

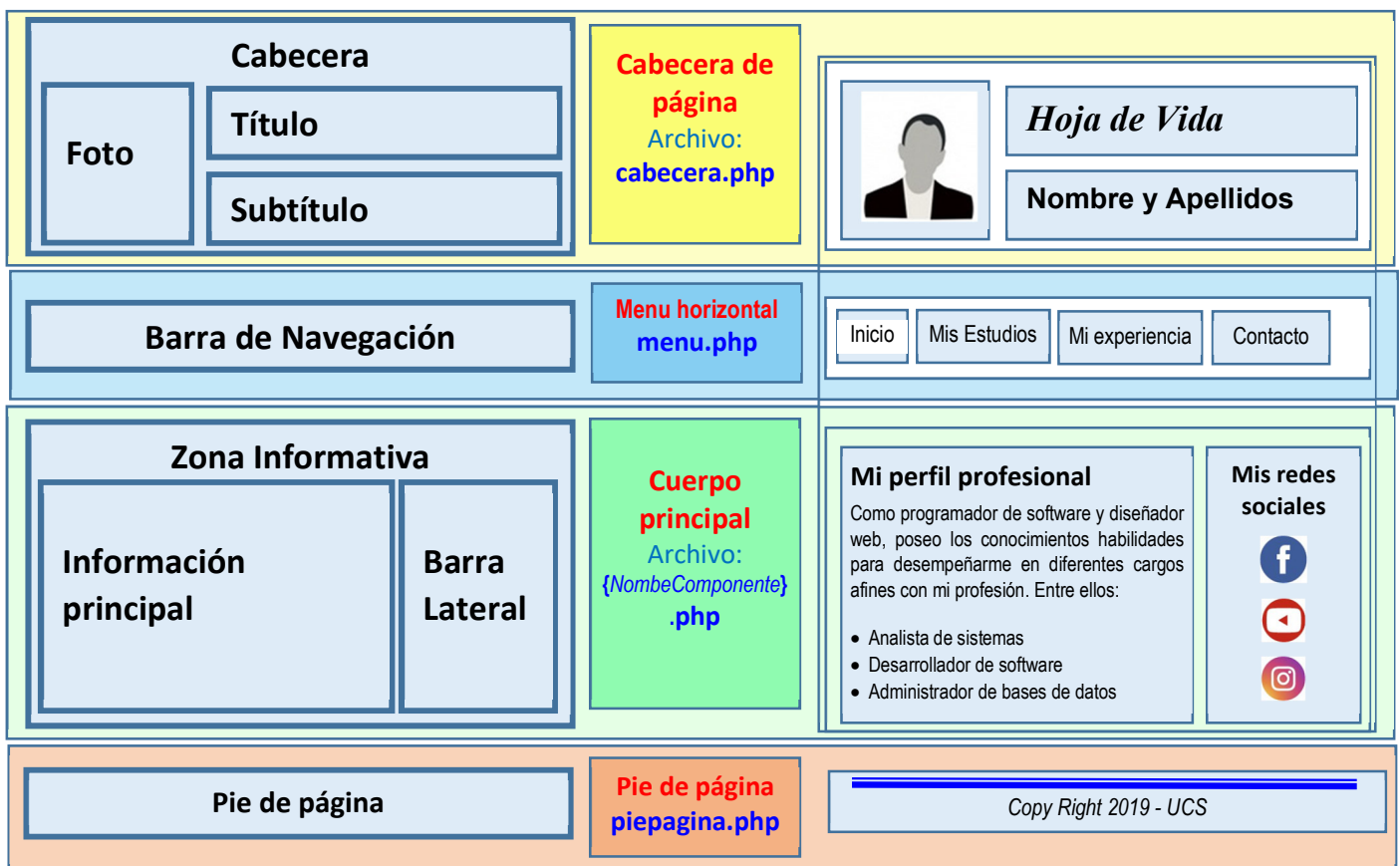
Integración de Bases de datos MySQL en proyectos WEB con PHP

TALLER de Conceptos básicos de consultas MySQL y volcado de tablas en PHP

ESQUEMA O ESTRUCTURA DE LA PÁGINA WEB

Continuando con el desarrollo del proyecto WEB correspondiente a la Hoja de vida personal. En el presente taller se explicarán y desarrollarán los procedimientos básicos para construir una página web dinámica con datos obtenidos del resultado de una consulta de datos previamente establecida, es decir sin petición externa requerida por el usuario con un buscador, sino el volcado general de información almacenada en una tabla, obteniendo el total de los datos con una instrucción SELECT en una Base de datos MySQL y volcando la información resultante de dicha consulta en una tabla HTML.

En el presente ejercicio igualmente se incorporarán elementos básicos estructurales de un diseño WEB, separando los contenidos de los 4 componentes del diseño del FRONT-END, dividiendo en 4 partes la plantilla WEB, de tal forma que el desarrollo se vuelva modular y los componentes repetitivos queden aislados en módulos independientes que pueden ser reutilizados en cada una de las páginas interiores de la navegación. En la siguiente imagen se identifican los 4 módulos principales de la plantilla principal del diseño o maqueta de la página web.



1. Descomposición de cada página en componentes funcionales:

Recordemos que el sitio WEB Básico está conformado por 4 módulos así:

- Index.html página principal o Home del sitio web
- misestudios.html página informativa referente al detalle de educación y otros estudios.
- miexperiencia.html página informativa referente al detalle de la experiencia laboral
- contacto.html página que contiene el formulario para contacto del visitante

Con el fin de darle un mejor orden a todos los componentes, teniendo en cuenta que en dichos módulos, los componentes, Cabecera, menú y pie, son idénticos o tienen la misma información. Estos serán separados del componente modular y ahora cada uno de estos archivos contendrá únicamente la información que le corresponde y los otros tres módulos se trasladarán a archivos separados. También es de tener en cuenta que estos archivos anteriormente estaban nombrados como extensión **.html**, ahora pasarán a nombrarse con extensión **.php**.

Nótese que lo que anteriormente era un solo archivo, por ejemplo el archivo principal **index.html** ahora ha quedado dividido en **4 componentes**.

- **header.php** Corresponde a la cabecera de cada una de las páginas. Este será el mismo componente que se utilizará en todos los otros módulos del sitio WEB para encabezar cada páginas publicada. En caso de requerir algún ajuste, la modificación se realizará una sola vez y esta se publicará en cualquier página que lo contenga `<header> ... </header>`.
- **nav.php** Corresponde al menú principal del sitio WEB, el cual en el diseño utilizado para el presente sitio WEB, se muestra en formato horizontal. Este se encuentra enmarcado dentro de las etiquetas `<nav> </nav>`.
- **{nombre componente}.php** Corresponde al componente central o PRINCIPAL del sitio. Este se encuentra enmarcado dentro de las etiquetas `<main> </main>`.

