

Universidad Nacional de San Agustín

Facultad de Ingeniería de Producción y Servicios Escuela
Profesional de Ciencia de la Computación



Proyecto Final

ASIGNATURA: Trabajo Interdisciplinar

DOCENTE: Yessenia Deysi Yari Ramos

PERIODO ACADÉMICO: SEMESTRE 2022-A

GRUPO: A

INTEGRANTES: Eduardo German Ruiz Mamani

López Condori, Andrea del Rosario

Sergio Sebastian Santos Mena Quispe

Linghai Joaquin Zhong Callasi

AREQUIPA - PERÚ

2022

1. INTRODUCCIÓN

1.1 Presentación y objetivos

El objetivo del presente proyecto se enfocará en la implementación de una página web para registro de asistencias y calificaciones de los alumnos, para docentes de la Escuela Profesional de Ciencia de la Computación, que pueda facilitar el registro de las asistencias y calificaciones de los alumnos universitarios, también generar reportes por semana, mes o semestre, incluso calcular datos estadísticos, además poder disponer de la información en el momento que sea necesario.

1.2 Planteamiento del problema

El problema que se nos ha planteado es la construcción del sitio web. La funcionalidad de la aplicación, a grandes rasgos, debía ser:

Asistencia:

En el caso de la asistencia, es necesario generar algunas estadísticas tales como:

- Cantidad de alumnos que han asistido a clase durante el semestre
- Cuántas clases se ha tenido en el semestre
- Cuántos alumnos han abandonado el curso (no tienen asistencia)
- Cuántos alumnos no han asistido durante el semestre
- Por clase generar: Cantidad de asistentes, faltantes.
- Por cada alumno, generar porcentaje de asistencia.

Notas:

- Cada periodo, se tiene 2 notas, una nota continua y una de examen.
- Para cada nota es necesario saber:
 - ¿Quién obtuvo la mejor nota y cual es?
 - ¿Quién obtuvo la peor nota y cuál es?
 - La nota promedio y cual es?
- Cuántos alumnos aprobaron.
- Cuántos alumnos jalaron.
- Quienes están en peligro de jalar el curso (teniendo la primera y segunda nota)
- Generar el registro con las notas completas y la nota final.

Las notas pueden salir :

La continua de trabajos que los estudiantes presentan.

