

**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
FACULTAD DE PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN**



**“DOCUMENTACIÓN DEL APLICATIVO WEB PARA EL MANEJO DE
ASISTENCIA Y NOTAS DE LOS ESTUDIANTES”**

ASIGNATURA:

TRABAJO INTERDISCIPLINAR

**SEMESTRE:
III**

ESTUDIANTES:

Mogollon Caceres Sergio Daniel
Parizaca Mozo Paul Antony
Apaza Apaza Nelson Jorge
Castillo Sancho Sergio Ahmed

DOCENTE:

Yesenia Deysi Yari Ramos

Arequipa - Perú

2022

DOCUMENTACIÓN DEL PROYECTO: “APLICATIVO WEB PARA EL MANEJO DE ASISTENCIA Y NOTAS DE LOS ESTUDIANTES”

Título del proyecto	“Historical Student: Un aplicativo web para el manejo y administración de la asistencia, cursos y notas de los estudiantes”
Código del proyecto	Enlace a repositorio en Github: https://github.com/heros789-sergio/proyecto-ti1-php-mysql
Tipo de documento adjunto	Informe del Proyecto (PDF)
Organización/Usuario para quien se redacta el proyecto	Yessenia Deysi Yari Ramos
Autor (es)	<ul style="list-style-type: none"> • Mogollon Caceres Sergio Daniel • Parizaca Mozo Paul Antony • Apaza Apaza Nelzon Jorge • Castillo Sancho Sergio Ahmed
Duración estimada en jornadas:	Fecha de inicio: 1 de Junio de 2022 Fecha de Entrega (II Parcial): 14 de Julio de 2022

1. Funciones de Conexión a Base De Datos

a) Archivo “config/db.php”:

Se define la **clase BaseDatos** que tiene como atributo público *instancia* y como constructor, la instanciación de la clase predefinida PDO para conexiones a bases de datos.

Se define la **función myquery** :

Toma una conexión y una consulta, y devuelve el resultado de la consulta.

Recibe dos parámetros (*conexcion,sql*) y retorna un array asociativo

* **conexcion** La conexión a la base de datos.

* **sql** La consulta SQL a ejecutar.

* **return** Un Array asociativo (Array de arrays).

b) Archivo “pages/logic/baseDatos.php”:

Se define la **clase baseDEdatos**, que también se utiliza para conectarse a la base de datos.

Se define la **función conectar**, que nos permite conectarnos a la base de datos y evalúa si existe un error al establecer la conexión.

Se define la **función insAsistencia** que toma tres parámetros: **asistencia**, **id**, **dia** y devuelve un valor booleano. Utiliza los dos primeros para actualizar una columna en la tabla, y el tercero para identificar la fila a actualizar:

asistencia: 1 o 0.

id: Número de identificación del estudiante.

dia: Columna de la base de datos.

Devuelve un valor booleano de acuerdo a la existencia o no de un error al ejecutar la función.

La función **insTotalDia** actualiza la tabla estadística_diaria con los valores de asistencias (id=1) y faltas (id=2).

Recibe tres parámetros:

totalP cuenta las asistencias.

totalA cuenta las faltas.

dia es el nombre la columna.

Devuelve un valor booleano de acuerdo a la existencia o no de un error al ejecutar la función.

La función **insTotaEstudiante** actualiza las columnas en la tabla estudiantes con los valores de asistencias y faltas totales por estudiante.

Recibe tres parámetros:

totalP cuenta las asistencias (la suma de los valores de la columna “puntos” en la tabla “puntos”).

totalA cuenta las faltas.

id es la identificación del estudiante.

Devuelve un valor booleano.

La función **insTotal** actualiza la columna Total de la tabla estadística_diaria con el valor de la variable \$totalT donde la columna id es igual al valor de totalT.

Recibe dos parámetros:

totalT es el total de suma de columnas.

id es la identificación del estudiante.

Devuelve un valor booleano.

La función **getTabla1** retorna el resultado de la búsqueda de la tabla estudiantes. Retorna NULL si existe un error.

La función **getTabla2** retorna el resultado de la búsqueda de la tabla estadística_diaria. Retorna NULL si existe un error.

La función **cerrar** cierra la conexión con la base de datos.

2. Funciones del Módulo Asistencia:

-Marcar asistencia

a) Archivo view_seleccionDia.php:

En este archivo muestra un botón por cada día de clases en este caso del día 1 al día 20, cada botón enviará un valor en este caso el día por el método POST via php al archivo view_tomar asistencia.

b) Archivo view_tomar_asistencia.php:

El archivo recibirá un valor por el método POST via php que representa el día de la respectiva clase, en base a eso nos mostrará un formulario el cual tendrá por cada estudiante la opción de escoger si asistió o faltó a través de la etiqueta HTML input type="radio", al final del formulario tendremos disponibles dos botones, uno para regresar a la vista de view_seleccionDia.php y otro botón llamado 'tomar asistencia' el cual está vinculado con un archivo javascript llamado codigo.js que nos servirá para enviar todos los datos de la asistencia a la base de datos.