El nombre del archivo para cada componente **{nombre componente}.php** corresponderá con el módulo funcional del sitio WEB, es decir, se tendrían los siguientes archivos:

- Index.php página principal o Home del sitio web
 - misestudios.php página informativa referente al detalle de educación y otros estudios.
 - miexperiencia.php página informativa referente al detalle de la experiencia laboral
 - contacto.php página que contiene el formulario para contacto
- **footer.php** Corresponde al componente que muestra la parte inferior del sitio WEB. Allí regulamente se publica o informan los datos de contacto del sitio, versión, derechos de autor entre otros datos. Este se encuentra enmarcado dentro de las etiquetas `<footer>...</footer>`.

De acuerdo con lo anterior, cada uno de los 4 archivos resultantes estarían conformados de la siguiente forma:

1.1. Archivo: **header.php**

Contiene únicamente las líneas de código PHP y HTML que permiten publicar el diseño y contenido de la cabecera de todas las páginas pertenecientes al sitio WEB. Este archivo será invocado e insertado desde los 4 módulos del sistema, según se indica más adelante.

Observe en el código que el menú se encuentra enmarcado dentro de las etiquetas < header> .. </ header>

```
<?php
// Archivo Componente: cabecera.php
// Proyecto: Página Web Mi hoja de vida personal
// Este archivo contiene los elementos que se muestran en la parte superior
// (cabecera) de todos los módulos principales que conforman el sitio web
?>

<link rel="stylesheet" type="text/css" href="css/style.css">
<header>
    <figure>
        
        </figure>

        <div id="titulo"> <!-- sector para el título HV -->
            <p>Hoja de Vida</p>
        </div>

        <div id="subtitulo"> <!-- sector para el subtítulo HV -->
            <p>Uriel Castañeda Sierra</p>
        </div>
    </header>
```

1.2. Archivo: **nav.php**

Contiene las líneas de código PHP y HTML que conforman la estructura del menú de navegación principal. De acuerdo con el diseño y la configuración establecida en el archivo de estilos CSS, este será publica en formato horizontal.

Observe en el código que el menú se encuentra enmarcado dentro de las etiquetas <nav> </nav>

Recordemos que cada opción del menú contiene configurado el hipervínculo a cada uno de los módulos del sistema. De esta forma se permitirá la navegación. En este caso, todos serán con extensión PHP, teniendo en cuenta que en cada uno de ellos habrán componentes dinámicos escritos en lenguaje php.

```
<?php
// Archivo Componente: menu.php
// Proyecto: Página Web Mi hoja de vida personal
// Este archivo contiene los elementos que conforman el MENÚ PRINCIPAL para
// navegación en todo el sitio WEB.
// aquí se definen los hipervínculos a cada módulo del sistema
?>

<link rel="stylesheet" type="text/css" href="css/style.css">
<nav> <!-- Aquí empieza la definición del menú de navegación -->
<ul>
  <li><a id="inicio"; href="index.php">Inicio</a></li>
  <li><a href="misestudios.php">Mis Estudios</a></li>
  <li><a href="miexperiencia.php">Mi Experiencia</a></li>
  <li><a id="contacto"; href="contacto.php">Contacto</a></li>
</ul>
</nav>
```

1.3. Archivo: footer.php

Contiene las líneas de código PHP y HTML que permiten publicar la parte final (costado inferior) de cada una de las páginas que conforman el sitio WEB. De acuerdo con el diseño, en este caso se mostrará una línea horizontal y un texto que contiene información de autoría y año de publicación.

Observe en el código que el menú se encuentra enmarcado dentro de las etiquetas <footer> ... </ footer>

```
<?php
// Archivo Componente: piepagina.php
// Proyecto: Página Web Mi hoja de vida personal
// Este archivo contiene los elementos que se muestran en la parte inferior
// (Pie de página) de cada uno de los módulos funcionales del sitio web
// aquí se publican datos como el Autor, fecha de versión, inf. de contacto, etc.
?>

<link rel="stylesheet" type="text/css" href="css/style.css">
<footer> <!-- Aquí empieza la definición del pie de página -->
  <hr id="lineaFooter">
  Power by: SENA - CDAE, Villeta Colombia 2022 - By Uriel C.S.
</footer>
```

1.4. EL CUERPO DE LA PAGINA WEB (Archivo componente MODULAR) **index.php**

Este archivo contiene toda la estructura del módulo funcional del sitio web. En este caso, el INDEX o archivo principal de acceso (Home Page).

Contiene las líneas de código PHP y HTML que permiten configurar la página, identificando, versión de HTML, idioma, título e ícono que se publica en el navegador, palabras clave, juego de caracteres, entre otros.

También hará referencia al archivo de estilos CSS y además realizará el llamado para incluir (insertar) los componentes definidos externamente que conforman toda la estructura de la página, en este caso, los archivos:

- **header.php**
- **nav.php**
- **footer.php**

El contenido principal de este archivo se enmarca en las etiqueta **<main> ... </main>**

Se debe tener en cuenta que la estructura de este archivo es la misma en todos los otros módulos del sistema que necesiten mostrar los también los componentes de CABECERA, MENÚ y PIE DE PÁGINA, en este proyecto hacemos referencia a los otros tres módulos funcionales (misestudios, miexperiencia y contacto). Solamente cambiará el contenido enmarcado en las etiquetas **<main>**.

En la parte inicial del archivo **index.php** se conservará un **espacio para el código PHP** que permitirá iniciar y ejecutar procesos en el servidor al momento de construir la página. En este espacio se codificarán los procesos para carga de información de bases de dato, lectura de información del sistema, entre otros.

```
<?php
// Archivo Componente: index.php
// Proyecto: Página Web Mi hoja de vida personal
// -----
// En este archivo se establece la estructura de la página WEB del módulo
// INDEX (HOME PAGE o Página principal) del sitio web.
// En su orden, aquí se organizan los COMPONENTES que conforman la página:
// - Parámetros que configuran la página, tales como:
//   Versión de html e idioma de la página
//   Título e ícono que se mostrará en la pestala del navegador
//   Juego de caracteres compatibles con, en este caso UTF-8 para latinoamérica
//   Archivo que contiene los estilos CSS
// - Cabecera,
// - Navegación,
// - Cuerpo principal (<Main>) en este caso, el HOME PAGE
// - Pie de página

// ----- en este espacio insertamos el código PHP necesario para
// iniciar y ejecutar procesos en el servidor al momento de construir la página

?>
```

