

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

MANUAL DE USUARIO

Gestor de exámenes y concursos virtuales para el juez en línea de la UAM Azcapotzalco

Martínez Alcántara Carlos 207300217

Trimestre 12-P
Agosto de 2012

Asesor: Rodrigo Alexander Castro Campos

MANUAL DE USUARIO

Gestor de exámenes y concursos virtuales para el juez en línea de la UAM Azcapotzalco

El sistema inicia en la lista de concursos generando tres listas diferentes dependiendo los criterios tomados los cuales son:

- Presentes (figura 1)
- Futuros si no hay concursos presentes (figura 2)
- Pasados si no hay concursos presentes ni futuros (figura 3)

Los concursos pasados se generan si tienen doce horas de que terminaron, los concursos futuros se toman si faltan al menos 12 horas para el inicio del concurso y presentes los que se encuentran dentro del rango descrito.

Pasados < 12 horas

Futuros > 12 horas

12 horas < presentes < 12 horas

La hora tomada como referencia es directamente la del servidor en la que se encuentre montado el sistema.

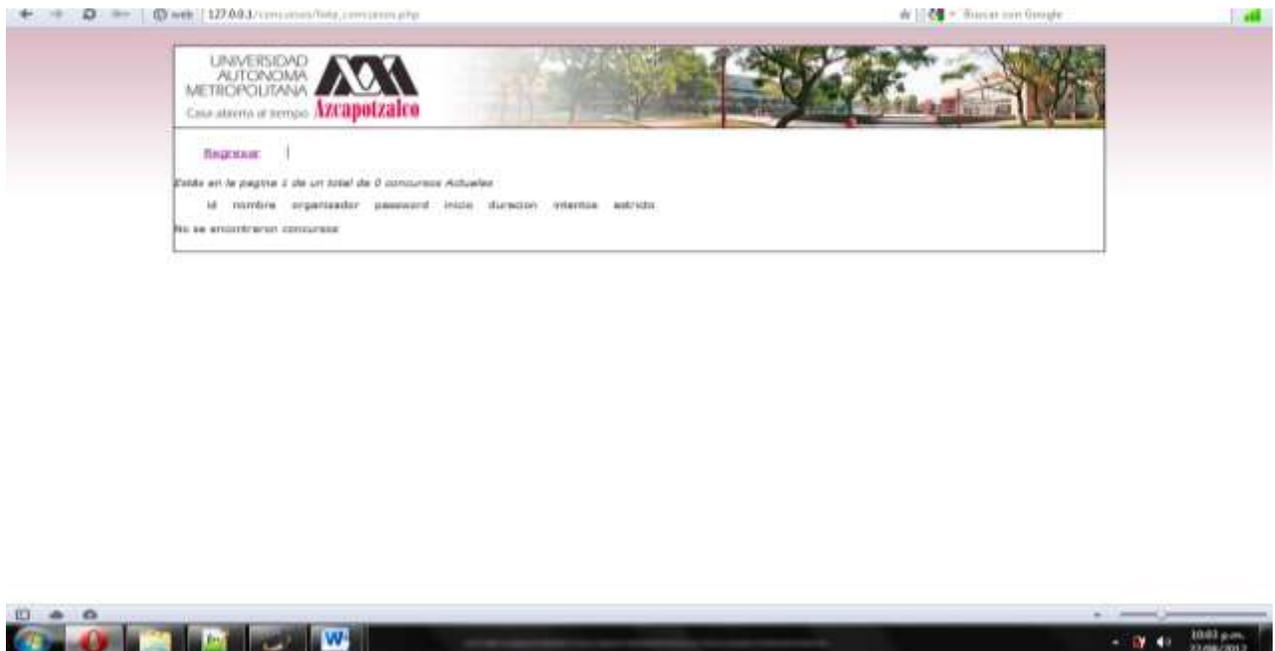


Figura 1 concursos Presentes

UNIVERSIDAD AUTÓNOMA METROPOLITANA Azcapotzalco

Registrar

Estás en la página 1 de un total de 69 concursos Futuros

id	nombre	organizador	password	inicio	duracion	intentos	asignatura
161	44			2013-09-01 16:59:49	16:00:49	3	3
28	17			2012-09-24 22:35:31	22:35:31	3	3
27	82			2012-10-24 01:53:15	01:53:15	7	7
23	7			2012-11-17 14:46:58	14:46:58	1	2
42	83			2012-12-04 23:16:43	23:16:43	1	1
4	73			2013-01-28 02:55:48	02:55:48	10	10
22	45			2013-02-14 11:22:29	11:22:29	1	2
38	38			2013-02-16 03:33:53	03:33:53	7	7
17	36			2013-02-20 07:52:12	07:52:12	1	2
11	8			2013-03-12 20:54:31	20:54:31	1	2

Páginas de concursos: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39 - 40 - 41 - 42 - 43 - 44 - 45 - 46 - 47 - 48 - 49 - 50 - 51 - 52 - 53 - 54 - 55 - 56 - 57 - 58 - 59 - 60 - 61 - 62 - 63 - 64 - 65 - 66 - 67 - 68 - 69

