PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- **Denominación del Programa de Formación:** TGO Análisis y Desarrollo de Sistemas de Información
- **Código del Programa de Formación:** 228106 V102
- **Nombre del Proyecto:** Construcción de un sistema de información que cumpla con los requerimientos del cliente en procesos que se lleven a cabo en el sector productivo del departamento de Caldas
- **Fase del Proyecto:** IMPLEMENTACIÓN
- **Actividad de Proyecto:** Seleccionar la alternativa de solución que cumpla con los requerimientos establecidos por el cliente
- **Competencia:** Construir el sistema que cumpla con los requisitos de la solución informática.
- **Resultados de Aprendizaje Alcanzar:** Realizar la codificación de los módulos del sistema y el programa principal, a partir de la utilización del lenguaje de programación seleccionado, de acuerdo con las especificaciones del diseño
- **Duración de la Guía:** 40 horas

2. PRESENTACIÓN

*PHP e*s un lenguaje de programación del lado del servidor y de alto nivel, embebido en páginas HTML. Ha tenido una gran aceptación en la comunidad de desarrolladores, debido a la potencia y simplicidad que lo caracterizan, así como al soporte generalizado en la mayoría de los servidores de hosting, hasta los más simples y económicos.

Otra de las claves del éxito de PHP es que la mayoría de los CMS más populares (WordPress, Joomla!, Drupal) y los sistemas de comercio electrónico (Prestashop, Woocommerce, Magento), así como otros cientos de herramientas, están desarrollados en PHP. Por lo tanto, usar PHP es sinónimo de ser capaz de introducirnos en muchas herramientas gratuitas y de código abierto para realizar cualquier cosa en el ámbito de la web.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

- **Descripción de la(s) Actividad(es):**
    - **Actividades de aprendizaje:**
        - Nueva sintaxis de PHP 8

- Dominar la creación y uso de funciones en PHP para modularizar el código y reutilizarlo.
- Familiarizarse y utilizar variables superglobales como $_GET, $_POST y $_SESSION
- Crear formularios HTML y procesar los datos enviados por los usuarios utilizando PHP 8
- Conexión y manipulación de bases de datos MySQL incluyendo la ejecución de consultas SQL y la protección contra ataques de inyección SQL
- Comprender los conceptos de POO en PHP, como clases, objetos, propiedades y métodos
- Manejo de errores y excepciones de manera efectiva en PHP para depurar y mejorar la robustez del código
- Implementar sistemas de autenticación de usuarios y control de acceso en aplicaciones web PHP 8.

- **Actividad de Reflexión inicial**

Actividad de reflexión inicial

Antes de comenzar nuestro proyecto en PHP 8, los aprendices deberán conocer la evolución de PHP. Puede parecer que no, pero PHP ha cambiado muchísimo en este tiempo. Vamos a ver cómo era construir una clase con PHP 5.3 y cómo lo es con PHP 8.2:

De PHP 5.3 a PHP 8.2
https://www.youtube.com/watch?v=BHAYO6esXlw

- **Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje**

Manejo de sesiones en el Proyecto Formativo

Las sesiones permiten mantener y manejar información de los usuarios en el servidor mediante el array `$_SESSION`

`$_SESSION` es un array especial utilizado para guardar información a través de los requests que un usuario hace durante su visita a un sitio web o aplicación. La información guardada en una sesión puede llamarse en cualquier momento mientras la sesión esté abierta.

A cada usuario que accede a la aplicación e inicia sesión se le asigna un session ID único, y es lo que le permite identificar la sesión y que esté disponible para ese usuario en concreto. La forma más segura de manejar sesiones es almacenando en el cliente sólo esta session ID, y cualquier información de la sesión guardarla en el lado del servidor.