A continuación se muestra la 2ª parte del archivo, correspondiente a la estructura lógica y organizada del código PHP y HTML integrados que permitirán construir la página WEB que el servidor entregará al cliente usuario cuando sea requerido el módulo **index.php** (continuación del código de la página anterior).

```
<!-- Parámetros de configuración de la página web  HTML -->
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Mi hoja de vida</title>
    <meta charset="utf-8">
    <link rel="icon" href="imagenes/icono_web1.png" type="image/png" sizes="20x20">
    <link rel="stylesheet" type="text/css" href="css/style.css">
  </head>

  <!-- Aquí inicia la definición de la Estructura general o CUERPO de la página WEB -->
  <body>

    <?php
      // --- Importamos LA CABECERA <header> definido en el archivo header.php
      include("header.php") ;

      // --- Importamos el MENU (componente <nav>) definido en el archivo nav.php
      include("nav.php") ;
    ?>

    <!-- Módulo funcional: INDEX (Home page) -->
    <!-- Aquí inicia el código del CUERPO PRINCIPAL de esta página -->

    <main>      <!-- definimos los contenidos agrupados en la etiqueta <main> "-->

      <section>  <!-- definimos secciones de contenidos (LADO IZQUIERDO)-->
        <article id="article1">
          <p> aquí presento el texto de mi primer artículo </p>
        </article>
        <article id="article2">
          <p> aquí presento el texto de mi segundo artículo </p>
        </article>
      </section> <!-- Aquí termina la sección de artículos -->

      <aside>    <!-- definimos los apartados o comentarios del LADO DERECHO -->
        <p> Aquí mostramos el menú de redes sociales</p>
        <p> incluiremos los íconos y links cada red</p>
      </aside>   <!-- Aquí terminan los apartados o comentarios

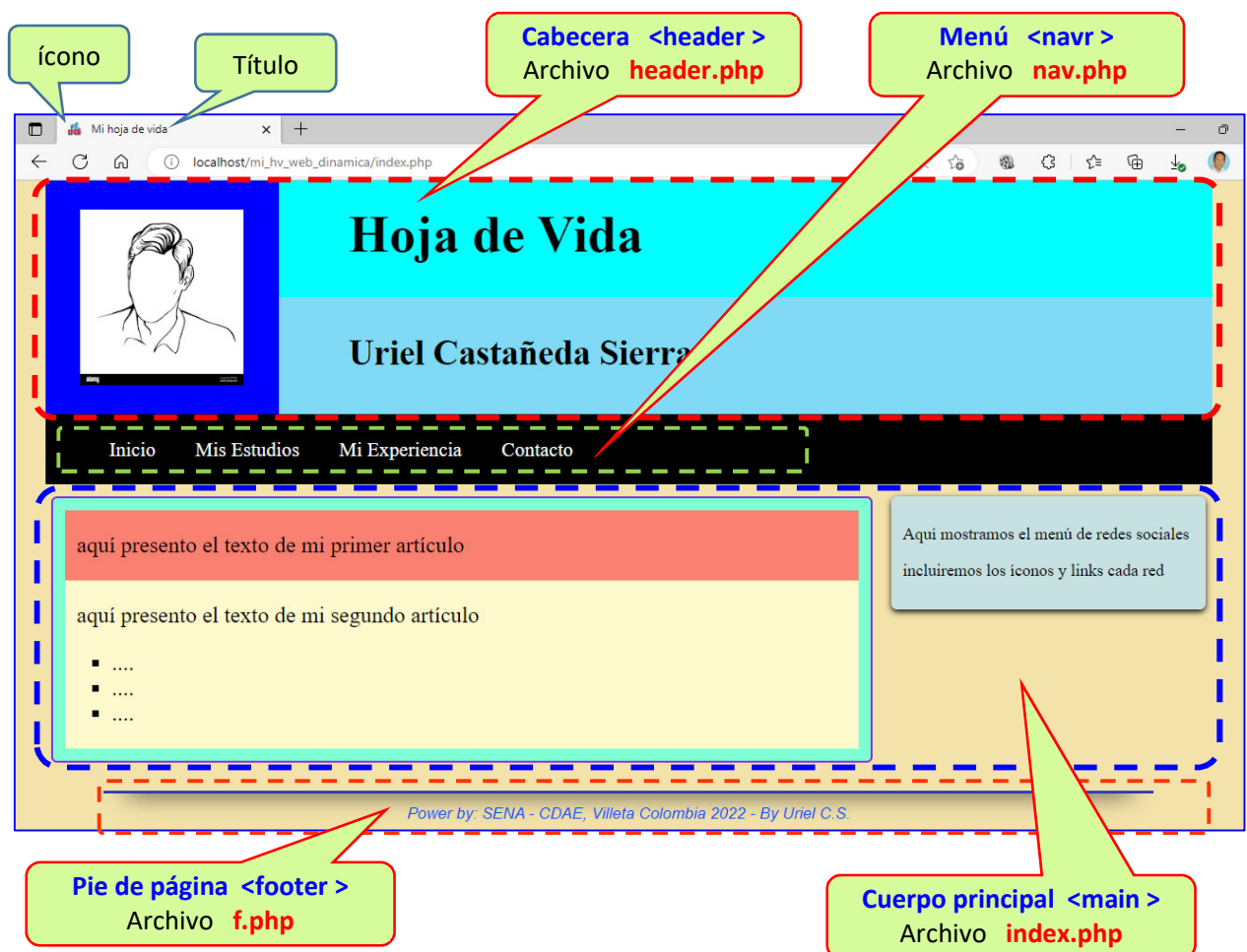
    </main>      <!-- Aquí terminan los contenidos principales -->

    <?php
      // --- Importamos el componente <footer> definido en el archivo cabecera.php
      include("footer.php") ;
    ?>

    </body>      <!-- Aquí terminan el cuerpo de la página WEB
  </html> >
```

Integrados los 4 archivos codificados previamente, observaremos una página web resultante con una estructura similar a la que se muestra a continuación, teniendo en cuenta que ya deberán estar debidamente codificados los archivos de ESTILOS CSS y el archivo del ÍCONO, estos deben estar almacenados en su carpeta o ruta correspondiente, así:

- [css/style.css](#)
- [imagenes/icono_web1.png](#)



2. Base de datos del sitio web dinámico:

En el desarrollo del presente proyecto estableceremos conexión con una base de datos MySQL, de donde extraerá la información que será publicada en las diferentes hojas que conforman el sitio web, En este caso solamente haremos referencia a la información de **ESTUDIOS** realizados por el propietario de la hoja de vida.

El nombre sugerido para la base de datos será: **web_personal_APRENDIZ**

Para el correcto funcionamiento del sistema, el computador donde se esté realizando el ejercicio deberá disponer del software necesario para ejecutar el servidor Apache y el sistema gestor de Bases de Datos (SGBD) MySQL.

**** IMPORTANTE ! ****

Teniendo en cuenta que en el desarrollo de los diferentes ejercicios de aprendizaje se tiene como objetivo conocer y llevar a la práctica los conceptos de Bases de datos relacionales y las condiciones de INTEGRIDAD REFERENCIAL, la base de datos y las tablas deberán ser creadas definiendo correctamente el **MOTOR innodb** y el **SET de CARACTERES utf8** con **COTEJAMIENTO utf8_spanish_ci**

2.1. Creación de la Base de Datos **web_personal_APRENDIZ**

Esta base de datos deberá ser creada utilizando cualquiera de los sistemas de administración de Bases de datos, como por ejemplo, PHPMyAdmin, MyWebSQL, HeidiSQL o MySQL WorkBench, entre otros.

