

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

En este documento, se explicarán las funcionalidades de los diferentes métodos y atributos utilizados en el proyecto realizado para el curso de Trabajo Interdisciplinar realizado entre los meses de abril y agosto del año 2022. Dónde se abordaron temas desde la generación de tablas en una base de datos SQL hasta el manejo dinámico de páginas web a través del uso de AJAX.

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. Conceptos previos:

Antes de comenzar a analizar el código fuente, es necesario conocer algunos conceptos previos utilizados durante el desarrollo del proyecto:

2.1 MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo. MySQL está escrito en C y C++. Su analizador SQL está escrito en yacc (Yet Another Compiler-Compiler), pero utiliza un analizador léxico casero. MySQL funciona en muchas plataformas de sistema, incluidas AIX, BSDi, FreeBSD, HP-UX, ArcaOS, eComStation, IBM i, IRIX, Linux, macOS, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos y Tru64. También existe un puerto de MySQL a OpenVMS.

2.3 MD5

El Message Digest 5 es un algoritmo de reducción criptográfica de 128 bits (este tamaño es independiente del tamaño del archivo) de alta velocidad para el procesamiento de texto y encriptación o también para la autenticación de texto dado el resumen generado. Durante el procesamiento, el mensaje será extendido hasta que su longitud en bits sea congruente con 448, módulo 512. Esto es, si se le resta 448 a la longitud del mensaje tras este paso, se obtiene un múltiplo de 512. Esta extensión se realiza siempre, incluso si la longitud del mensaje es ya congruente con 448, módulo 512. En el último bloque del 512 bits se aplica un relleno de un 1 con tantos como haga falta, reservando los últimos 64 bits para indicar el tamaño del archivo. MD5 trata la lectura de bytes de menos significativos a más significativos (little endian). Se utilizan 4 vectores públicos de 32 bits cada uno (A, B, C, D) para mezclarse con las 16 palabras de cada bloque de entrada de 512 bits, realizando un total de 64 vueltas. Luego de las 64 vueltas, los vectores A, B, C, D, habrán cambiado varias veces de valor y luego se procede a usarlos como los nuevo 4 vectores que se mezclan con el segundo bloque de bits de entrada y así sucesivamente mientras se encadenan, el último valor de los vectores A, B, C y D es el resumen de 128 bits del archivo ingresado. Debido a la detección de colisiones (diferentes entradas, mismas salidas) su nivel de seguridad se ha visto drásticamente reducida, pero no obstante aún sigue en uso aunque no es recomendado para la trata de información sensible.