Figura 2 concursos futuros

UNIVERSIDAD AUTÓNOMA METROPOLITANA Azcapotzalco

Registrar

Estás en la página 1 de un total de 39 concursos Pasados

id	nombre	organizador	password	inicio	duracion	intentos	asignatura
18	93			2000-10-02 00:02:03	04:24:33	6	6
13	27			2000-10-08 00:03:03	24:40:04	4	4
12	95			2000-10-02 00:03:00	24:26:58	5	5
10	62			2000-10-02 00:02:03	24:22:54	9	9
86	81			2000-10-08 00:03:03	24:28:38	2	2
11	41			2011-01-22 21:20:22	21:20:22	7	7
29	77			2011-02-02 02:02:44	02:02:44	4	4
25	20			2011-03-13 04:02:29	04:02:29	6	6
14	34			2011-03-17 10:32:12	10:32:12	6	6
12	38			2011-04-04 15:19:13	15:19:13	5	5

Páginas de concursos: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13 - 14 - 15 - 16 - 17 - 18 - 19 - 20 - 21 - 22 - 23 - 24 - 25 - 26 - 27 - 28 - 29 - 30 - 31 - 32 - 33 - 34 - 35 - 36 - 37 - 38 - 39

Figura 3 concursos pasados

Desde este modulo se puede acceder al registro de los usuarios a un concurso determinado, esto se hace solo seleccionando el id del concurso en el cual se desea inscribir un nuevo usuario (figura 4).

UNIVERSIDAD AUTÓNOMA METROPOLITANA Azcapotzalco

Regresar

Estás en la pagina 1 de un total de 69 concursos Futuros

id	nombre	organizador	password	inicio	duracion	intentos	estricto
101		44		2012-09-01 16:59:49	16:59:49	5	5
26		17		2012-09-24 22:35:31	22:35:31	3	3
77		92		2012-10-24 01:53:15	01:53:15	7	7
29		7		2012-11-17 14:49:58	14:49:58	2	2
42		83		2012-12-04 23:16:43	23:16:43	1	1
4		73		2013-01-29 02:55:46	02:55:46	10	10
20		45		2013-02-14 11:23:29	11:23:29	2	2
99		38		2013-02-16 01:33:50	01:33:50	7	7

Figura 4

haciendo click en cualquier id envía a una nueva ventana

Figura 5

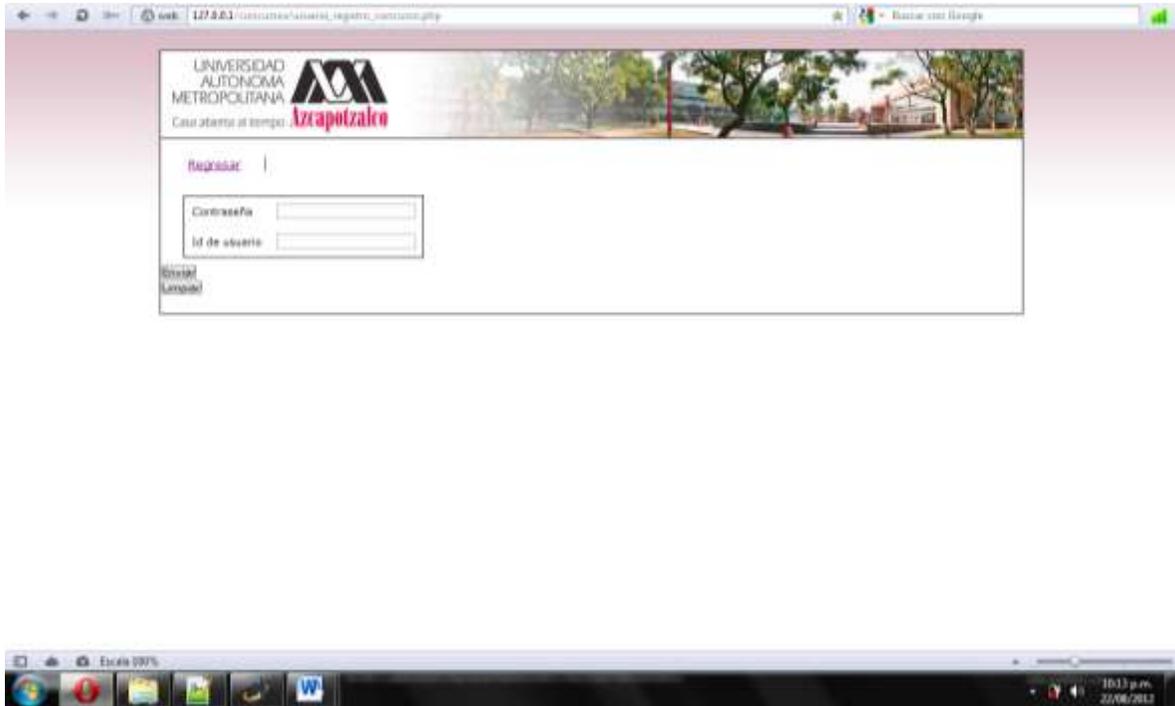


Figura 5 inscripción de un usuario

Para poder inscribir un usuario se necesita mandar la contraseña del concurso a inscribir y el id del usuario, el usuario se inscribe al concurso solo si las contraseñas coinciden, si de alguna forma la contraseña no coincide no el sistema no envía ningún error solo mantiene el formulario limpio de datos y si se llega a aprobar el registro aparecerá un mensaje "usuario registrado con éxito".

De una forma independiente se obtiene el listado de las estadísticas de los concursos (figura 6).