Utilizando la ventana de consola SQL, cree la base de datos utilizando el siguiente script SQL ...

```
-- Creación de la base de datos web_personal_APRENDIZ incorporando el juego  
-- de caracteres utf8_spanish_ci  
-- En caso de ya tener creada la Base de datos, comentar la línea que incluye  
-- el comando CREATE DATABASE.
```

```
CREATE DATABASE web_personal_APRENDIZ CHARACTER SET utf8 COLLATE  
utf8_spanish_ci;
```

2.2. Creación de la tabla **estudios**

Una vez creada la base de datos, procedemos con la creación de la tabla donde se guardará la información de los estudios realizados. Para ello, después de analizar las necesidades de información que requiere ser mostrada en pantalla, se determinó que será necesario informar por cada estudio realizado los siguientes datos:

- tipo_estudio (por ejemplo, Primaria, Bachillerato, técnico, Tecnólogo, profesional, etc)
- nombre_estudio (para indicar el nombre del estudio cursado, p.ej: Programación de Software)
- institucion_educ (para indicar el nombre del establecimiento educativo donde estudió.)
- ciudad (ciudad sede de los estudios realizados)
- fecha_graduacion (fecha que finalizó y obtuvo el título de dicho estudios, en formato aaaa-mm-dd)
- cantidad_horas (cantidad de horas acreditadas según lo indicado en el certificado de estudios)

Para este caso se incluirá un **campo para llave** para identificación de cada este registro. Este será definido con el nombre **id** y será numérico autoincrementable.

De igual forma que la creación de la base de datos, en la línea de comando de la consola de código SQL ingresar el siguiente script...

```
-- Creación de la tabla ESTUDIOS
-- Primero borramos la tabla anterior en caso que esta exista.

DROP TABLE IF EXISTS estudios;

-- aquí creamos la tabla

CREATE DATABASE web_personal_APRENDIZ CHARACTER SET utf8 COLLATE
utf8_spanish_ci;

TABLE estudios (
    id                int PRIMARY KEY NOT NULL AUTO_INCREMENT,
    tipo_estudio      varchar (25) NOT NULL,
    nombre_estudio    varchar (60) NOT NULL,
    institucion_educ  varchar (60) NOT NULL,
    ciudad            varchar (40),
    fecha_graduacion  DATE,
    cantidad_horas    int )
ENGINE = InnoDB DEFAULT CHARSET = utf8 COLLATE utf8_spanish_ci;
```

2.3. Carga de datos en la Tabla estudios

Teniendo en cuenta que el presente ejercicio será solamente para consultar datos almacenados en una tabla, es necesario realizar previamente el proceso de carga o volcado que los datos en la base de datos, en este caso, en la tabla “estudios”.

De acuerdo con la estructura definida para la tabla de estudios, realice el proceso de carga de información tomando como ejemplo el siguiente script SQL para inserción de datos ... (estos datos pueden ser reemplazados acorde con la información perteneciente a cada estudiante).

```
-- Aquí insertamos todos los registros en la tabla estudios

INSERT INTO estudios (tipo_estudio, nombre_estudio, institucion_educ, ciudad,
fecha_graduacion, cantidad_horas) VALUES
('Posgrado', 'Especialización en Negocios en Internet', 'Universidad ICESI -
Universidad de la Sabana', 'Chia', '2004-10-04', null),
('Profesional', 'Ingeniería de Sistemas', 'Universidad Católica de
Colombia', 'Bogotá', '1998-03-28', null),
('Bachillerato técnico', 'Bachiller Técnico en Mecánica Industrial', 'Instituto
técnico central - La Salle', 'Bogotá', '1988-12-06', null),
('Diplomado', 'Gerencia de Proyectos bajo lineamientos PMI', 'Universidad
Piloto', 'Bogotá', '2016-12-10', 240);
```



3. Estructura de archivos que conformarán el sitio web:

Ya conocemos la estructura básica de archivos sugerida para un proyecto de sitio WEB Estático. Tomando como referencia dicha estructura, nuestro proyecto de página web dinámica en PHP estará conformado por una carpeta principal que regularmente es identificada con el mismo nombre del proyecto o nombre del dominio del sitio WEB.

En nuestro caso a dicha carpeta le llamaremos **mi_hv_web_dinamica**.

En esta carpeta se alojarán las mismas carpetas utilizadas en una página web estática, es decir, las carpetas, **imágenes**, **css** y **js**. Adicionalmente se incorpora una nueva carpeta que llamaremos **include**, que será utilizada para alojar los archivos extras que serán incluidos o importados dentro del código de los diferentes componentes PHP del sitio web. En este caso haremos referencia a los siguientes archivos que utilizaremos en el desarrollo del presente ejercicio.

- **script_bd1.sql** Contiene el script en lenguaje SQL para crear las tablas que se utilizarán en el proyecto, para este caso el script1 solo hará referencia a la creación de la tabla **estudios** de la base de datos **web_personal_APRENDIZ**. (scripts utilizados en el paso anterior y que pueden seguir siendo utilizados para realizar la carga de la Base de datos en el Hosting).
- **herramientas.php** Contiene varios métodos escrito en lenguaje PHP, que serán utilizados con varias finalidades para ayuda al programador durante los procesos de desarrollo o construcción del código del sitio web. Especialmente para escribir en consola el contenido de variables o matrices al momento de depurar el código de la página y también al renderizar (construir) las tablas o listados resultantes de las consultas SQL a la base de datos. Para el presente ejercicio haremos uso de las siguientes funciones incluidas en esta librería personal:

write_to_console(\$Nombre_variable)

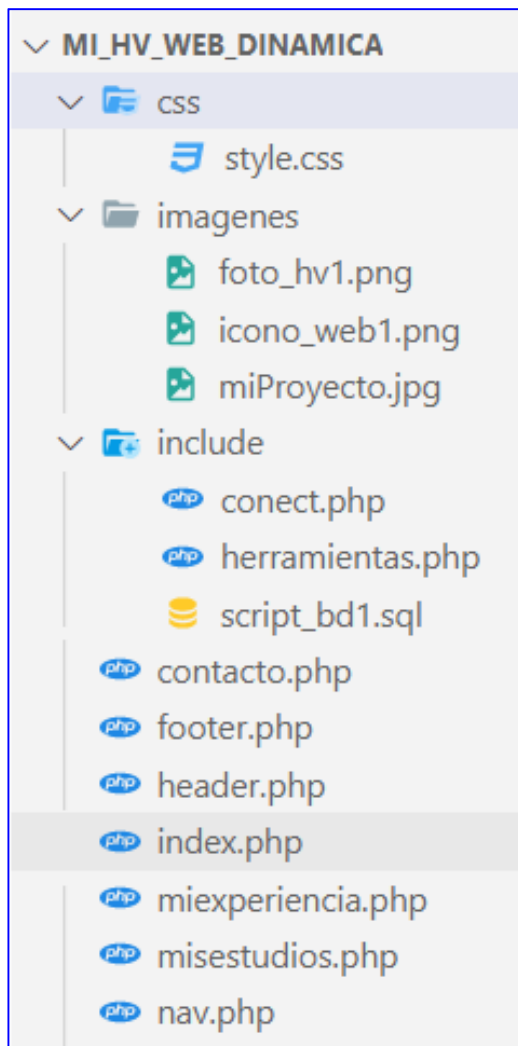
Esta función permite escribir en consola cualquier tipo de dato. Incluyendo el contenido completo de los diferentes tipos de matrices. El resultado escribe en la primera línea el nombre de la variable y luego todos los valores de los atributos o variables que contiene.