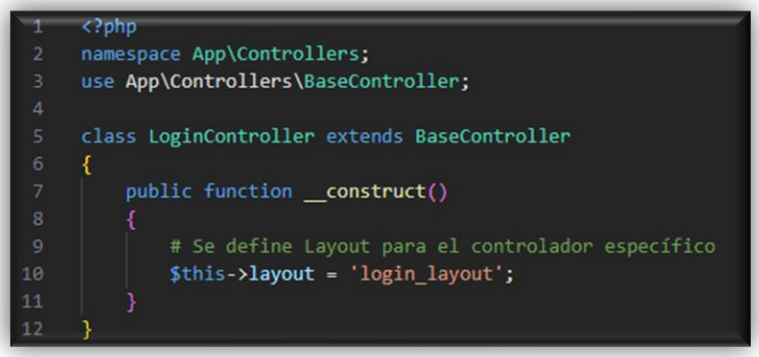
Iniciar una sesión

Antes de guardar cualquier información en una sesión es necesario iniciar el manejo de la sesión (session handling). Esto se hace al principio del código PHP, y debe hacerse antes de que cualquier texto, HTML o JavaScript se envíe al navegador. Para comenzar la sesión tan sólo hay que utilizar la función `session_start()`

Para inicializar y destruir las variables de sesión de los usuarios logueados debemos crear un nuevo controlador.

Actividad 01

Los aprendices deberán abrir el proyecto **Senamás** en la carpeta `app/controllers` y crear un nuevo controlador llamado: `LoginController.php`, el cual contiene los métodos initLogin y logout , para manejo de la autenticación de usuarios



```php
<?php
namespace App\Controllers;
use App\Controllers\BaseController;

class LoginController extends BaseController
{
    public function __construct()
    {
        # Se define Layout para el controlador específico
        $this->layout = 'login_layout';
    }
}
```

Acciones del controlador LoginController:

initLogin: En esta acción se recibe por POST los campos usuario y contraseña enviados de la vista login. Se crean las validaciones requeridas para garantizar que los campos del formulario no sean vacíos y se hace el llamado a la función validarLogin del modelo. En caso de que la función validar login devuelva un false, lo redireccionará a la vista Login, con un mensaje de error.

```php
public function initLogin()
{
    if (isset($_POST['txtUser']) && isset($_POST['txtPassword'])) {
        $user = trim($_POST['txtUser']) ?? '';
        $password = trim($_POST['txtPassword']) ?? '';
        $errors = '';
        if ($user == '' || $password == '') {
            $errors = "El usuario y/o contraseña no pueden ser vacios";
            $data = [
                "errors" => $errors
            ];
            $this->render("/login/login", $data);
        } else {
            $usrObj = new UserModel(null, $user, $password );
            if($usrObj->validarLogin()){
                header("Location:/pacientes/index");
            }else{
                $errors = "El usuario o contraseña son incorrectos";
                $data = [
                    "errors" => $errors
                ];
                $this->render("/login/login", $data);
            }
        }
    }else{
        $this->render("/login/login");
    }
}
```

Logout: tan importante es iniciar una sesión como cerrarla. Incluso cuando realmente una sesión es sólo una forma temporal de almacenar datos, es importante limpiarla después para asegurar la máxima seguridad especialmente cuando se trata con información personal.

- Para eliminar un valor de sesión simplemente se utiliza **unset():**
  `unset($_SESSION["usuario"]);`
- Para desvincular todos los valores de sesión de una vez se puede emplear `session_unset()`
  `session_unset();`

Ambas acciones anteriores sólo afectan a los valores guardados en la sesión, no a la propia sesión. Todavía se pueden guardar nuevos valores en `$_SESSION`. Si se quiere dejar de utilizar por completo la sesión, por ejemplo cuando el usuario hace log out, se utiliza `session_destroy()`.

Es muy recomendable que cuando ya no se necesite la sesión se destruya con `session_destroy()`, en vez de desvincular el valor de sus valores con session_unset(). Si sólo

se desvinculan los valores la sesión permanecerá activa, lo que deja la puerta abierta a intrusos.
Ahora implementaremos la acción logout del controller:

```php
public function logout()
{
    // Destruir todas las sesiones activas
    session_destroy();
    header("Location:/login/init");
}
```

Como se mencionó anteriormente para hacer uso de las variables de sesión se debe iniciar el manejo de la sesión por medio de `sessión_start().`

La función `sessión_start()` inicia la sesión entre el usuario y el servidor, y permite a los valores guardados en `$_SESSION` ser accesibles después. Como todo va a ser controlado a través de acciones de los controladores, se inicia la sesión en controlador base, ya que todos heredan de él y de esta forma garantizamos que cada vez que se ejecute una acción de *x* controlador se ejecutará el inicio de sesión permitiendo mantener la sesión a lo largo del tiempo de vida de la aplicación y por lo tanto acceder a las variables creadas en el controlador login.