Figura 6

Para poder acceder a una lista de estadísticas solo es necesario ingresar el id del concurso y muestra las estadísticas (figura7 y 8).

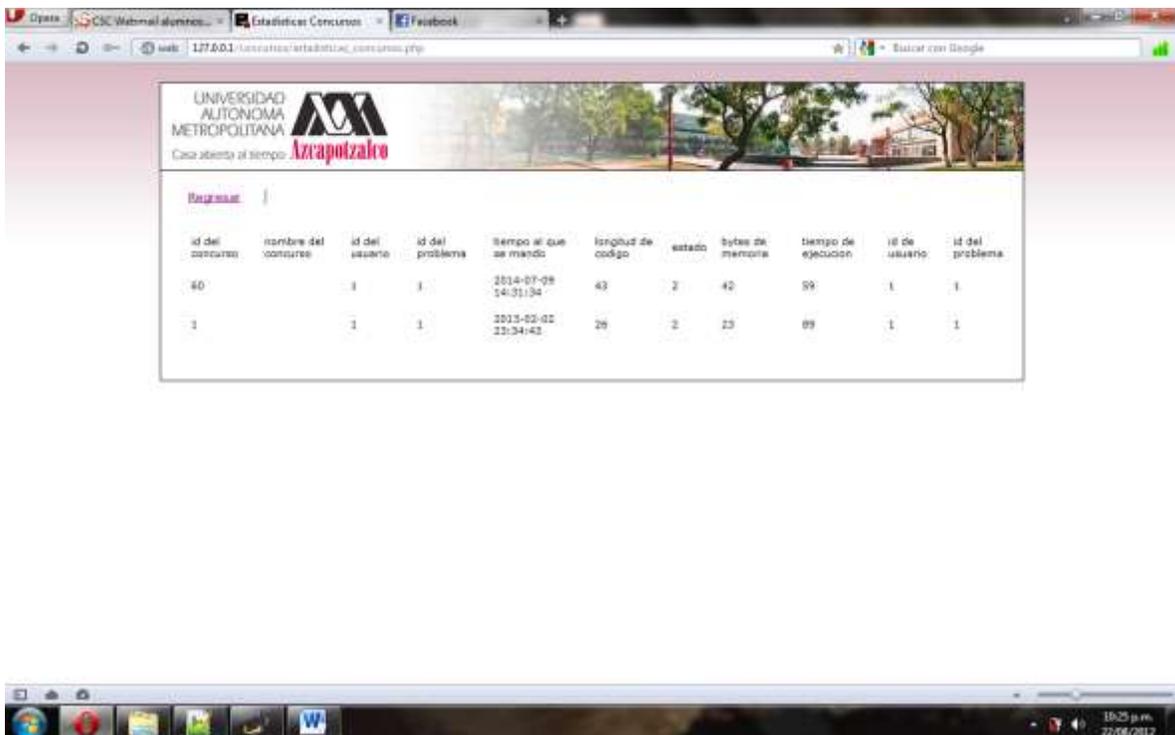


Figura 7

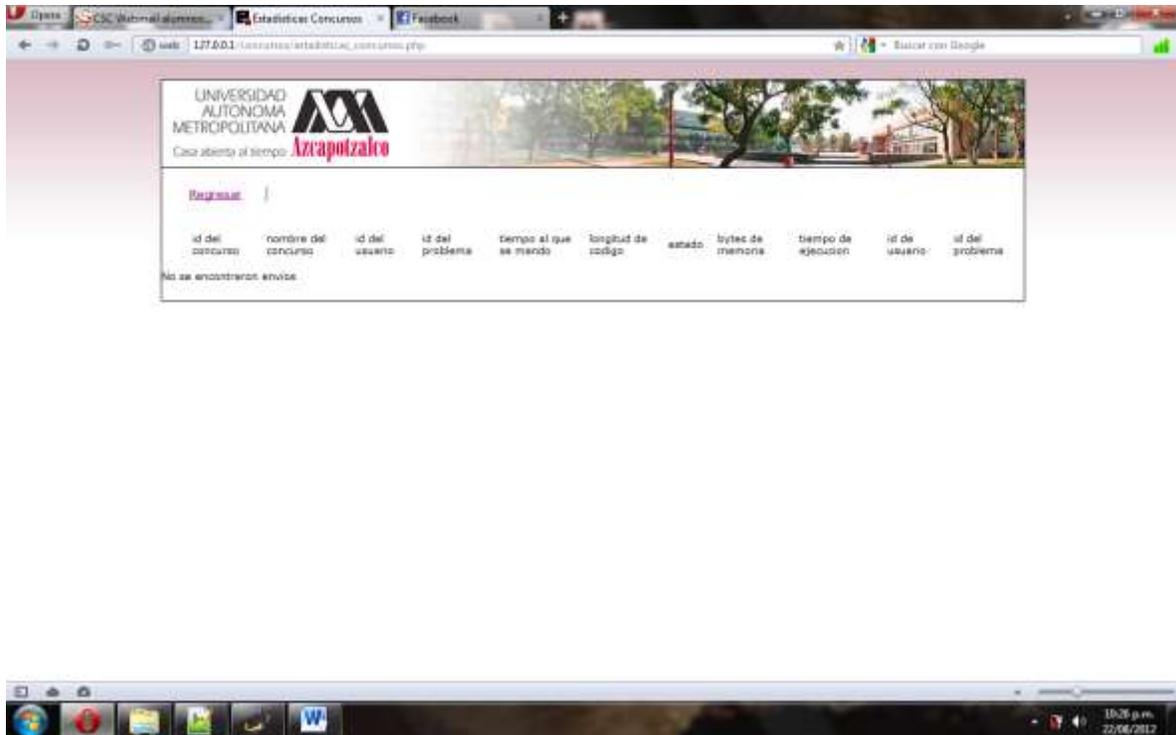


Figura 8

El registro de un concurso se hace solamente realizando el llenado de l formulario (figura 9), si los datos insertados cumplen con el tipo de formato el concurso es agregado de otra manera el concurso no se inserta y el formulario se sigue mostrando.