write_estado_var(\$Nombre_variable)

Esta función despliega una ventana de mensaje emergente y dentro de ella muestra el valor y actual de una variable.

- **conec.php** Contiene las instrucciones requeridas por el motor de la Base de datos MySQL para realizar la conexión y posterior extracción de datos.

En resumen, en el siguiente gráfico se muestra la estructura de archivos que debe encontrarse construida hasta este momento, considerando los tres archivos antes mencionados, cuyo contenido será mostrado también a continuación.



Estructura de carpetas que conforman el sitio WEB

Ahora podemos observar que los archivos que anteriormente tenían extensión .html han cambiado a extensión .php

Cada uno de los módulos principales se ha simplificado extractando las cabeceras, menú y pie de página en archivos separados, dejando un único archivo para cada uno de estos tres componentes y serán utilizados en cada uno de los componentes que lo requieran.

También se incorporan nuevos archivos que aportarán los componentes dinámicos para que los contenidos del sitio WEB se alimenten desde una Base de Datos.

3.1. Contenido del archivo de estilos (style.css)

Uno de los componentes principales del sitio WEB corresponde al archivo o a los archivos que entregan la configuración a cada elemento codificado en el FRONT. Con el fin de establecer un buen orden en la codificación y organización de los archivos, para el presente proyecto y en general, en la mayoría de sitios web contruidos metódicamente estos archivos se alojan en la **carpeta CSS**.

El archivo **style.css** contiene las definiciones de estilo para la mayoría de etiquetas utilizadas en la codificación de las páginas en todos los módulos del sitio WEB (páginas index, misestudios, miexperiencia, contacto), además de los otros componentes que se incorporen al personalizar su sitio.

A continuación podrá observar la codificación básica establecida para el <body>, las imágenes <figure>, el menú de navegación <nav>, las sesiones <section>, los artículos <article>, el menú de navegación lateral <aside>, y otros componentes enmarcados en etiquetas <div> con su respectivo identificador (#) y las

clases (.). En algunos de los casos configurando efectos especiales tras evento como al pasar el mouse por encima, al hacer click, al cambiar etc.

```
body{
    background: #F3E2A9;
    width: 1000px;
    margin: 0 auto;
}
figure{
    width: 200px;
    height: 200px;
    display: flex;
    justify-content: center;
    align-items: center;
    background: blue;
    float: left;
    margin: 0;
}
#titulo{
    width: 800px;
    height: 100px;
    background: cyan;
    float: left;
}
#subtitulo{
    width: 800px;
    height: 100px;
    background: #81DAF5;
    float: left;
}
#titulo p{
    font-size: 45px;
    margin-top: 20px;
    margin-left: 60px;
    font-weight: bold;
}
#subtitulo p{
    font-size: 30px;
    margin-left: 60px;
    font-weight: bold;
}
section{
    width: 68%;
    background-color: aquamarine;
    border: 2px solid blueviolet;
    border-radius: 5px;
    font-size: 1.2em;
    text-align: justify;
    padding: 10px;
}
section ul {
    list-style-type: square;
}
```

```
nav{
    width: 1000px;
    height: 60px;
    background: black;
    float: left;
}
nav ul li{
    display: inline-block;
    margin: 5px 15px 5px 15px;
}
nav ul li a{
    color: white;
    font-size: 17px;
    text-decoration: none;
    transition: .3s;
}
nav ul li a:hover{
    font-size: 25px;
    color: cyan;
}
nav ul li #contacto:hover{
    font-size: 25px;
    color: red;
}
nav ul li #inicio:hover{
    font-size: 25px;
    color: rgb(255, 255, 0);
}
section, aside{
    display: inline-block;
    margin: 10px 5px 5px 5px;
    vertical-align: top;
    text-align: justify;
}
#article1{
    width: 100%;
    float: left;
    background-color: salmon;
}
#article2{
    width: 100%;
    float: left;
    background-color: lemonchiffon;
}
section p{
    padding: 0px 10px ;
}
section h4 p{
    margin: -15px 0px;
}
```

```
aside{
    width: 180px;
    position: relative;
    background-color: #C7E1E2;
    box-shadow: 0px 2px 5px #1c1a19;
    border-radius: 5px;
    padding: 10px;
    color: black;
    font-size: 0.9em;
}
#mi_proyecto{
    width: 100%;
}
#ImgMiProyecto1 {
    height: 85%;
    width: 85%;
    border: 5px solid #eee;
    margin: 5px 5px 5px 5px;
    padding: 5px 5px ;
    -webkit-transition: all 1s ease;
    -moz-transition: all 1s ease;
    -o-transition: all 1s ease;
}
#ImgMiProyecto1:hover {
    height: 110%;
    width: 110%;
    margin-left: -20px;
}
footer{
    text-align: center;
    font-family: sans-serif;
    font-style: italic;
    font-size: 0.85em;
    color: rgb(0, 76, 255);
}
#lineaFooter {
    border:none;
    height: 0px;
    width: 90%;
    height: 50px;
    margin-top: 0;
    border-bottom: 3px solid #2d3db2;
    box-shadow: 0px 20px 20px -20px #333;
    margin: -50px auto 10px;
}
.labelTab1 {
    text-align: left;
    font-family: sans-serif;
    font-style: italic;
    font-size: 0.6em;
    color: rgba(36, 59, 112, 0.705);
}
```

3.2. Contenido del archivo de herramientas (herramientas.php)

Este archivo debe ser alojado en la carpeta **include**, contiene varios métodos que servirán al programador para realizar seguimiento a cerca del estado de las variables utilizadas en los procesos en determinado momento de ejecución de los programas.