2.4 phpMyAdmin

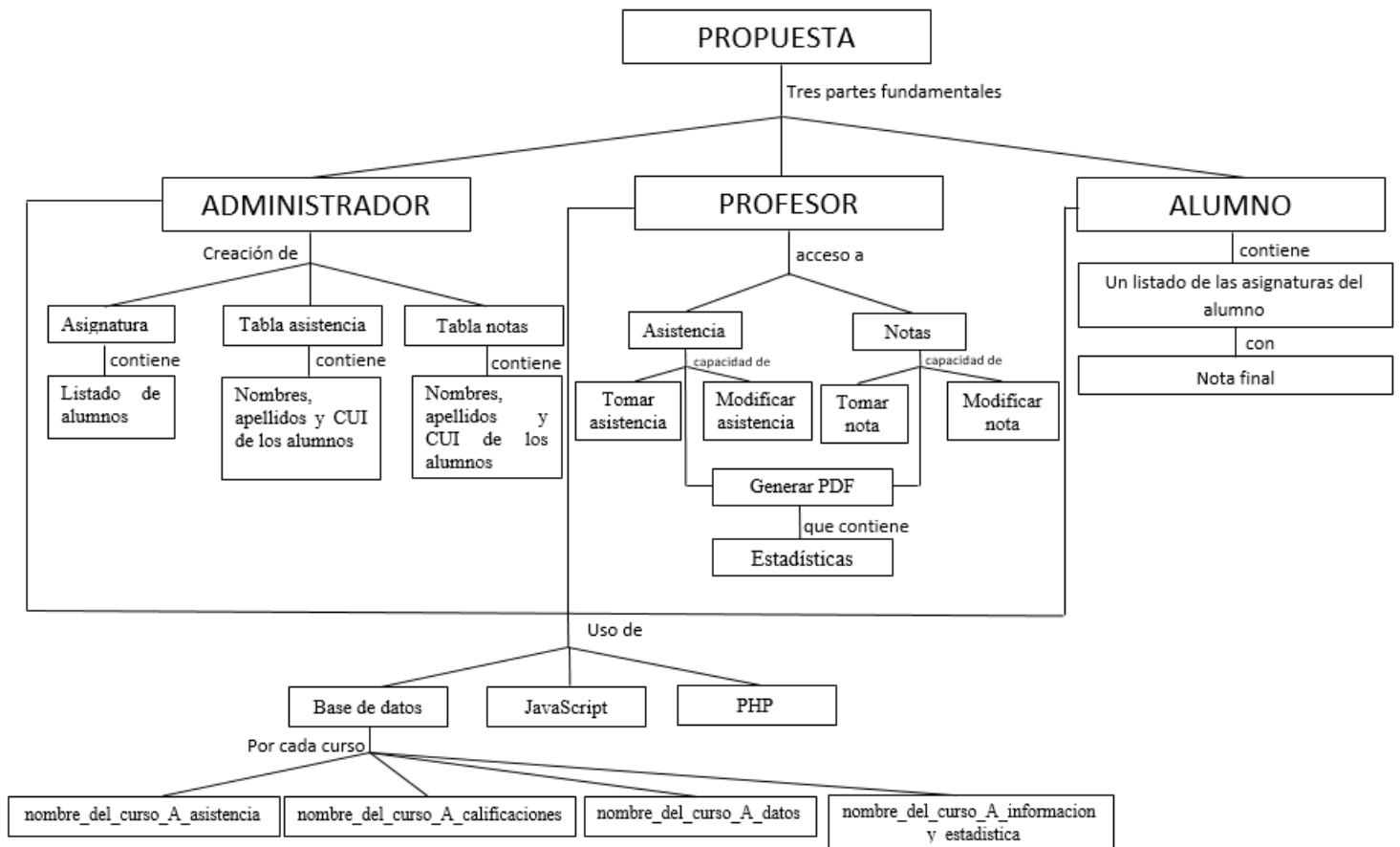
Es una herramienta de código abierto usada para la administración de bases de datos en MySQL y MariaDB, escrita en gran parte en PHP, cuenta con interfaz web, importación de data desde archivos CSV, JSON y SQL. Administración de múltiples servidores. Búsqueda global de información dentro de las bases de datos, entre otras funciones.

2.5 Canvasjs

Contiene una API en javascript simple especializada en la generación, en alta velocidad, de cuadros y gráficos HTML5 de alta velocidad. Es utilizable en React, jQuery, php, javascript, jsp, spring.

3. Desarrollo:

El proyecto, dividido en 3 fases, el desarrollo se realizó siguiendo el siguiente diagrama:



4. 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

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

```
75 function insesion($codcursoturn) {
76     $Date = date('d_m_Y',time());
77     mysqli_query($this->conexion, "ALTER TABLE $codcursoturn . "_asistencia" . " ADD $Date CHARACTER NULL");
78     mysqli_query($this->conexion, "UPDATE cursos SET total_Horas = total_Horas + 1 WHERE codigo = '$codcursoturn'");
79     $error = mysqli_error($this->conexion);
80
81     if (empty($error)) {
82         return true;
83     }
84     echo "Error al ingresar sesion!";
85     return false;
86 }
```

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.

```
89 function insasistenciaclase($codcursoturn, $valor, $sesion, $ident) {
90     $comand = "UPDATE $codcursoturn . "_asistencia" . " SET $sesion = '$valor' WHERE cui = '$ident'";
91     mysqli_query($this->conexion, $comand);
92     $error = mysqli_error($this->conexion);
93
94
95     if ($valor == 'P') {
96         $comand = "UPDATE $codcursoturn . "_datos" . " SET total_Asistencia = total_Asistencia + 1 WHERE cui = '$ident'";
97         mysqli_query($this->conexion, $comand);
98         $error = mysqli_error($this->conexion);
99     }
100
101     if (empty($error)) {
102         return true;
103     }
104     echo "Error al insertar valores!";
105     return false;
106 }
107
```