c) Archivo codigo.js

Este archivo javascript le asigna una función al botón de 'tomar asistencia' para que cuando el botón sea pulsado(onclick()) la función se ejecute, la función se llama Agregar() la cual recibirá los datos de la asistencia del archivo view_seleccionDia.php y guardaremos todos los datos en variables que nos mostraran el total de asistencias y faltas. Además, mandará con el metodo POST via php los datos a un archivo llamado insertar.php.

d) Archivo insertar.php

Este archivo será el encargado de hacer la conexión a la base de datos:

```
$BaseDatos = new baseDEdatos("localhost","root","","bd_asistencia");  
$BaseDatos->conectar();
```

y recibir los datos de la asistencia para trabajar con ellos.

```
if($_SERVER["REQUEST_METHOD"] == "POST"){  
    $dia = $_POST["dia"];  
    for ($a = 1; $a<=40; $a++){  
        $asist = $_POST["asist$a"];  
        $BaseDatos->insAsistencia($asist,$a,$dia);  
    }  
}  
  
if($_SERVER["REQUEST_METHOD"] == "POST"){  
    $totalAsistentes = $_POST["totalP"];  
    $totalAusentes = $_POST["totalA"];  
    $BaseDatos->insTotalDia($totalAsistentes,$totalAusentes,$dia);  
}
```

Además invocará a dos funciones `inAsistencia()` la cual insertará la asistencia de cada alumno en la tabla estudiante. Y la otra función `insTotalDia` la cual insertará el total de asistencias y faltantes en la tabla de `estadistica_diaria`

e) Archivo `view_VerAsistencia.php`

El siguiente archivos hará la consultas respectivas a la base de datos para extraer la información a mostrar

```
$sql = "SELECT * FROM estudiantes";  
$consulta = $conexionDB->prepare($sql);  
$consulta->execute();  
$listaDeEstudiantes = $consulta->fetchAll();
```

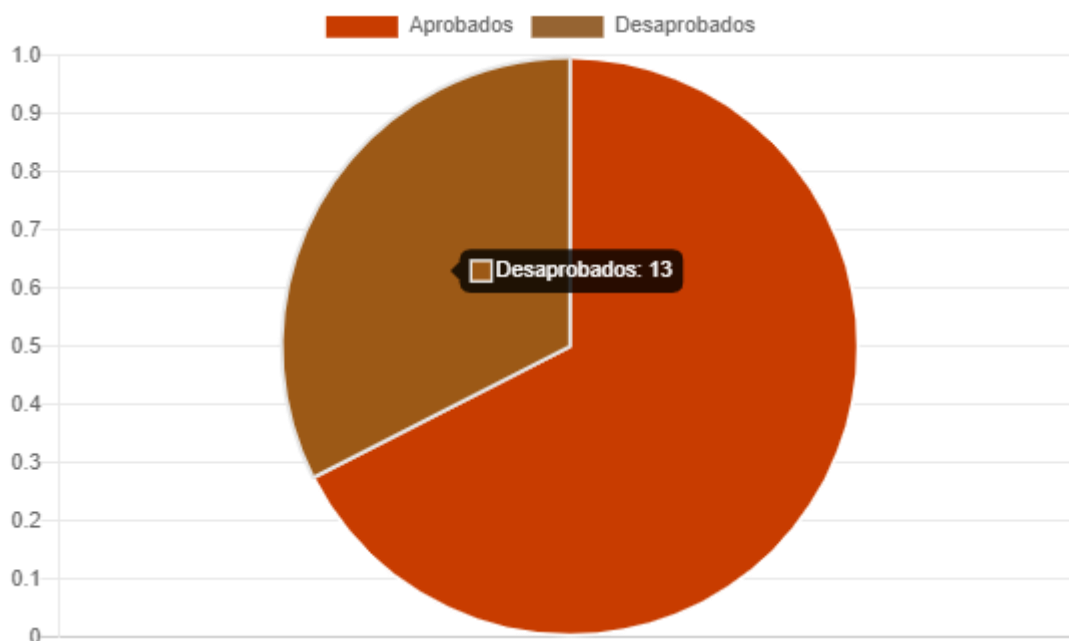
```
$sql = "SELECT * FROM estadistica_diaria";  
$consulta = $conexionDB->prepare($sql);  
$consulta->execute();  
$listaDecondiciones = $consulta->fetchAll();
```

Con estas consultas tenemos lo necesario para mostrar la tablas para que puedan ser visualizadas por el usuario.

a) Archivo “`pages\graphics\asistentes_y_faltantes_bar.php`”

Crea el gráfico estadístico (variable `myChart`) y permite su descarga a través de la función `Descargar()`.

Ejemplo de objeto `myChart`:



3. Funciones del Módulo Notas

a) Archivo “pages/logic/cursos.php”

Se crea una nueva instancia de la clase BaseDatos y se empiezan a asignar valores a las variables que contienen aprobados, desaprobados.

Asimismo, se obtienen las máximas notas, mínimas notas y promedios de notas a través de la función myquery().

b) Archivo “js\codigo.js”

La función Agregar permite obtener el valor de cada radio button y los agrega a las variables de asistencia.

Luego, envía las variables a un archivo .php (insertar.php).

c) Archivo “pages\logic\registroNotas.php”:

Se genera el archivo .pdf a del registro de notas a partir de una instancia de la base de datos.

d) Las demás funciones se realizan a través de consultas sql que son almacenadas en array asociativos del lenguaje php