Métodos: `write_to_console($Nombre_variable)` y `write_estado_var($Nombre_variable)`

También incluye otras herramientas que en el momento no se comentarán en la presente guía, pero que serán utilizadas en el desarrollo de los talleres de programación orientada a objetos con PHP.

```
<?php
//--- Esta función permite escribir en consola cualquier tipo de dato.
// Incluyendo el contenido completo de los diferentes tipos de matrices.
// En la primera línea escribe el nombre de la variable y luego el contenido.
function write_to_console($data) {

    $console = 'console.log("' . getVariableName($data) . ' (' . gettype($data) . ')");';
    $console = sprintf('<script>%s</script>', $console);
    echo $console;

    $console = 'console.log(' . json_encode($data) . ')';
    $console = sprintf('<script>%s</script>', $console);
    echo $console;
}

// Función que retorna el nombre de la variable
function getVariableName($var) {
    foreach($GLOBALS as $varName => $value) {
        if ($value === $var) {
            return $varName;
        }
    }
    return;
}
?>
```

3.3. Contenido del archivo de conexión con la Base de Datos (conect.php)

Este archivo debe ser alojado en la carpeta **include**, contiene los métodos o funciones que permiten establecer la conexión con la base de datos y también los encargados de inserción, modificación y borrado de contenidos almacenados en las tablas de la base de datos.

```
<?php
// Archivo Componente:  conect.php
// Proyecto:  Página Web  Mi hoja de vida personal
// Este archivo contiene la librería de funciones mySQL para conectar con
// la base de datos y para realizar operaciones sobre las diferentes tablas
// que la conforman..

function Conectarse( $servidor , $usuario , $password , $basedatos ) {

    // --- aquí se realiza el intento de conexión con los datos suministrados
    if (!($link = mysqli_connect($servidor, $usuario, $contrasena))) {
        echo "Error al conectar con el servidor de base de datos.<br/>";
        // --- En caso de error el proceso termina forzosamente.
        exit();
    }
    if (!mysqli_select_db($link, $basedatos))
    {
        echo "Error al conectar al acceder a la BD, (no existe la BD $basedatos).<br/>";
        // --- En caso de error  el proceso termina forzosamente.
        exit();
    }
    else {
        // --- si la conexión fue satisfactoria, se configura el CharSet UTF-8
        if (!mysqli_set_charset($link, "utf8"))
        {
            printf("Error cargando el conjunto de caracteres utf8: %s\n", mysqli_error($link));
            exit();
        }
    }
    return $link;
}

?>
```

4. Publicación de contenidos almacenados en la Base de Datos:

Como se mencionó previamente al momento de identificar la estructura del registro que conforma los datos de cada ESTUDIO o CURSO realizado, se utilizará un PROMATO o PLANTILLA básica que permitirá visualizar los SEIS (6) datos almacenados por cada registro de estudio.

A continuación analicemos la estructura del formato que se espera obtener como salida del proceso.

<i>Tipo de estudio cursado</i> TECNOLÓGÍA	<i>Nombre</i> Análisis y Desarrollo de Software	
<i>Institución educativa</i> SERVICIO NACIONAL DE APRENDIZAJE		
<i>Ciudad</i> VILLETA (Cundinamarca)	<i>Fecha de finalización</i> 2022/12/16	<i>Horas cursadas</i> 3330

Recordemos que previamente se debió ejecutar los procesos de alistamiento y carga de información en la base de datos.

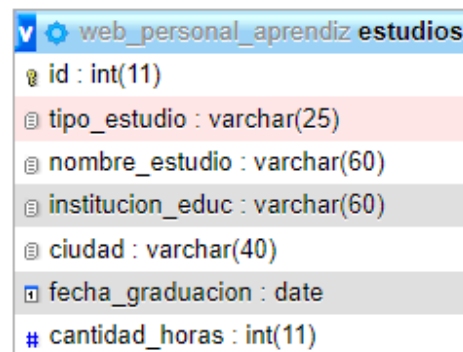
Entonces se deben conocer los nombres de los datos almacenados, nombres de las tablas, nombre de la base de datos y también los datos para conexión con el servidor.

Información de los datos:

Base de datos: **web_personal_APRENDIZ**

Nombre de la tabla: **estudios**

Estructura de datos:



web_personal_aprendiz estudios
id : int(11)
tipo_estudio : varchar(25)
nombre_estudio : varchar(60)
institucion_educ : varchar(60)
ciudad : varchar(40)
fecha_graduacion : date
cantidad_horas : int(11)

El proceso de extracción de los datos se realizará mediante la ejecución de una instrucción SELECT sobre la tabla estudios. En este caso se tomarán todos los registros sin ningún tipo de filtro y sin ningún orden específico, solamente el orden en que hayan guardados en la tabla. En pantalla se publicarán todos los datos, excepto el campo id, aunque este deberá ser extraído para su posterior procesamiento al igual que todos los otros campos. El script SQL de esta consulta será el siguiente:

```
$ScriptSQL = "SELECT * FROM estudios";
```

Este script será entregado a una de las funciones de PHP que realizan interface con la base de datos MySQL y se encarga de tomar la información según el requerimiento allí informado.

4.1. Consideraciones iniciales para la interface con la Base de datos MySQL

El lenguaje PHP permite conectar con diferentes tipos de bases de datos y para ello utiliza un conjunto de funciones especialmente construidas para gestionar el tratamiento de los datos y realizar con ellos las operaciones necesarias para la escritura, consulta, modificación y borrado, operaciones comúnmente conocidas con la sigla en inglés CRUD, Create, Read, Update, Delete. A continuación haremos referencia a las principales funciones que PHP incorpora para su interface con MySQL.

- ✓ Conectar con el servidor de bases de datos:
mysqli_connect().- Devuelve un identificador de la conexión en caso de éxito, que servirá de enlace o Link para las operaciones que se realizarán sobre ella. También puede devolver false en caso de alguna falla en la conexión.
- ✓ Seleccionar una base de datos en la conexión activa:
mysqli_select_db().- Devuelve true en caso de éxito y false en caso de error, por ejemplo, que no exista la base de datos, por nombre errado o alguna otra inconsistencia.
- ✓ Enviar o ejecutar una instrucción SQL a la base de datos:
mysqli_query().- Devuelve un identificador o true (dependiendo de la instrucción) cuando la instrucción SQL (Query) se ejecuta correctamente. Retorna false en caso contrario.
- ✓ Obtener y procesar los resultantes de una consulta, descomponiéndolos en registros u objetos independientes para que puedan ser tratados por el programa:
 - **mysqli_num_rows()** Devuelve el número de filas afectadas por una operación o resultantes de una consulta.
 - **mysqli_fetch_array()** Permite obtener en forma separada una a una cada fila o registro resultante de una consulta. Habitualmente se utiliza dentro de una instrucción While(), para que en cada iteración se obtenga la fila o registro siguiente. Por ejemplo,

```
while($fila = mysqli_fetch_array($DatosEstudio))
```

En este caso la variable \$fila recibirá un Vector o Array de tipo Asociativo con el listado de datos correspondientes al conjunto de campos que conforman la fila o registro.
- ✓ Configurar el juego de caracteres (Char SET) que MySQL debe utilizar al momento de entregar la información, en este caso para Latinoamérica utilizamos UTF8.
 - **mysqli_set_charset()** Esta configuración regularmente se realizarse en el momento de confirmar la correcta conexión y selección de la Base de datos, informando el \$link resultante.

```
mysqli_set_charset($link, "utf8")
```
- ✓ Cerrar la conexión con el servidor de bases de datos y descarga o liberación de memoria:
 - **mysqli_free_result()** Permite liberar de la memoria el conjunto de datos resultante de una consulta. Siempre es aconsejable hacerlo después de finalizar los procesos. Entendiendo que los datos ya han sido publicados, tratados o trasladados hacia otras variables.
 - **mysqli_close()** Esta operación es necesaria para liberar la conexión establecida entre el programa con el motor de la Base de datos. Siempre se hace después de haber obtenido la información y cuando ya no se necesite utilizar más dicho enlace.