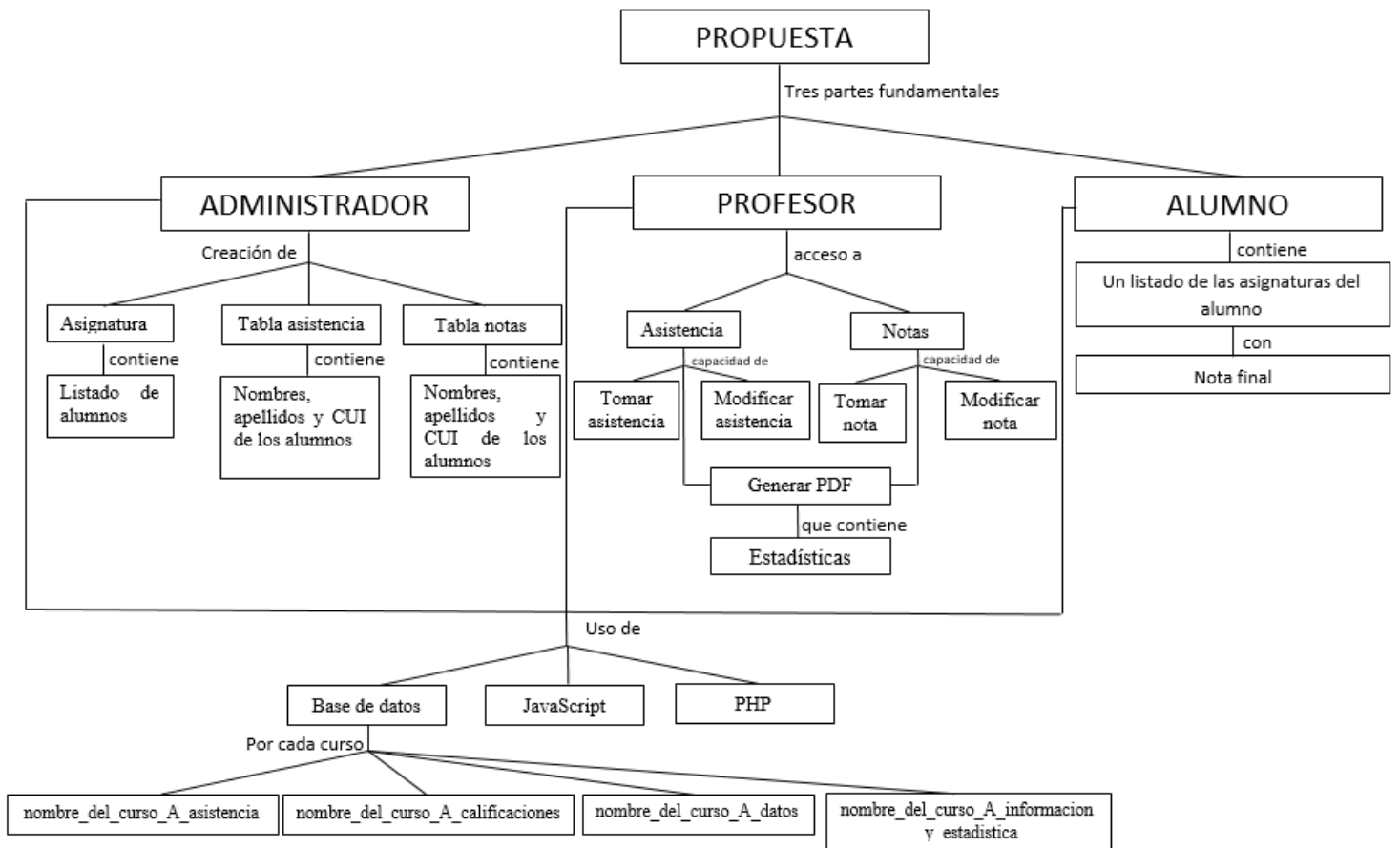
El examen es una sola nota

Generar un reporte por alumnos (Notas y Asistencia por curso)

Tener en consideración que

- Un alumnos puede llevar varios cursos
- Generar el histórico de notas (todos los cursos llevados y su nota final)
- Indicar qué cursos le faltan (por año)
- Indicar qué cursos pueden llevar el siguiente semestre.

2. DIAGRAMA DE PROPUESTA



3. MÓDULO ASISTENCIA

A continuación se muestra el contenido de los ficheros que conforman el módulo Asistencia, estos ficheros las principales funciones de base_datos.php

```
function crear($curso) {
    $comand = "CREATE TABLE " . $curso . "_asistencia (cui INT(8) PRIMARY KEY, nombres VARCHAR(100),
    apellidos VARCHAR(100), hora_1 VARCHAR(100), hora_2 VARCHAR(100));";
    mysqli_query($this->conexion, $comand);
    $error = mysqli_error($this->conexion);
}
```

Fig. 1

En la figura 1 se muestra la función de creación de las tabla de asistencia de un curso (contiene lo que es el CUI, nombres, apellidos, hora_1 y hora_2), se guardará esta tabla como “nombredelcurso_asistencia”.

```
58 function insesion($curso) {
59     $Date = date('d_m_Y',time());
60     mysqli_query($this->conexion, "ALTER TABLE $curso" . "_asistencia" . " ADD $Date CHARACTER NULL");
61     mysqli_query($this->conexion, "UPDATE cursos SET total_Horas = total_Horas + 1 WHERE nombre = '$curso'");
62     $error = mysqli_error($this->conexion);
63
64     if (empty($error)) {
65         return true;
66     }
67     echo "Error al ingresar sesion!";
68     return false;
69 }
```

Fig. 2

En esta parte (Figura 2) es la función contar la cantidad de veces que el estudiante ingreso al curso, lo toma como el total de horas, que va aumentando dependiendo de las veces que el profesor toma asistencia.

```
function insasistenciaclase($curso, $valor, $sesion, $ident) {
    $comand = "UPDATE $curso" . "_asistencia" . " SET $sesion = '$valor' WHERE cui = '$ident'";
    mysqli_query($this->conexion, $comand);
    $error = mysqli_error($this->conexion);

    if ($valor == 'P') {
        $comand = "UPDATE $curso" . "_datos" . " SET total_Asistencia = total_Asistencia + 1 WHERE cui = '$ident'";
        mysqli_query($this->conexion, $comand);
        $error = mysqli_error($this->conexion);
    }

    if (empty($error)) {
        return true;
    }
    echo "Error al insertar valores!";
    return false;
}
```

Fig. 3

En la figura 3 muestra la función que tiene como finalidad contar las asistencias que ha tenido cada alumno, esta como el total de asistencias, que va aumentando si el valor es “P”.

```
function getTablaAsistencia($curso){
    $result = mysqli_query($this->conexion,"SELECT * FROM ".$curso."_asistencia");
    $error = mysqli_error($this->conexion);
    $i = 0;
    if(empty($error)){
        if (mysqli_num_rows($result) > 0) {
            while ($row = mysqli_fetch_assoc($result)){
                $array[$i] = $row;
                $i = $i + 1;
            }
        }
        return json_encode($array);
    }else{
        echo "Error";
    }
    return null;
}
```