Actividad 02

Los aprendices deberán abrir el proyecto formativo en la carpeta `app/controllers` e iniciar la sesión en el archivo `BaseController.php`.

```php
<?php
namespace App\Controllers;
# Comienzo de la sesión
session_start();

class BaseController
{
    protected string $layout = 'admin_layout';
```

Actividad 03

Ahora los aprendices deberán implementar la función `validarLogin()` en el modelo que está asociado a los usuarios del sistema.

Para este ejemplo se incluirá la función `validarLogin` en el modelo `UserModel.php`. Esta función permite consultar en la base de datos la información del usuario y validar si la información ingresada en el formulario de login es correcta.

Se crearán 3 variables de sesión:

- `$_SESSION['tipo_usuario]`: Servirá para saber qué tipo de usuario está logueado en el sistema y a que acciones tendrá derecho.
- `$_SESSION['nombre']`: Servirá para mostrar en el menú principal quien está logueado actualmente en el sistema.
- `$_SESSION['id_usuario']`: Servirá para saber el ID del usuario que está logueado en el sistema y poder realizar acciones con el mismo.
- `$_SESSION['timeout']`: Servirá para almacenar la fecha en que el usuario inicia la sesión.

```php
public function validarLogin()
{
    try {
        $sql = $this->dbConnection->prepare("SELECT * FROM usuario WHERE usuario = ?");
        #$user = $this->usuario;
        $sql->bindParam(1, $this->usuario);
        # Ejecutamos
        $sql->execute();
        $resultSet = [];
        # Ahora vamos a indicar el fetch mode cuando llamamos a fetch
        while($row = $sql->fetch(\PDO::FETCH_OBJ)){
            $resultSet[] = $row;
        }
        # Si encuentra el usuario
        if(count($resultSet) >  0){
            //$option = ['cost' => 12];
            //$contra = password_hash($this->password, PASSWORD_BCRYPT, $option);
            $hash = $resultSet[0]->password;
            if(password_verify( $this->password, $hash)){
                $_SESSION['tipo_usuario'] = "paciente";
                $_SESSION['nombre'] = $resultSet[0]->usuario;
                $_SESSION['documento'] = $resultSet[0]->password;
                $_SESSION['timeout'] = time();
                session_regenerate_id();
                return true;
            }
        }
        return false;
    } catch (PDOException $ex) {
        echo $ex->getMessage();
        die();
    }
}
```

Sesión Time-Outs. Establecer un tiempo de vida a las sesiones es algo importante para trabajar con las sesiones de usuarios en aplicaciones. Si un usuario inicia sesión en un lugar y se olvida de cerrarla, otros pueden continuar con la misma, por eso es mejor establecer un tiempo máximo de sesión.

Actividad 04

Los aprendices definirán una nueva variable global en el proyecto de formación llamada `INACTIVE_TIME`. Para ellos se deberá modificar el archivo: `config/global.php`

```php
<?php
define('MAIN_APP_ROUTE',__DIR__."/../app/");

# Tiempo inactividad dada en minutos
define('INACTIVE_TIME', 1);

# más constantes de configuración
```

En la función `validarLogin()` del modelo se creó una variable de sesión llamada `timeout`, en la que se almacena la fecha en que el usuario inicia la sesión.

Posteriormente en el constructor de la clase controlador Base se realiza la comprobación del tiempo de actividad del usuario, y si este tiempo se supera, se hace la redirección a la página de logout.

Actividad 05

Modificar el archivo `core/BaseController.php` e incluir en el constructor la validación de time out.

```php
class BaseController
{
    protected string $layout = 'admin_layout';
    # Método para renderizar una vista con los datos

    public function __construct()
    {
        /* Validar que el tiempo de inactividad del usuario no supere el tiempo
            definido en la variable global INACTIVE_TIME  */
        if(isset($_SESSION['timeout'])){
            # Calcular el tiempo de vida de la sesión
            $tiempoSesion = time() - $_SESSION['timeout'];
            if($tiempoSesion > (INACTIVE_TIME * 60)){
                session_destroy();
                header("Location:/login/logout");
            }else{
                $_SESSION['timeout'] = time();
            }
        }
    }
```

