tabla
- Insertar datos en una tabla
- Buscar datos en una tabla
- Modificar datos en una tabla
- Eliminar datos de una tabla
- Generar informe
- Cargar datos desde un archivo CSV en una tabla de la Base de Datos
- Exportar datos a un archivo en formato CSV los datos desde una tabla de la Base de Datos
-

Se debe crear una clase en PHP denominada **“BaseDatos”** con los métodos y atributos necesarios para realizar la funcionalidad requerida en el menú.

Tarea 3. Crear la Base de Datos con una tabla denominada **“PruebasUsabilidad”** con los campos/columnas necesarias para almacenar la siguiente información para evaluar una tarea de una prueba de usabilidad:

- Código de identificación de la persona que realiza la prueba (puede ser el DNI)
- Nombre de la persona que realiza la prueba
- Apellidos de la persona que realiza la prueba
- E-mail de la persona que realiza la prueba
- Teléfono de la persona que realiza la prueba
- Edad de la persona que hace la prueba de usabilidad
- Sexo
- Nivel o pericia informática de la persona calificada por el evaluador con un número de 0 a 10.
- Tiempo que ha tardado en hacer la tarea en segundos
- Si la tarea la ha realizado correctamente o no
- Comentarios sobre problemas encontrados al usar la aplicación
- Propuestas de mejora de la aplicación
- Valoración de la aplicación por parte del usuario de 0 a 10

Tarea 4. Implementar las interfaces de usuario para realizar las operaciones de inserción, búsqueda, modificado y borrado de los datos en la tabla **“PruebasUsabilidad”**.

Tarea 5. Diseñar una interfaz de usuario que muestre el informe con los siguientes datos:

- Edad media de los usuarios
- Frecuencia del % de cada tipo de sexo entre los usuarios
- Valor medio del nivel o pericia informática de los usuarios
- Tiempo medio para la tarea
- Porcentaje de usuarios que han realizado la tarea correctamente

- Valor medio de la puntuación de los usuarios sobre la aplicación

Tarea 6. Implementar la exportación de los datos de la tabla “PruebasUsabilidad” de la Base de Datos a un archivo de texto en formato CSV denominado “pruebasUsabilidad.csv”.

Tarea 7. Implementar la carga de datos desde un archivo de texto en formato CSV en la tabla “PruebasUsabilidad” de la Base de Datos.

Epílogo. Todas las tareas se presentan en la misma sub-carpeta denominada **Ejercicio-6** que debe contener todos los archivos especificados. Se debe obligatoriamente proporcionar el archivo “**pruebasUsabilidad.csv**”.

Guía para resolver el ejercicio 6

Consultar los ejercicios del tema de teoría “3 - PHP: Archivos y MySQL”

Por seguridad cuando las sentencias de SQL provienen de una fuente externa (como por ejemplo un formulario) siempre se deben preparar y no ejecutarse directamente.

Consultar en el manual de PHP:

<https://www.php.net/manual/es/class.mysqli.php>

Consultar el tutorial:

<https://www.ionos.es/digitalguide/servidores/know-how/guia-para-aprender-a-utilizar-mysql/>

Ejercicio 7

DATOS DEL USUARIO, HOST Y PASSWORD – OBLIGATORIAMENTE DEBEN SER:

USUARIO: **DBUSER2021**

Nombre del HOST: **Local** (localhost)

PASSWORD: **DBPSWD2021**

Diseñar una base de datos de temática libre con al menos **4 tablas** utilizando PHP y MySQL.

La base de datos debe estar normalizada en, al menos, tercera forma normal.

Se debe incluir un archivo en PDF con el diagrama entidad-relación de la base de datos diseñada

Se deben utilizar obligatoriamente clases en PHP.

Se valorará la presentación, la complejidad de la aplicación, la originalidad, la creatividad y los elementos usados de PHP.

Epílogo. Todos los archivos se presentan en la misma sub-carpeta denominada **Ejercicio-7** que debe contener todos los archivos.

Se deben proporcionar archivos con datos para cargarlos en las tablas y poder reproducir las pruebas sobre la Base de Datos.

Guía para resolver el ejercicio 7

Consultar los ejercicios del tema de teoría “3 - PHP: Archivos y MySQL”

Por seguridad cuando las sentencias de SQL provienen de una fuente externa (como por ejemplo un formulario) siempre se deben preparar y no ejecutarse directamente.

Consultar en el manual de PHP:

<https://www.php.net/manual/es/class.mysql.php>

Consultar el tutorial:

<https://www.ionos.es/digitalguide/servidores/know-how/guia-para-aprender-a-utilizar-mysql/>

Criterios de evaluación

La práctica entregada debe cumplir estrictamente las siguientes normas, **en caso contrario la calificación será cero puntos**:

- Se deben entregar todos los ejercicios y todas las tareas del guion de la práctica.
- Se debe utilizar PHP “puro” sin bibliotecas ni extensiones. No se permite el uso de ECMAScript, ni de JavaScript.
- Se usará el paradigma de orientación a objetos obligatoriamente en PHP.
- Los archivos CSS estarán independientes, no incrustados en archivos HTML o PHP.
 - El objetivo es tener en archivos separada la presentación.
- TODOS los documentos HTML deben ser HTML5 válidos utilizando el validador de lenguajes de marcado del W3C
- TODAS las hojas de estilo CSS que se utilizan deben ser comprobadas utilizando el validador CSS del W3C
- Debe comprobarse la **adaptabilidad** con las herramientas de comprobación de la adaptabilidad: Google Mobile-friendly y Screenfly
- Debe comprobarse la **accesibilidad** con las herramientas de accesibilidad: TAW, Wave y aChecker. Deben tener 0 errores en modo automático para nivel AAA de las WCAG 2.0 o superior.
- TODOS los ejercicios deben funcionar en todos los navegadores de referencia, Chrome, Firefox, Microsoft Edge y Opera. Opcionalmente para sistemas MacOS el navegador Safari.
- Por seguridad es obligatorio “preparar” las sentencias SQL cuando se usen fuentes externas (por ejemplo formularios).
- La base de datos debe estar normalizada al menos en tercera forma normal.
- Es obligatorio usar los identificadores que se especifican para el usuario y la *password* y que están dados en los ejercicios de bases de datos.

Entrega de la práctica

Se deben entregar **todo en un archivo** comprimido denominado “UOXXXXXX-practica4.zip” (tamaño máximo 90MB) donde el UOXXXXXX será el identificador del estudiante en la Universidad de Oviedo

El archivo comprimido debe contener una carpeta “PRACTICA-4” con TODOS los ejercicios de la práctica, una carpeta por ejercicio:

- Ejercicio-1
- Ejercicio-2
- ...
- Ejercicio-7

Cada ejercicio puede dividirse en tareas, que se presentará en carpetas separadas: tarea1, tarea2, etc. Según lo especificado en el “Epílogo” de cada ejercicio.

Se considerarán como “no entregadas” las prácticas que no cumplan el formato de entrega, el nombrado de los archivos y que no contengan todos los ejercicios de la práctica.

El plazo de entrega se encuentra publicado en el Campus Virtual y la tarea de entrega se habilitarán con anterioridad suficiente.