Fig.4

Retorna todos los datos de la tabla asistencia de un curso

3.1 GRÁFICOS

```
function graficarBarras(data,div){
    var chart = new CanvasJS.Chart(div, {
        animationEnabled: true,
        axisY: {
            title: "Asistentes",
            titleFontColor: "#4F81BC",
            lineColor: "#4F81BC",
            labelFontColor: "#4F81BC",
            tickColor: "#4F81BC"
        },
        axisY2: {
            title: "Faltos",
            titleFontColor: "#C0504E",
            lineColor: "#C0504E",
            labelFontColor: "#C0504E",
            tickColor: "#C0504E"
        },
        tooltip: {
            shared: true
        },
        legend: {
            cursor:"pointer",
            itemclick: toggleDataSeries
        },
        data: [{
            type: "column",
            name: "Asistentes",
            legendText: "Asistentes",
            showInLegend: true,
            dataPoints: data[0]
        },
        {
            type: "column",
            name: "Faltos",
            legendText: "Faltos",
            axisYType: "secondary",
            showInLegend: true,
            dataPoints: data[1]
        }
    ]
});
chart.render();
}
```

Fig. 5

Función para realizar el gráfico de barras con la librería CanvasJS

```
function graficarPie(obj,div){
    var chart = new CanvasJS.Chart(div, {
        animationEnabled: true,
        data: [{
            type: "pie",
            startAngle: 240,
            yValueFormatString: "##0.00\\\"%\\\"\"",
            indexLabel: "{label} {y}",
            dataPoints: obj
        }]
    });
    chart.render();
}
```

fig. 6

Función para realizar un gráfico de pie.

4. MÓDULO DE CALIFICACIONES

El módulo de calificaciones permite que el profesor pueda ingresar las calificaciones de sus alumnos en las 6 principales notas de los estudiantes NC_1, NC_2, NC_3, EX_1, EX_2 y EX_3 y calcular la nota final (NF) respecto a eso,

```
$notas = $BaseDatos->getColumnasClases($clase . "_calificaciones");
if(!is_null($notas)) {
    $porcentajesNotas=$BaseDatos->getInfoCursos("cursos",$clase);
    while($row=mysqli_fetch_assoc($notas)){
        if(!is_null($porcentajesNotas)) { /*Por si se actualiza*/
            while ($row2=mysqli_fetch_assoc($porcentajesNotas)){
                $parcial1porc=floatval($row2['EP_1']);
                $parcial2porc=floatval($row2['EP_2']);
                $parcial3porc=floatval($row2['EP_3']);
                $continua1porc=floatval($row2['EC_1']);
                $continua2porc=floatval($row2['EC_2']);
                $continua3porc=floatval($row2['EC_3']);
            }
        } /*Por si se actualiza*/
    }
}
```

Fig.7

En la figura 7 podemos ver el procedimiento que se sigue para obtener los porcentajes de cada nota principal que se sacan de la tabla cursos, usando la función getInfoCursos mostrada en la siguiente figura.

```
function getInfoCursos($tabla, $curso) { /*Agregado*/
    $result = mysqli_query($this->conexion, "SELECT * FROM `$tabla` WHERE nombre = '" . $curso . "'");
    $error = mysqli_error($this->conexion);
    if (empty($error)) {
        if (mysqli_num_rows($result) > 0) {
            return $result;
        }
    } else {
        echo "Error al obtener clases!";
    }
    return null;
}
```