4.2. Extracción de los datos de la Base de datos MySQL

Para la extracción de los datos, se debe seguir un orden de pasos, así:

- 1) Importar las librerías o componentes extras necesarios para el proceso. En el presente ejercicio se utilizarán los componentes externos **conect.php** y **herramientas.php**. Recordar que estos se encuentran alojados en la carpeta **include**.

```
<?php

// --- Importamos el componente <header> definido en el archivo cabecera.php
include("header.php");

// --- Importamos el componente <nav> definido en el archivo menu.php
include("nav.php");

// --- Importamos los métodos de ayuda para seguimiento y otras herramientas
//       de conversión de datos
include("../include/herramientas.php");

// --- Importamos los metodos encargados de la conexión y gestión con la base de datos
include("../include/conect.php");

?>
```

- 2) Establecer datos de conexión (nombre de servidor, usuario, password, base de datos).
- 3) Ejecutar el procedimiento de conexión. Si es exitoso, el sistema devolverá un OBJETO APUNTADOR que identifica la conexión activa con la base de datos solicitada. En caso contrario el sistema suspenderá la carga de la página mostrando el error correspondiente.

Para la ejecución de estos dos primeros pasos, se utilizará el **método Conectarse()**. Este deberá ser importado desde el componente externo **"conect.php"**.

```
// -- Lanzamos la conexión con la base de datos
//     para este ejemplo se asumirá que los datos están correctos, sin embargo
//     es necesario implementar una mejora posterior que controle los posibles
//     errores de conexión
$linkBD=Conectarse("localhost" , "root" , "root" , "web_personal_APRENDIZ");
```

- 4) Realizada la conexión se lanza el proceso de extracción informando el script SQL que deberá ser procesado por el motor de la B.D.

```
// -- Ejecutamos el QUERY (consulta) para extraer la información requerida
$ScriptSQL = "SELECT * FROM estudios";
```

- 5) En caso de ser exitoso el proceso, el sistema devuelve el conjunto de datos resultantes en un objeto especial tipo matriz asociativa, en la cual, cada columna es identificada por el nombre de cada campo. Este objeto matriz la identificaremos con el nombre **\$DatosEstudio**

```
if ($DatosEstudio = mysqli_query($LinkBD, $ScriptSQL )) {  
  
    // -- Iniciamos la publicación de los datos, mostrando la cabecera de la tabla.  
    //    en este caso, un solo renglón con el título "Estudios Realizados"  
    echo "<table width='700' border='1' class='tabla' align='center'>";  
    echo "<tr><td colspan='3' class='titulotabla'>Estudios Realizados</td></tr>";  
    echo "<tr class='titulotabla'></tr></table>";  
    echo "<br>";
```

Hasta aquí el proceso se realiza una sola vez.

- 6) A partir de este paso, se repite el proceso en un ciclo **While()** hasta que se hayan procesado todos los registros resultantes de la consulta. Para ello, dentro de validador del While invocamos la función **mysqli_fetch_array(\$DatosEstudio)**. Esta función extraerá uno a uno hasta completar todos los registros. Por cada ciclo, la información del registro deberá ser volcada en la plantilla prediseñada, haciendo uso de los identificadores asociativos asignados a cada campo. Las líneas de código PHP para este proceso serán las siguientes:

```
// -- inicio ciclo repetitivo que recorre la matriz de $DatosEstudio procesando  
//    línea por línea los datos serán distribuidos en las diferentes celdas  
//    de la tabla (plantilla prediseñada)  
while($fila = mysqli_fetch_array($DatosEstudio)){  
    write_to_console($fila);  
    // -- write_estado_var($fila);  
    // -- analizamos si el estudio a mostrar tiene horas reportadas.  
    //    esto para evaluar si se debe mostrar o no este campo en la tabla  
    if (is_null($fila['cantidad_horas'])) {  
        $Horas = 0;  
        $HorasColspan = "colspan='2'";  
    }  
    else {  
        $Horas = $fila['cantidad_horas'];  
        $HorasColspan = "";  
    }  
    // -- A continuación renderizamos la plantilla de ESTUDIOS con la información  
    //    de cada registro...
```

```

        echo "<table width='700' border='1' class='tabla' align='center' >";
        echo "<tr class='listado'>";
        echo "<td><div class='labelTab1'>Tipo de estudio cursado<br></div>" .
$fila['tipo_estudio'] . "</td>";
        echo "<td colspan='2' ><div class='labelTab1'>Nombre<br></div>" .
$fila['nombre_estudio'] . "</td>";
        echo "</tr>";
        echo "<td colspan='3' ><div class='labelTab1'>Institución Educativa<br></div>" .
$fila['institucion_educ'] . "</td>";
        echo "</tr>";
        echo "<td $HorasColspan><div class='labelTab1'>Ciudad<br></div>" .
$fila['ciudad'] . "</td>";
        echo "<td><div class='labelTab1'>Fecha de finalización<br></div>" .
$fila['fecha_graduacion'] . "</td>";
        if ($Horas>0) echo "<td><div class='labelTab1'>Horas<br></div>$Horas</td>";
        echo "</tr>";
        echo "</table><br>";
    } // --- aquí finaliza el ciclo while
    // -- liberar el conjunto de resultados
    mysqli_free_result($DatosEstudio);

}
else {
    printf("Hubo errores al leer los datos");
}

    //Cerrar la conexión con el servidor de bases de datos:
    mysqli_close($LinkBD);

?>

</main> <!-- Aquí termina la definición del cuerpo principal del módulo
funcional -->

<?php
    // --- Importamos el componente <footer> definido en el archivo cabecera.php
    include("footer.php");
?>

</body>
</html>

```

- 7) En la parte final de la estructura del código es importante identificar los cierres de los validadores de la conexión.

También se deben ordenar (ejecutar) los procesos para relacionados con el cierre de la base de datos y del objeto que contiene toda de resultados.

- 8) El FRONT del sitio WEB, una vez verificada la codificación y realizadas las correcciones a que hubiese lugar, tendrá la siguiente apariencia.