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”.

```
362 function getTabla($codcursoturn,$tabla){
363     $result = mysqli_query($this->conexion,"SELECT * FROM ".$codcursoturn."_".$tabla);
364     $error = mysqli_error($this->conexion);
365     $i = 0;
366     if(empty($error)){
367         if (mysqli_num_rows($result) > 0) {
368             while ($row = mysqli_fetch_assoc($result)){
369                 $array[$i] = $row;
370                 $i = $i + 1;
371             }
372         }
373         return json_encode($array);
374     }else{
375         echo "Error";
376     }
377     return null;
378 }
```

Fig.4

Retorna todos los datos de la tabla asistencia de un curso

5. Módulo Calificaciones

A continuación se mostrará el contenido del módulo de calificaciones. Comienza en la vista de las calificaciones del curso, en el archivo Calificaciones.php, donde columna por columna se van mostrando las calificaciones de los alumnos.

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

Fig. 5

En la figura 5 se puede observar la función que obtiene las calificaciones del estudiante según curso y también según cui.

```
<thead>  
    <tr>  
        <th>Nota</th>  
        <th>Mejor Nota</th>  
        <th>CUI</th>  
        <th>Nombre</th>  
        <th>Peor Nota</th>  
        <th>CUI</th>  
        <th>Nombre</th>  
    </tr>  
</thead>  
<tbody id="contenido">  
<?php  
    $BaseDatos = new base_datos("localhost", "root", "", "ti_ciencias_computacion");  
    $BaseDatos->conectar();  
    $clase = $_GET["clase"];  
    $datosestadisticos = $BaseDatos->getNotasEstadistica($clase);  
    echo "<tbody>";  
    if(!is_null($datosestadisticos)) {  
        while ($row = mysqli_fetch_assoc($datosestadisticos)) {  
            echo "<tr>";  
            echo "<td>" . $row["notas"] . "</td>";  
            echo "<td class='nota'>" . $row["mejorNota"] . "</td>";  
            echo "<td class='CUI'>" . $row["cuiMejorNota"] . "</td>";  
            echo "<td class='nombre'>" . $row["nomMejorNota"] . "</td>";  
            echo "<td>" . $row["peorNota"] . "</td>";  
            echo "<td class='nota'>" . $row["cuiPeorNota"] . "</td>";  
            echo "<td class='nombre'>" . $row["nomPeorNota"] . "</td>";  
            echo "</tr>";  
        }  
    }  
    echo "</tbody>";
```

Fig. 6

Como se visualiza en la figura 6, también mostramos datos del propio curso, como la mejor nota, y la peor nota, así también sus correspondientes datos.

```
<form method="post" action="Calificaciones.php?clase=<?php echo $_GET['clase'] ?>">
  <button type="submit" id="btnTomarAssist" />Guardar</button>
```

Fig. 7

Para el proceso de ingreso de notas usamos un formulario que referencie a la tabla de Calificaciones antes mostrado, como se ve en la figura 7.

```
if ($row_not['column_name'] = 'NC_1') {
    $camposnot = $BaseDatos->getCamposNota($row_not['column_name'], $clase);
    if(!is_null($camposnot)) {
        echo "<td>";
        if ($row_nxe[$row_not['column_name']] != NULL) {
            echo floatval($row_nxe[$row_not['column_name']]);
        }
        echo "</td>";
    } else {
        echo "<td><input name='". $row["cui"] . "_" . $row_not['column_name'] . 'value=' . "'' type='text'></td>";
    }
}
```

Fig. 8

El proceso para generar los inputs para el ingreso de cada calificación es el mismo que el mostrado en la figura 8.