Figura 9

Al insertar el concurso correctamente se muestra un mensaje de el concurso fue agregado y posteriormente se puede revisar dentro del listado de concursos futuros o presentes (figura 10).

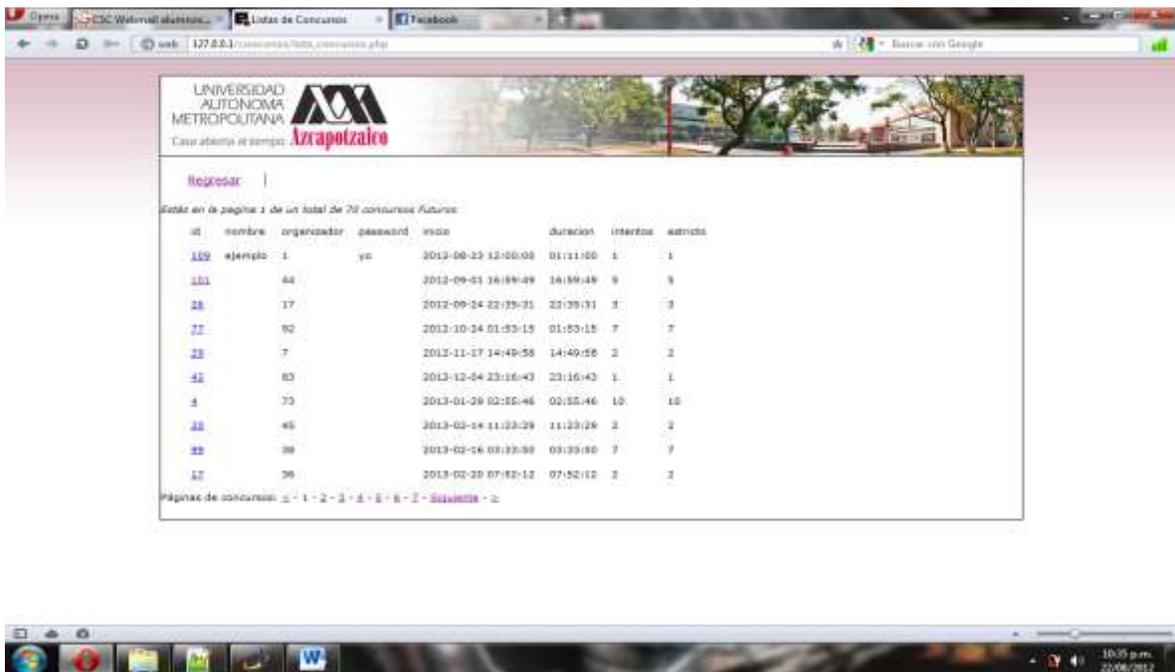


Figura 10

Como se muestra en la figura anterior el concurso se pone al inicio de la lista de concursos futuros debido a que se ordena al mas próximo en tener apertura y este fue agregado con una fecha del día siguiente de la prueba.

La utilidad Diff se utiliza pegando los códigos a comparar dentro de los cuadros de texto y al calcular muestra de una forma tachada lo que se quita para que los códigos sean iguales y subrayada la parte en la que si se pone serian similares (figura 11).