Hoja de Vida

Uriel Castañeda Sierra

[Inicio](#)
[Mis Estudios](#)
[Mi Experiencia](#)
[Contacto](#)

Estudios Realizados			
Tipo de estudio cursado		Nombre	
Posgrado		Especialización en Negocios en Internet	
Institución Educativa			
Universidad ICESI - Universidad de la Sabana			
Ciudad		Fecha de finalización	
Chia		2004-10-04	
Tipo de estudio cursado		Nombre	
Especialización técnica		Especialización en Gerencia de Proyectos	
Institución Educativa			
SENA			
Ciudad		Fecha de finalización	Horas
Villeta		2013-05-05	550
Tipo de estudio cursado		Nombre	
Profesional		Ingeniería de Sistemas	
Institución Educativa			
Universidad Católica de Colombia			
Ciudad		Fecha de finalización	
Bogotá		1998-03-28	
Tipo de estudio cursado		Nombre	
Bachillerato técnico		Bachiller Técnico en Mecánica Industrial	
Institución Educativa			
Instituto técnico central - La Salle			
Ciudad		Fecha de finalización	
Bogotá		1988-12-06	
Tipo de estudio cursado		Nombre	
Diplomado		Gerencia de Proyectos bajo lineamientos PMI	
Institución Educativa			
Universidad Piloto			
Ciudad		Fecha de finalización	
Bogotá		2016-12-10	

Power by: SENA - CDAE, Villeta Colombia 2022 - By Uriel C.S.



5. Actividades de evaluación:

Con el fin de reforzar su aprendizaje, a continuación encontrará el listado de instrucciones que deberá seguir en el desarrollo de sus ejercicios de práctica y entrega de evidencias.

- 1) Incorporar los diferentes componentes elaborados en la versión de su página web Estática, adaptándolos a la estructura establecida en la guía de maquetación y según el modelo expuesto en la presente guía. La nueva versión de su página WEB deberá tener como mínimo los siguientes componentes:
 - Home page (index.php)
 - Mis Estudios (misestudios.php) con contenido dinámico
 - Mi Experiencia (miexperiencia.php) con contenido dinámico
 - Glosario (glosario.php) con contenido dinámico
 - Contacto (contacto.php) con contenido dinámico(*)
- 2) Incorporar mejoras de estilo al modelo presentado en la página de “MisEstudios”
- 3) Diseñar la estructura de las tablas faltantes y elaborar los Scripts para construcción y carga de contenidos.
- 4) Realizar las consultas necesarias para alimentar los componentes dinámicos de los diferentes módulos funcionales del sitio web.
- 5) (*) Investigar los procedimientos para INSERTAR datos en una tabla MySQL, diseñar la estructura de la tabla de información de CONTÁCTENOS. Elaborar los programas necesarios para habilitar el botón guardar del formulario de contáctenos, de tal forma que este lance el proceso para grabar la información capturada en la tabla correspondientes.
- 6) Investigar y ejecutar los procesos necesarios para instalar la base de datos en el servidor remoto que oficia como hosting de su sitio web. Realizar las adaptaciones necesarias en su página web para que el sistema enlace la B.D remota y no la BD local.
- 7) Crear el repositorio en GitHub y subir el proyecto actualizado
- 8) Publicar en el Hosting la nueva versión de su sitio web dinámico.
- 9) Incorporar los demás componentes estáticos no intervenidos, de la versión antigua a la nueva versión de su página web.
- 10) Su página WEB finalizada deberá cumplir con los requisitos indicados en la siguiente lista de chequeo.

Lista de chequeo revisión página WEB

ítem	Aspecto a verificar y evaluar...	% pond.
1	La página WEB fue elaborada utilizando HTML y CSS de acuerdo con las instrucciones entregadas en la Guía y taller Números 1, 2 y 10 de desarrollo WEB	10%
2	La página WEB muestra el ícono y el título correcto en la pestaña del navegador	5%
3	La estructura de la página WEB define los 4 componentes (Cabecera, navegación, cuerpo principal y pie de página)	5%
4	La cabecera de la página muestra correctamente la fotografía, título y subtítulo de la página.	5%
5	El menú de navegación es funcional (enlaza correctamente los 4 archivos html) y ofrece efectos de visualización configurados con estilos CSS.	5%
6	El cuerpo principal muestra la utilización de SECCIONES, ARTÍCULOS y MENÚ de navegación lateral (ASIDE) con los links de redes sociales	5%
7	La página WEB muestra en la línea de pie de página información relacionada con autor y año de actualización (Pie de página)	5%
8	Los componentes de CABECERA, MENU navegación horizontal y PIE DE PÁGINA son exactamente iguales en los 4 componentes (Home, mis estudios, mi experiencia y Contáctenos)	5%
9	La URL suministrada (con el dominio del sitio de instalación de su página WEB) funciona correctamente.	10%
10	La página Web muestra mínimo 3 imágenes y estas fueron debidamente procesadas configurando su tamaño de tal forma que al ser publicadas no hayan sido alteradas en tamaño +/- 50%	10%
11	El formulario de contáctenos presenta el uso de mínimo 5 tipos de componentes de captura de información (cuadros de texto, selectores, cajas de chequeo, botones de radio, áreas de texto, entre otros).	10%
12	La conexión con la Base de datos Remota (Servidor MySQL del hosting) se realiza correctamente y muestra los datos previamente cargados en la Base de datos	15%
13	El diseño de los componentes del módulo funcional "MisEstudios" permite ver varios ítems y estos coinciden con la información registrada en las tablas de la base de datos	10%
14	Los componentes adicionales, El Glosario está correctamente enlazado con la base de datos y construido con estructura dinámica, de tal forma que replique la cantidad de veces necesarias según la cantidad de registros almacenados o consultados en el glosario	10%

Los % se evaluarán % definido en cada ítem si cumple ó 0% si no cumple. No se emiten valores parciales o intermedios.

El % total a evaluar es del 100% Esta evidencia se aprobará con puntaje mínimo de 70%



Sitios WEB recomendados:

Paleta de Colores: <https://html-color-codes.info/codigos-de-colores-hexadecimales/>

Diseño de una página web básica:

<https://www.w3.org/Style/Examples/011/firstcss.es.html>

Animación en CSS3: after, before y hover

<http://www.cristalab.com/tutoriales/animacion-en-css3-after-before-y-hover-c113195/>

12 efectos css en una imagen con el evento “hover”

<https://blog.interactius.com/12-efectos-css-en-una-imagen-con-el-evento-hover-5026b5cf1983>

Configurar la base de datos MySQL o MariaDB, las tablas y los campos de las tablas a utf8

<https://proyectoa.com/solucion-al-error-al-mostrar-tildes-acentos-enes-php-mysql-mariadb/>

Bibliografía:

J.D Gauchat, El gran libro de HTML5, CSS3 y JavaScript, 3ª Edición, Editorial MARCOMBO S.A. Barcelona 2017. ISBN: 978-84-267-2463-2