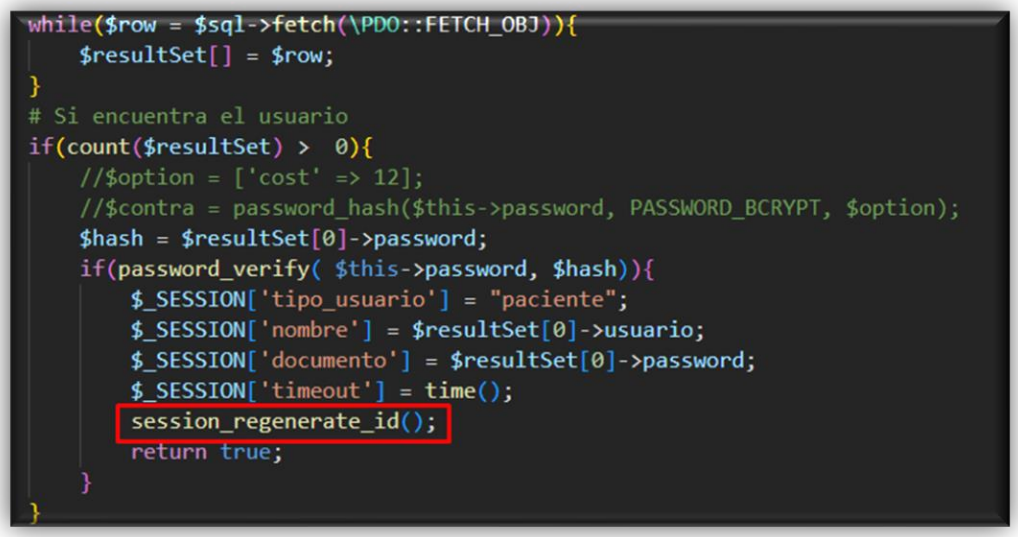
Se utiliza el constructor de `BaseController` ya que cada que un usuario hace un llamado a una acción desde el menú se invoca a este constructor y en ese momento se comprueba que aun esté dentro del tiempo permitido de inactividad, y si este tiempo aún no se ha superado, se actualiza el valor de la variable con el tiempo actual.

El código asegura que, si no hay ninguna actividad en x minutos, cualquier **request** en adelante redirigirá a la página de `logout`.

También debemos ajustar nuestros controladores para que el constructor del controlador, llame al constructo del papá, como se visualiza en el siguiente código:

Otra buena práctica recomendada para garantizar la seguridad en las sesiones es **regenerar el Session ID**. La función `session_regenerate_id()` crea un nuevo ID único para representar la sesión actual del usuario. Esto se debe realizar cuando se realizan acciones importantes como loguearse o modificar los datos del usuario. Darle a las sesiones un nuevo ID reduce la probabilidad de ataques session hijacking.

Como se puede observar en el modelo implementado anteriormente, en la función `validarLogin()` ya se implementó la función `sesión_regenerate_id()`:



Otra recomendación es utilizar almacenamiento permanente. Utilizar una base de datos para almacenar los datos que se cree que serán persistentes. Dejarlos en la sesión por demasiado tiempo abre de nuevo

puertas a ataques. Depende del desarrollador decidir qué datos serán almacenados o se mantendrán en `$_SESSION`.

Actividad 06

Se deben modificar el **BaseController** y cada uno de las acciones de los controladores que se desean restringir para usuarios registrados, como se indica a continuación

Para garantizar que cada acción de un controlador sea accedida solo por usuarios autorizados (si es una acción permitida solo para usuarios registrados en el sistema), se debe realizar la verificación de que exista una variable de sesión y que el tipo de usuario logueado tenga permisos para esta.

Para esto se crea en la acción de cada uno de los controladores, el siguiente código que permita validar que exista un usuario logueado y que esté autorizado para acceder a la acción, en caso contrario se redirige a la vista Login. En este código, se restringen cada una de las acciones para los usuarios que puedan realizar dicha acción, indicando por medio de un array cuáles son los id de tipo de usuario que tienen acceso a la misma.