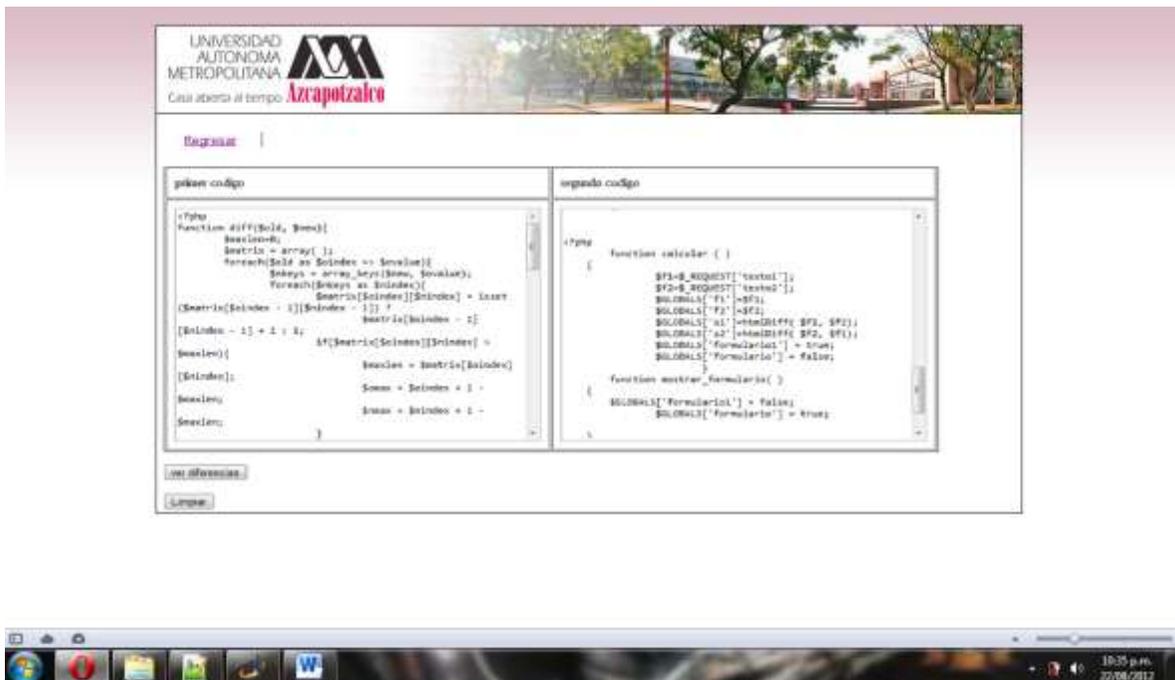


Figura 11

Al mostrar las diferencias se conservan ambos códigos y en la parte posterior se muestran los análisis de el primero contra el segundo y viceversa (figura 12).



Figura 12

REFERENCIAS CONSULTADAS:

TJU ACM-ICPC Online Judge [En línea]. Disponible: <http://acm.tju.edu.cn/toj/>

Teddy Online Judge [En línea]. Disponible: <http://teddy.itc.mx/>

Wikipedia enciclopedia en línea. Disponible: <http://es.wikipedia.org/>

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

REPORTE DE PROYECTO TERMINAL 2

Gestor de exámenes y concursos virtuales para el juez en línea de la UAM Azcapotzalco

Martínez Alcántara Carlos 207300217

Trimestre 12-P
Agosto de 2012

Asesor: M. en C. Rodrigo Alexander Castro Campos

REPORTE DE PROYECTO TERMINAL 2

Gestor de exámenes y concursos virtuales para el juez en línea de la UAM Azcapotzalco.

ACERCA DE ESTE REPORTE:

El presente toca los puntos mencionados en el plan del trabajo de la Propuesta de Proyecto Terminal para la UEA Proyecto Terminal 2:

Trimestre 12-P	1	2	3	4	5	6	7	8	9	10	11
Continuación de la implementación del módulo de organización de concursos											
Implementación del módulo de presentación de la información obtenida mediante el detector de copias de códigos fuente.											
Integración de los módulos del sistema											
Pruebas de cada uno de los módulos implementados											
Generación del reporte final y documentación final (manual de usuario, manual de instalación, etc.)											

SEMANA 1 – 2 – 3 – 4 – 5: Continuación de la implementación del módulo de organización de concursos

Debido a los cambios que se realizaron en el orden de implementación de los módulos en esta sección en lugar de hacer una continuación del modulo se genero la implementación completa de esta sección del Proyecto.

El desarrollo fue tal y como se describió en la propuesta del proyecto, los principales problemas que se presentaron fueron de cierta forma nulos.

La implementación de este modulo requirió de diversos scrpts los cuales se implementaron de forma separada para poder diferenciar las partes que componen dicho modulo, de tal manera los errores manejados como antes ya se menciono fueron escasos, estas dificultades fueron meramente técnicas debido a una mala escritura de las queries o de falla de sintaxis en los códigos.


```
$GLOBALS['formulario'] = true;
}
?>
```

Como se muestra anteriormente en el código se hace un llamado a diversas clases que fueron diseñadas para no implementar en todos los scripts las mismas líneas de los filtros que se utilizan y la conexión a la base de datos.

Inserción_concrso.html

```
<html>
<head>
  <title>Insercion Concurso</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <link rel="stylesheet" type="text/css" href="$RAIZ/css/estilo.css">
  <link rel="stylesheet" type="text/css" href="$RAIZ/css/panel.css">
  <link rel="stylesheet" type="text/css" href="$RAIZ/css/editar.css">
  <script type="text/javascript" src="$RAIZ/javascript/formulario.js"></script>
</head>
<body>
  <div class="cuerpo">
    <div class="header">
      <a href="http://www.azc.uam.mx"></a>
    </div>
    <div class="central">
      <ul class="navegar">
        <li><a href="$RAIZ">Regresar</a>
      </ul>
    </div>
  </body>
  <si condicion="$formulario == true">
    <form action="insercion_concurso.php" method="post">
```

```

        <table cellpadding=10 border align = "l">
            <tr><td>Nombre del concurso</td><td><input type="text"
name="nombre"></td>
            <tr><td>Contraseña</td> <td><input type="password"
name="password"></td>
            <tr><td>Inicio Concurso</td> <td><input type="datetime" name="inicio"></td>
            <tr><td>Duracion </td> <td><input type="time" name="duracion"> horas</td>
            <tr><td>Intentos</td> <td><input type="int"
name="intentos"></td>
            <tr><td>Estricto</td> <td><input type="checkbox"
name="estricto" value="1"></td>

        </table>

        <P> <INPUT type="submit" value="Enviar">
        <P> <INPUT type="reset" value="Limpiar">

    </form>
<si>
    el concurso fue agregado
</si>
    </div>
    </div>
</body>
</html>