6. Modulo Alumno

A continuación se mostrará el contenido de los archivos del módulo, empezando por el login que nos envía el código del alumno con el que estamos ingresando. Una vez lo obtenemos somos redireccionados a Portada.php de la carpeta Estudiante.

```
38
39 $clases_estudiante = $BaseDatos->getClases("semestre_actual");
40 $clases_codigo = $BaseDatos->getEncabezados("semestre_actual");
41
42 $i = 0;
43 if(!is_null($clases_codigo)){
44     while($row = mysqli_fetch_assoc($clases_codigo)){
45         $codigos[$i] = $row["Field"];
46         $i = $i + 1;
47     }
48 }
49
50 $i = 0;
51 if(!is_null($clases_estudiante)){
52     while ($row = mysqli_fetch_assoc($clases_estudiante)) {
53         if($row["cui"] == "est" . $_SESSION["usuario"]){
54             foreach ($row as $key => $value) {
55                 if($value == 1){
56                     echo "<option value='". $codigos[$i] . "'>". $BaseDatos->getNombreClase($codigos[$i]) . "</option>";
57                 }
58                 $i = $i + 1;
59             }
60         }
61     }
62 }
```

En la figura __ podemos ver cómo se seleccionan las clases a las que pertenece el estudiante, las clases en las que está matriculado, de acuerdo a nuestra tabla semestre_actual donde con valores booleanos identificamos si un alumno lleva o no un curso en el presente semestre.

Seleccionamos la clase de la que queremos saber nuestros datos como estudiante y nos envía al archivo Clase.php

```

34
35 $BaseDatos = new base_datos("localhost", "root", "", "ti_ciencias_computacion");
36 $BaseDatos->conectar();
37 $codigo = $_GET["clase"];
38 $valnotas = $BaseDatos->getCalifEstudiantes($codigo,"est".$_SESSION["usuario"]);
39 $nomnotas = $BaseDatos->getEncabezados($codigo."_calificaciones");
40
41 $i = 0;
42 if (!is_null($nomnotas)) {
43     while($row = mysqli_fetch_assoc($nomnotas)) {
44         $nombresnotas[$i] = $row["Field"];
45         $i = $i + 1;
46     }
47 }
48
49 /* Se muestran los datos de los alumnos en una tabla */
50 $i = 0;
51 if(!is_null($valnotas)){
52     while ($row = mysqli_fetch_assoc($valnotas)) {
53         foreach ($row as $key => $value) {
54             if ($nombresnotas[$i] != "cui") {
55                 echo "<tr>";
56                 echo "<td>" . $nombresnotas[$i] . "</td>";
57                 echo "<td>" . $row[$nombresnotas[$i]] . "</td>";
58                 echo "</tr>";
59             }
60             $i = $i + 1;
61         }
62     }
63 }

```

En la figura __ se puede ver como obtenemos el nombre de todas las notas del curso y lo guardamos en un array, después buscamos del cui del estudiante en la tabla de calificaciones del curso y obtenemos sus notas hasta ahora, filtramos estos parámetros e imprimimos en la tabla los datos,

Al ingresar se muestra las notas del estudiante en el curso, pero tenemos otras opciones como historico_notas.

```

25
26 $BaseDatos = new base_datos("localhost", "root", "", "ti_ciencias_computacion");
27 $BaseDatos->conectar();
28 $valcursos = $BaseDatos->getHistoricoest("est".$_SESSION["usuario"]);
29 $codcursos = $BaseDatos->getEncabezados("historico_notas");
30
31 $i = 0;
32 if (!is_null($codcursos)) {
33     while($row = mysqli_fetch_assoc($codcursos)) {
34         $nombrecursos[$i] = $row["Field"];
35         //echo "<p>". $nombrecursos[$i]. "</p>";
36         $i = $i + 1;
37     }
38 }
39
40 /* Se muestran los datos de los alumnos en una tabla */
41 $i = 0;
42 if(!is_null($valcursos)){
43     while ($row = mysqli_fetch_assoc($valcursos)) {
44         foreach ($row as $key => $value) {
45             if ($nombrecursos[$i] != "cui" && $row[$nombrecursos[$i]] != NULL) {
46                 echo "<tr>";
47                 echo "<td>" . $nombrecursos[$i] . "</td>";
48                 echo "<td>" . $row[$nombrecursos[$i]] . "</td>";
49                 echo "</tr>";
50             }
51             $i = $i + 1;
52         }
53     }
54 }
55

```

En la figura __ se muestra cómo obtenemos los nombres de los cursos, en este caso su código de la tabla historico_notas donde están todos los cursos de la carrera, seleccionamos el

estudiante y vemos que notas tiene, si ya pasó el curso mostrará su nota, sino tendrá una nota NULL y no se tomará en cuenta.