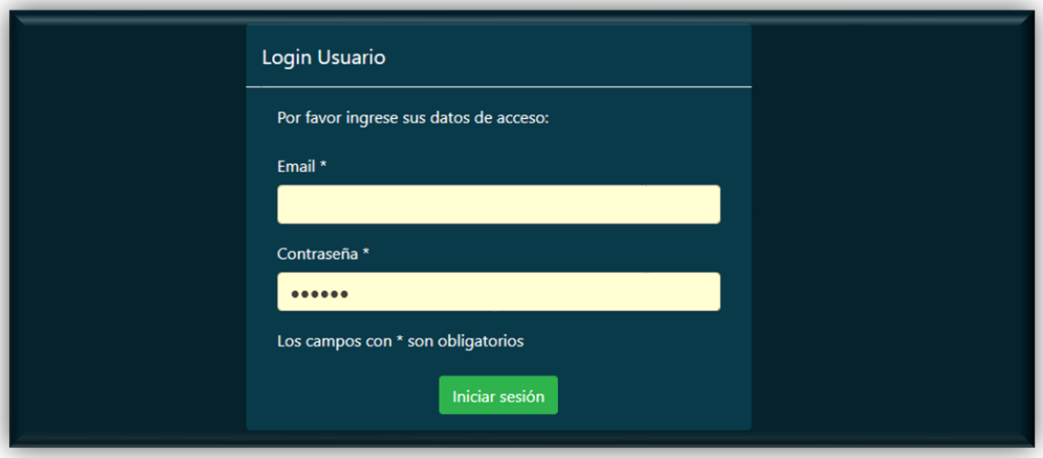
```php
public function index(){
    /* Cada acción del controlador deberá tener la validación de tipo de usuario
       loggueado para verificar si el usuario tiene derecho a realizar la acción
    */
    if(!isset($_SESSION['tipo_usuario'])){ # Se verifica que la sesión de tipo de usuario exista
        # Si no es así, será redireccionado al login
        header("Location:/login/init");
    }else{
        if(!in_array($_SESSION['tipo_usuario'], ['paciente','admin'])){
            header("Location:/login/init");
        }
    }

    # Se crea objeto del modelo Paciente
    $paciente = new PacienteModel();
    # Se obtienen todos los pacientes en un array
    $pacientes =  $paciente->getAll();
    $data = [
        "pacientes" => $pacientes,
        "cant_pacientes" => count($pacientes)
    ];
    $this->render("/paciente/index", $data);
}
```

Actividad 07

Los aprendices crearán las vistas login y error para manejo de autenticación de usuarios como se muestra a continuación.

Se crea el archivo `loginView.php` dentro de la carpeta de view/login, el cual debe tener un aspecto similar al de la siguiente imagen:



Se debe tener en cuenta que en el html de la vista login, se debe crear un div que muestre los errores que se devuelven desde la acción login del controlador:

```
<div class="form-group-error">
    <p><?php echo (isset($errors)) ? $errors : "";?></p>
</div>
```

Los aprendices deberán personalizar la página de manera que sea agradable al usuario.

Actividad 08

Modificar el archivo `views/layouts/admin_layout.php`, para ocultar los menús restringidos solo para usuarios registrados por medio de la validación de la existencia de la variable de sesión

```
<header class="header">
    <h1>SENAmásEPS</h1>
    <ul>
        <li><a href="#">Nosotros</a></li>
        <?php
            # Validación para que solamente usuarios tipo "admin" vean cierto menús
            if(isset($_SESSION['tipo_usuario']) && $_SESSION['tipo_usuario'] == "admin"):
        ?>
        <li><a href="#">Documentación</a></li>
        <?php endif ?>
        <li><a href="#">Politicas Internas</a></li>
        <li><a href="#">Acerca de</a></li>
        <?php
        if(isset($_SESSION['tipo_usuario'])){
            echo "<li><a href='/login/logout'>Cerrar sesión {$_SESSION['nombre']}</a></li>";
        }
        ?>
    </ul>
</header>
```

Se incluye además un nuevo ítem de menú para que el usuario tenga la opción de desloguearse, mostrándole además el nombre que está almacenado en la variable de sesión:

```php
<header class="header">
    <h1>SENAmásEPS</h1>
    <ul>
        <li><a href="#">Nosotros</a></li>
        <li><a href="#">Documentación</a></li>
        <li><a href="#">Politicas Internas</a></li>
        <li><a href="#">Acerca de</a></li>
        <?php
        if(isset($_SESSION['tipo_usuario'])){
            echo "<li><a href='/login/logout'>Cerrar sesión {$_SESSION['nombre']}</a></li>";
        }
        ?>
    </ul>
</header>
```