```

Registro de usuarios a un concurso

El registro de los usuarios y/o concursantes se inicia desde el modulo de listado de concursos por tal motivo fue que se cambio el desarrollo de los módulos del proyecto.


```

    $GLOBALS['formulario'] = true;
}
?>

```

En este script lo que se hace es después de que el usuario envía los datos desde el formulario que se encuentra dentro del HTML es evaluar los campos para determinar si realmente contienen el tipo de dato que la base de datos admite, así mismo pide por método Get el id del concurso en el cual se va a registrar para obtener el password y comparar con el password solicitado al usuario para aprobar si se agrega el usuario o no.

usuario_registro_concurso.html

```

<html>
<head>
    <title>Registro usuario a concurso</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <link rel="stylesheet" type="text/css" href="$RAIZ/css/estilo.css">
    <link rel="stylesheet" type="text/css" href="$RAIZ/css/panel.css">
    <link rel="stylesheet" type="text/css" href="$RAIZ/css/editar.css">
        <script type="text/javascript" src="$RAIZ/javascript/formulario.js"></script>
</head>
<body>
    <div class="cuerpo">
        <div class="header">
            <a href="http://www.azc.uam.mx"></a>
        </div>
        <div class="central">
            <ul class="navegar">
                <li><a href="$RAIZ">Regresar</a>
            </ul>
        </div>
    </body>
    <si condicion="$formulario == true">

```

```

        <form action="usuario_registro_concurso.php"
method="post">
        <table cellpadding=10 border align = "l">
                <tr><td>Contraseña</td> <td><input type="password"
name="password1"></td>
                <tr><td>Id de usuario</td> <td><input type="int" name="id_usuario"></td>

        </table>

                <P> <INPUT type="submit" value="Enviar">
                <P> <INPUT type="reset" value="Limpiar">

        </form>

<si>
        fue inscrito al concurso correctamente
</si>

        </div>
        </div>
</body>
</html>

```

Tabla de estadísticas

En esta sección hay tres puntos que destacar los cuales son:

- Envíos por usuario: esta será opcional para el administrador del concurso, este designara si se pueden mandar más de una sola vez alguno de los problemas.
- Tiempo de cada envío: tiempo transcurrido desde el inicio del concurso o examen hasta el momento en el que se realiza el envío.
- Ranking: se generará una tabla de posiciones tomando en consideración los intentos fallidos y exitosos de cada uno de los participantes del concurso o examen además de los tiempos de envío.

estadísticas concursos.php

```
<?php
```

```
require_once('includes/entorno.php');
    main::inicia(__FILE__, './', 'html');
    entrada::filtra($_POST, array('concurso', dato::ENTERO_POSITIVO));
    entrada::examina('consultar');
```

```
    entrada::termina('mostrar_formulario');
```

```
        main::termina( );
```

```
?>
```

```
<?php
```

```
function consultar( )
```

```
{
```

```
$pagina = 0;
```

```
$opcion = -1;
```

```
if (isset($_GET['pagina']) && ctype_digit($_GET['pagina'])) {
```

```
    $pagina = $_GET['pagina'];
```

```
}
```

```
$consulta=$_REQUEST['concurso'];
```

```
    $cuantos = 10;
```

```
$offset = $pagina * $cuantos;
```

```
$GLOBALS['pagina'] = $pagina;
```

```
$GLOBALS['cuantos'] = $cuantos;
```

```

$GLOBALS['filas'] = db::query("SELECT SQL_CALC_FOUND_ROWS
C.id,C.nombre,l.usuario,PC.problema,E.* FROM concursos AS C INNER JOIN inscritos
AS I ON C.id=I.concurso INNER JOIN problemas_concurso AS PC ON C.id=PC.concurso
INNER JOIN envios AS E ON (E.usuario=I.usuario AND E.problema=PC.problema)
WHERE C.id=$consulta ORDER BY E.estado DESC,E.tiempo_ejecucion ASC,
E.bytes_memoria ASC, E.longitud_codigo ASC, E.tiempo ASC, E.id ASC LIMIT $offset,
$cuantos")->fetch_all_assoc( );

```

```

$GLOBALS['total'] = db::found_rows( );

```

```

$GLOBALS['formulario'] = false;

```

```

}

```

```

function mostrar_formulario( )

```

```

{

```

```

    $GLOBALS['formulario'] = true;

```

```

}

```

```

?>

```

La consulta de las estadísticas de cada concurso relaciona 4 tablas de la base de datos lo que implica realizar varios joins para realizar el enlace sin problema alguno, para lograr esto se pide al usuario que introduzca el id del concurso dentro de un formulario el cual se muestra en el HTML.

estadisticas_concursos.html

```

<html>

```

```

<interpreta archivo="includes/presentacion/paginado.php">

```

```

<head>

```

```

    <title>Estadísticas Concursos</title>

```

```

    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

```

```

    <link rel="stylesheet" type="text/css" href="$RAIZ/css/estilo.css">

```

```

    <link rel="stylesheet" type="text/css" href="$RAIZ/css/panel.css">

```

```

    <link rel="stylesheet" type="text/css" href="$RAIZ/css/editar.css">

```

```

        <script type="text/javascript" src="$RAIZ/javascript/formulario.js"></script>