Fig.8

Además de estas funciones también es posible agregar nuevos campos, para el caso de la NC_1, NC_2 y NC_3 (notas continuas), ya que sabemos que el profesor puede calcular su nota continua a partir de notas pequeñas como prácticas o laboratorios.

```

<?php include("base_datos.php"); ?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8"/>
    <title>
        <?php
            $clase = $_GET["clase"];
            echo "$clase";
        ?>
    </title>
    <link rel="stylesheet" href="../../css/formulario_carpeta.css">
</head>
<body>
    <h2 style="text-align: center;">Notas del Curso<br><br></h2>
    <form>
        <h3>Añadir nota</h3>
        <div>
            <label>Nombre de la nota</label>
            <input id="nombrenewnota" type="text"/>
        </div>
        <div>
            <label>Nota Superior</label>
            <select name="notaSuperiornewnota" id="notaSuperiornewnota">
                <option value="NC_1">NC_1</option>
                <option value="NC_2">NC_2</option>
                <option value="NC_3">NC_3</option>
            </select>
        </div>
        <div>
            <label>Porcentaje</label>
            <select name="porcentajenewnota" id="porcentajenewnota">
                <?php
                    for ($i = 1; $i <= 100; $i++) {
                        echo "<option value='i'>$i</option>";
                    }
                ?>
            </select>
        </div>
        <button id="btnAgregar" type="button">Agregar</button>
    </form>

    <!-- <form method="post" action="Proyecto_Calificaciones.php?clase=<?php echo $_GET['clase'] ?>" -->
    <h2>Notas Existentes</h2>
    <table id="tablaEstudiantes" class="tabla">
        <thead>
            <tr>
                <th>Nota</th>
                <th>Nota Superior</th>
                <th>Porcentaje</th>
            </tr>
        </thead>
        <tbody id="contenido">
            <?php
                $BaseDatos = new base_datos("localhost", "root", "", "ti_ciencias_computacion");
                $BaseDatos->conectar();
                $clase = $_GET["clase"];

                $campnotas = $BaseDatos->getInfoNotas("notas", "notaSuperior", "porcentaje", $clase . "_informacion_y_estadistica");
                if(!is_null($campnotas)) {
                    while ($row = mysqli_fetch_assoc($campnotas)) {
                        echo "<tr>";
                        echo "<td>" . $row["notas"] . "</td>";
                        echo "<td>" . $row["notaSuperior"] . "</td>";
                        echo "<td>" . $row["porcentaje"] . "</td>";
                        echo "</tr>";
                    }
                }
                $BaseDatos->cerrar();
            ?>
        </tbody>
    </table>
    <button onclick="location='Proyecto_Calificaciones.php?clase=<?php echo $_GET['clase'] ?>'>Volver</button>

```

Figura 9. Envío de datos para crear un nuevo campo de nota

Para el envío de estos datos al servidor para la creación de las nuevas posiciones mostradas a partir de las notas creadas, se utiliza AJAX.

```

console.clear();

const isEmpty = (str) => str.trim() === '';

function agregar () {
    var nombrenota = document.getElementById('nombrenewnota');
    var notaSuperiornota = document.getElementById('notaSuperiornewnota');
    var porcentajenota = document.getElementById('porcentajenewnota');

    nombreVal = nombrenota.value;
    notaSuperiorVal = notaSuperiornota.value;
    porcentajeVal = porcentajenota.value;

    tablaEstudiantes = document.getElementById('tablaEstudiantes');

    if (window.XMLHttpRequest) {
        ajax = new XMLHttpRequest();
    } else {
        ajax = ActiveXObject("Microsoft.XMLHttpRequest")
    }

    ajax.onreadystatechange = function() {
        if(ajax.readyState == 4 && ajax.status == 200) {
            contenido.innerHTML = ajax.responseText;
        } else {
            contenido.innerHTML = "<img width='50' height= '50' src='../resources/cargando.gif'>"
        }
    }

    const queryString = window.location.search;
    const urlParams = new URLSearchParams(queryString);
    const clase = urlParams.get('clase');

    ajax.open("POST", "../php/insertar_campo_notas.php?clase=" + clase);
    ajax.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    ajax.send("nombre=" + nombreVal + "&notaSuperior=" + notaSuperiorVal + "&porcentaje=" + porcentajeVal);
}

function asignar() {
    btnAgregar = document.getElementById('btnAgregar');
    btnAgregar.addEventListener("click", agregar);
};

window.addEventListener("load", asignar);

```

Figura 9. Usando AJAX para la página de ingreso de nuevos campos.