- **Actividades de transferencia del conocimiento**
  Los aprendices deberán demostrar los conocimientos conceptuales y procedimentales que adquirieron a partir de las actividades de apropiación mediante el siguiente instrumento de evaluación:

Los aprendices deberán realizar cada una de las actividades propuestas para adquirir dominio en los conocimientos relacionados con los diferentes temas. Se evaluarán los conocimientos adquiridos mediante instrumentos de evaluación de Desempeño y Producto relacionados en la presente guía

- **Ambiente Requerido**
  - Ambiente de SISTEMAS con conexión eléctrica e internet
- **Materiales**
  - Computadores (30)
  - Sillas (3)
  - Televisor (1)

- Resma tamaño carta (1)
- Marcadores (3)
- Lápiz (1)
- Lapicero (1)

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento : **Evidencias de Desempeño** **Asistencia y participación activa en las diferentes actividades propuestasz** **Evidencias de Producto:**	Crea la base de datos en el motor de base de datos seleccionado, siguiendo especificaciones técnicas del informe, según normas y protocolos de la empresa.	**Observación:** EXC_D_01 **Valoración del Producto:** EXC_P_01

5. GLOSARIO DE TÉRMINOS

Sistema de información: es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad o un objetivo

Sistema operativo – Es un conjunto de programas que sirven para manejar un ordenador.

Software - El conjunto de programas, procedimientos y documentación asociado a un sistema informático.

Javascript: es un lenguaje de programación del lado del cliente que se utiliza con frecuencia en diseño WEB para generar efectos más complejos que no se puedan alcanzar usando HTML.

HTML: Siglas de las palabras inglesas: Hypertext Markup Language. Es decir, lenguaje de marcado de hipertexto. Lenguaje informático para crear páginas web. Conjunto de etiquetas o instrucciones que permiten estructurar el contenido de una web e incluir los hipervínculos o enlaces enlaces a otras páginas. Este lenguaje lo inventó en 1991 el Doctor Berners-Lee del CERN en Suiza.

HTTPS: Siglas de las palabras inglesas: HyperText Transfer Protocol Secure o versión segura del protocolo HTTP. Es el protocolo empleado para la transferencia de ficheros HTML cifrados que puedan contener información confidencial.

HTTP: siglas de las palabras inglesas: Hypertext Transfer Protocol. A saber en español: Protocolo de Transmisión de Hipertexto. Protocolo estándar de transferencia de hipertexto. Es decir: el protocolo de comunicaciones en el que está basado la Word Wide Web.

Script: es un archivo de órdenes o archivo de procesamiento por lotes. Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

MySQL: es un sistema de gestión de bases de datos de código abierto que, junto con PHP, permite darle a las páginas web cierto dinamismo, es decir, disponer de manera adecuada los datos solicitados por los navegadores. Es un sistema multiplataforma y su uso está tan extendido en las bases de datos que podría considerarse un estandar.

SEO (Search Engine Optimisation) Optimización en buscadores: técnica utilizada para asegurar que una página Web es compatible con los motores de búsqueda y así tener la posibilidad de aparecer en las posiciones más altas en los resultados de búsqueda.

Diseño web adaptable (responsive web design): se llama así al diseño web de aquellas páginas que se adaptan al tamaño de la pantalla o ventana en que se despliegan, por medio del uso de, idealmente, un solo documento HTML y un solo documento CSS. Esto permite hacer una sola página web para smartphones, phablets, tablets y PC.

Diagrama o Modelo Entidad Relación (DER): denominado por sus siglas en inglés, E-R "Entity relationship", o del español DER "Diagrama de Entidad Relación") es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades

Bases de Datos (BD): es un banco de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto

6. REFERENTES BILBIOGRÁFICOS

- Documentos técnicos relacionados en la plataforma

https://www.youtube.com/watch?v=6ERdu4k62wI https://www.freecodecamp.org/news/create-an-mvc-framework-from-scratch-with-php/	Use PHP to Create an MVC Framework - Full Course
https://codeburst.io/mvc-design-pattern-analogy-to-an-old-school-landline-3dcd2e994063	MVC Design pattern: analogy to an old school landline
https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/	The Model View Controller Pattern – MVC Architecture and Frameworks Explained

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	**Julian Salazar Pineda**	**Instructor**	**CPIC**	**14-11-2022**

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					