La parte de los cursos que próximamente se pueden llevar en el semestre y el año no se logró completar debido a problemas con la interfaz, pero dentro de la base de datos ya se tenía la idea.

codigo	semestre	año	nombre	turno	total_Horas	EP_1	EP_2	EP_3	EC_1	EC_2	EC_3	NF	cui_docente_1	cui_docente_2	curso_1	curso_2	curso_3
1702117A	0	2	ARQUITECTURA DE COMPUTADORES	A	0	20.0	20.0	27.0	9.0	9.0	15.0	100.0	NULL	NULL	1703239	NULL	NULL
1702118A	0	2	CIENCIAS DE LA COMPUTACION II	A	0	12.0	12.0	18.0	18.0	18.0	24.0	100.0	NULL	NULL	1702224	1703132	NULL
1702119A	0	2	DESARROLLO BASADO EN PLATAFORMAS	A	0	9.0	9.0	15.0	20.0	20.0	27.0	100.0	NULL	NULL	NULL	NULL	NULL
1702120A	0	2	TRABAJO INTERDISCIPLINAR	A	5	10.0	10.0	30.0	10.0	10.0	30.0	100.0	docMMXXII08	NULL	1703240	NULL	NULL
1702120B	0	2	TRABAJO INTERDISCIPLINAR	B	0	10.0	10.0	30.0	10.0	10.0	30.0	100.0	docMMXXII07	NULL	1703240	NULL	NULL
1702121A	0	2	CALCULO EN VARIAS VARIABLES	A	0	24.0	24.0	32.0	6.0	6.0	8.0	100.0	NULL	NULL	1702228	1703135	NULL
1702122A	0	2	INGLES TECNICO PROFESIONAL III	A	0	20.0	20.0	27.0	9.0	9.0	15.0	100.0	NULL	NULL	1705160		
1702123A	0	2	CIUDADANIA E INTERCULTURALIC	A	0	10.0	15.0	15.0	20.0	20.0	20.0	100.0	NULL	NULL	NULL	NULL	NULL
1702224A	1	2	ALGORITMOS Y ESTRUCTURAS DE DATOS	A	0	10.0	10.0	15.0	20.0	20.0	25.0	100.0	NULL	NULL	NULL	NULL	NULL
1702225A	1	2	TEORIA DE LA COMPUTACION	A	0	NULL	NULL	NULL	NULL	NULL	NULL	100.0	NULL	NULL	NULL	NULL	NULL
1702226A	1	2	BASE DE DATOS I	A	0	NULL	NULL	NULL	NULL	NULL	NULL	100.0	NULL	NULL	NULL	NULL	NULL
1702227A	1	2	ALGEBRA LINEAL NUMERICA	A	0	NULL	NULL	NULL	NULL	NULL	NULL	100.0	NULL	NULL	NULL	NULL	NULL
1702228A	1	2	ESTADISTICA Y PROBABILIDADES	A	0	10.0	10.0	15.0	20.0	20.0	25.0	100.0	NULL	NULL	NULL	NULL	NULL
1702229A	1	2	ECOLOGIA Y CONSERVACION AMBIENTAL	A	0	NULL	NULL	NULL	NULL	NULL	NULL	100.0	NULL	NULL	NULL	NULL	NULL
1703132A	0	3	INGENIERIA DE SOFTWARE I	A	0	10.0	10.0	15.0	20.0	20.0	25.0	100.0	NULL	NULL	NULL	NULL	NULL
1703135A	0	3	ECUACIONES DIFERENCIALES	A	0	10.0	10.0	15.0	20.0	20.0	25.0	100.0	NULL	NULL	NULL	NULL	NULL
1703238A	1	3	SISTEMAS OPERATIVOS	A	0	10.0	10.0	15.0	20.0	20.0	25.0	100.0	NULL	NULL	NULL	NULL	NULL

Cada curso tiene llaves que abren otros cursos y esos cursos están dentro de la base de datos, también hay una columna que guarda el año en el que se llevan y el semestre, siendo 0 el primer semestre y 1 el segundo semestre, también guarda los porcentajes y otros datos relevantes, con esto se podía guiar los cursos que los alumnos llevan y saber cuales podrían llevar en un futuro.

7. 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.