También enviamos nuevos datos usando el código mostrado en la figura 10, de forma que para los campos de notas de los alumnos se puedan ingresar notas numéricas que luego son procesadas en los cálculos de la página principal de calificaciones (generada por Proyecto_Calificaciones.php).


```

45     if(!is_null($estudiantes)) {
46         while ($row = mysqli_fetch_assoc($estudiantes)) {
47             echo "<tr>";
48             echo "<td class='CUI'>" . $row["cui"] . "</td>";
49             echo "<td class='nombre'>" . $row["nombre"] . "</td>";
50             echo "<td class='apellido'>" . $row["apellido"] . "</td>";
51
52             $notaxest = $BaseDatos->getInfoEstudiantes($clase . "_calificaciones", $row["cui"]);
53             if(!is_null($notaxest)) {
54                 $row_nxe = mysqli_fetch_assoc($notaxest);
55                 $notas = $BaseDatos->getColumnasClases($clase . "_calificaciones");
56                 while ($row_not = mysqli_fetch_assoc($notas)) {
57                     if ($row_not['column_name'] != 'cui' && $row_not['column_name'] != 'NF') {
58                         if ($row_not['column_name'] == 'NC_1') {
59                             $camposxnot = $BaseDatos->getCamposNota($row_not['column_name'], $clase);
60                             if(!is_null($camposxnot)) {
61                                 echo "<td>";
62                                 if ($row_nxe[$row_not['column_name']] != NULL) {
63                                     echo floatval($row_nxe[$row_not['column_name']]);
64                                 }
65                                 echo "</td>";
66                             } else {
67                                 echo "<td><input name='". $row["cui"] . "_" . $row_not['column_name'] . "' value=''. '' type='text'></td>";
68                             }
69                         } else if ($row_not['column_name'] == 'NC_2') {
70                             $camposxnot = $BaseDatos->getCamposNota($row_not['column_name'], $clase);
71                             if(!is_null($camposxnot)) {
72                                 echo "<td>";
73                                 if ($row_nxe[$row_not['column_name']] != NULL) {
74                                     echo floatval($row_nxe[$row_not['column_name']]);
75                                 }
76                                 echo "</td>";
77                             } else {
78                                 echo "<td><input name='". $row["cui"] . "_" . $row_not['column_name'] . "' value=''. '' type='text'></td>";
79                             }
80                         } else if ($row_not['column_name'] == 'NC_2') {
81                             $camposxnot = $BaseDatos->getCamposNota($row_not['column_name'], $clase);
82                             if(!is_null($camposxnot)) {
83                                 echo "<td>";
84                                 if ($row_nxe[$row_not['column_name']] != NULL) {
85                                     echo floatval($row_nxe[$row_not['column_name']]);
86                                 }
87                                 echo "</td>";
88                             } else {
89                                 echo "<td><input name='". $row["cui"] . "_" . $row_not['column_name'] . "' value=''. '' type='text'></td>";
90                             }
91                         } else {
92                             if ($row_nxe[$row_not['column_name']] != NULL) {
93                                 echo "<td><input name='". $row["cui"] . "_" . $row_not['column_name'] . "' value='". floatval($row_nxe[$row_not['column_name']]) . "' type='text'></td>"; //Convertimos el valor recibido a flotante
94                             } else {
95                                 echo "<td><input name='". $row["cui"] . "_" . $row_not['column_name'] . "' value=''. '' type='text'></td>"; //Si es nulo, no escribe nada, porque no hay nota, si no se usa esto, va a convertir el NULL en 0 y entonces ese valor pasa a la base de datos como numérico.

```

Figura 10. Ingresando nuevos datos y enviando a la base de datos.

5. CONCLUSIONES

Una vez avanzado el proyecto hasta esta parte, podemos decir que hemos cumplido los objetivos definidos inicialmente, puesto que hemos creado un aplicación web capaz de permitir a un centro docente pasar asistencia y tomar notas de las diferentes asignaturas, sin embargo, sabemos que hay maneras de hacerlo más eficiente y óptimo, haremos lo posible para pulir esos pequeños detalles para la 3era fase, además de agregar los nuevos contenidos que se piden. Por otro lado, el desarrollo de la aplicación ha permitido mejorar el conocimiento en diseño web y los lenguajes utilizados para este fin, durante la realización de esta hemos podido comprobar lo útiles que son los conocimientos adquiridos durante la carrera en asignaturas como Desarrollo Basado en Plataformas.

Muchas mejoras y ampliaciones, como las que se citan en el siguiente punto, quedan fuera del alcance de este proyecto por su duración y extensión, sin embargo, las funciones descritas en la especificación

de requisitos están implementadas. Luego, bajo el punto de vista de nuestro grupo, se trata de un trabajo realizado satisfactoriamente.

6.1 Limitaciones y dificultades

Considerando las diferentes limitaciones, la más importante ha sido la falta de conocimientos en los diferentes lenguajes de programación, sumando a la situación actual, las reuniones virtuales no fueron igual de eficientes que las reuniones presenciales.

Enlace al repositorio:

<https://github.com/Sommerfield3/Trabajo-Interdisciplinar>

6. REFERENCIAS

Librería CanvasJS utilizada para mostrar los gráficos:

<https://canvasjs.com/>

Método para convertir HTML a PDF

<https://parzibyte.me/blog/2020/09/05/html-pdf-javascript/>

Liberia html2pdf

<https://github.com/eKoopmans/html2pdf.js/tree/master/dist>

Referencias de formato

<https://riunet.upv.es/bitstream/handle/10251/10273/Memoria.pdf>