```

```

</head>

```

```

<body>

```

```

<div class="cuerpo">
  <div class="header">
    <a href="http://www.azc.uam.mx"></a>
  </div>
  <div class="central">
    <ul class="navegar">
      <li><a href="$RAIZ">Regresar</a>
    </ul>
  </div>
</body>
<si condicion="$formulario == true">
  <form action="estadisticas_concursos.php" method="post">
    <tr><td>id del concurso a buscar</td><td><input
type="text" name="concurso"></td>
    <P> <INPUT type="submit" value="Enviar">
    <P> <INPUT type="reset" value="Limpiar">
  </form>
</si>
<table>
  <tr>
    <td>id del concurso</td>
    <td>nombre del concurso</td>
    <td>id del usuario</td>
    <td>id del problema</td>
    <td>id del envio</td>
    <td>tiempo al que se mando </td>
    <td>longitud de codigo</td>
  </tr>
</table>

```

```

                <td>estado</td>
                <td>bytes de memoria</td>
                <td>tiempo de ejecucion</td>
                <td>id de usuario</td>
            <td> id del problema</td>
        </tr>
    <ciclo var="$filas">
    <tr>
        <td>$id</td>
        <td>$nombre</td>
        <td>$usuario</td>
        <td>$problema</td>
        <td>$id</td>
        <td>$tiempo</td>
        <td>$longitud_codigo</td>
        <td>$estado</td>
        <td>$bytes_memoria</td>
        <td>$tiempo_ejecucion</td>
        <td>$usuario</td>
        <td>$problema</td>

    </tr>
</ciclo>
</table>

    ${paginado('estadisticas_concursos.php', 20, ' - ', 'Páginas de envíos: ', 'No se
encontraron envíos', $total, $cuantos, $pagina)}

    </si>

</body>
</div>

```

```
</div>  
</html>
```

Al término de cada uno de los scripts se realizaron las pruebas correspondientes las cuales arrojaron diversos resultados, en algunos casos fueron erróneos debido a un mal formato de la query los cuales se corrigieron en el momento antes de continuar con alguno el paso siguiente, en el proceso de las pruebas se requirió llenar las tablas utilizadas con valores aleatorios que se generaron con diversos códigos en c y c++ los cuales se mostraran a continuación:

Tabla envios

```
#include<stdlib.h>  
  
#include<iostream>  
  
#include <fstream>  
  
using namespace std;  
  
int main()  
{  
    int dia,mes,anio,hora,min,seg,num,c,ins,edo,num1,num2;  
    int j=1;  
    ofstream fs("insert_into4.txt");  
    srand(time(NULL));  
    for(c=0;c<=100;c++)  
    {  
        num=1+rand()%(101-1);  
        num1=1+rand()%(101-1);  
        num2=1+rand()%(101-1);  
        edo=1+rand()%(3-1);  
        ins=1+rand()%(3-1);  
        dia=1+rand()%(31-1);  
        mes=1+rand()%(13-1);  
        anio=2011+rand()%(2017-2011);
```

```
hora=1+rand()%(25-1);
min=1+rand()%(60-1);
seg=1+rand()%(60-1);
fs<<"insert into envios values (";
fs<<(j+c);
fs<<" , ";
fs<<ins;
fs<<" , ";
fs<<anio;
fs<<" - ";
fs<<mes;
fs<<" - ";
fs<<dia;
fs<<" ";
fs<<hora;
fs<<":";
fs<<min;
fs<<":";
fs<<seg;
fs<<" , ";
fs<<num;
fs<<" , ";
fs<<edo;
fs<<" , ";
fs<<num1;
fs<<" , ";
fs<<num2;
fs<<");";
fs<<endl;
```

```

    }
    fs.close();

}

```

En el código anterior se diseñó para llenar la tabla de envíos la cual se utilizaría para desplegar la lista del Rankin de cada uno de los concursos la generación de este código no tuvo dificultad alguna ya que se trata de números aleatorios el cual se tiene conocimiento desde la UEA de Programación Estructurada.

Tabla inscritos

```

#include<stdlib.h>
#include<iostream>
#include <fstream>
using namespace std;

int main()
{
    int num1,num2,num3,c;
    int j=1;
    ofstream fs("insert_into3.txt");
    srand(time(NULL));
    for(c=0;c<=100;c++)
    {
        num1=1+rand()%(101-1);
        num2=1+rand()%(101-1);
        num3=1+rand()%(101-1);

        fs<<"insert into inscritos values (";
        fs<<(j+c);
        fs<<" ,";
        fs<<num1;

```

```

        fs<<" ";

        fs<<num2;

        fs<<"");";

        fs<<endl;

    }

    fs.close();

}

```

Este código al igual que el anterior fue mas sencillo debido a que solo fueron números aleatorios no importaría el rango al contrario del anterior que se tenia que delimitar a un rango especifico para poder determinar el formato de hora y fecha.

Tabla usuarios

```

#include<stdlib.h>

#include<time.h>

#include<stdio.h>

int main(int c, char** s){

    int i,j;

    FILE *pf;

    j=1;

    pf = fopen("insert_into2.txt","w");

    srand(time(NULL));

    for(i = 0; i < 100; i++){

        fprintf(pf,"insert into usuarios values(
        '%d',null,'%c%c%c%c%c%c%c','%c%c%c%c%c%c%c','%c%c%c%c%c%c%c','%c%c%c%c%c%c%c',
        '%c%c%c%c%c%c%c','1');\n",j+i, 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' +
        rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand()
        % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z'
        - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a')
        + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1),
        'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' +
        rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand()
    )
}

```

```

% (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z'
- 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a')
+ 1), 'a' + rand() % (('z' - 'a') + 1), 'a' + rand() % (('z' - 'a') + 1));
}
return 0;
}

```

En General el código anterior que es el encargado de generar los insert para poder llenar la tabla de usuarios fue de cierta forma el mas complicado de todos debido a que los campos requeridos están conformados por caracteres alfanuméricos, por ello se decidió hacer el cambio de lenguaje de programación para facilitar el trabajo del desarrollador.

Al momento de requerir el llenado de la tabla problemas se genero un código similar al anterior (el cual se muestra mas adelante), solo que se mantuvo el lenguaje original con el que se inicio el desarrollo de estos pequeños programas.

Tabla concursos

```

#include<stdlib.h>
#include<iostream>
#include <fstream>
using namespace std;

int main()
{
    int dia,mes,anio,hora,min,seg,num,c,org;
    int j=1;
    ofstream fs("insert_into.txt");
    srand(time(NULL));
    for(c=0;c<=100;c++)
    {
        num=1+rand()%(11-1);
        org=1+rand()%(101-1);
        dia=1+rand()%(31-1);
        mes=1+rand()%(13-1);
    }
}

```

```
anio=2011+rand()%(2017-2011);
hora=1+rand()%(25-1);
min=1+rand()%(60-1);
seg=1+rand()%(60-1);
fs<<"insert into concursos values (";
fs<<(j+c);
fs<<" ,NULL,"";
fs<<org;
fs<<" ,NULL,"";
fs<<anio;
fs<<"-";
fs<<mes;
fs<<"-";
fs<<dia;
fs<<" ";
fs<<hora;
fs<<":";
fs<<min;
fs<<":";
fs<<seg;
fs<<" ,";
fs<<hora;
fs<<":";
fs<<min;
fs<<":";
fs<<seg;
fs<<" ,";
fs<<num;
fs<<" ,";
```

```
    fs<<num;
    fs<<"");";
    fs<<endl;
}
fs.close();

}
```

Tabla problemas

```
#include<stdlib.h>
#include<iostream>
#include <fstream>
using namespace std;

int main()
{
    int num1,num2,num3,num4,c,autor,archivos,mem,tiem,dif;
    int j=1;
    ofstream fs("insert_into6.txt");
    srand(time(NULL));
    for(c=0;c<=100;c++)
    {
        num1=97+rand()%(123-97);
        num2=97+rand()%(123-97);
        num3=97+rand()%(123-97);
        num4=97+rand()%(123-97);
        autor=1+rand()%(101-1);
        archivos=1+rand()%(101-1);
```

```
mem=1+rand()%(101-1);
tiem=1+rand()%(101-1);
dif=1+rand()%(11-1);
fs<<"insert into problemas values (";
fs<<(j+c);
fs<<" , ";
fs<<char(num1);
fs<<char(num2);
fs<<char(num3);
fs<<char(num4);
fs<<" , ";
fs<<autor;
fs<<" , ";
fs<<char(num3);
fs<<char(num1);
fs<<char(num4);
fs<<char(num2);
fs<<char(num1);
fs<<char(num2);
fs<<char(num3);
fs<<char(num4);
fs<<" , ";
fs<<archivos;
fs<<" , ";
fs<<tiem;
fs<<" , ";
fs<<mem;
fs<<" , ";
fs<<dif;
```

```
    fs<<"");";
    fs<<endl;
}
fs.close();
}
```

Tabla problemas concursos

```
#include<stdlib.h>
#include<iostream>
#include <fstream>
using namespace std;

int main()
{
    int num1,num2,num3,c;
    int j=1;
    ofstream fs("insert_into5.txt");
    srand(time(NULL));
    for(c=0;c<=100;c++)
    {
        num1=1+rand()%(101-1);
        num2=1+rand()%(101-1);
        num3=1+rand()%(101-1);

        fs<<"insert into problemas_concurso values (";
        fs<<num1;
        fs<<" , ";
        fs<<num2;
```

```

        fs<<"");";
    fs<<endl;
}
fs.close();
}

```

Los códigos anteriores imprimían en archivos de texto los datos para a si solo exportar el texto al manejador de base de datos, el manejo de estos códigos en los lenguajes mencionados con anterioridad se eligió debido a la facilidad del manejo de los números y letras aleatorias en cada uno de ellos.

En general los códigos eran sencillos debido a que no requerían un mayor conocimiento al de la UEA antes mencionada, estos códigos fueron desarrollados al momento de requerir el uso de las tablas para la generación de las pruebas.

SEMANA 5 – 6 – 7 – 8: IMPLEMENTACIÓN DEL MÓDULO DE PRESENTACIÓN DE LA INFORMACIÓN OBTENIDA MEDIANTE EL DETECTOR DE COPIAS DE CÓDIGOS

Dentro de este módulo se implemento la utilidad diff el cual está basado en el algoritmo de subsecuencia común más larga.

La utilidad diff es para la comparación de archivos que genera las diferencias entre dos archivos o los cambios realizados en un archivo determinado comparándolo con una versión anterior del mismo archivo. Diff expone los cambios realizados por línea en los archivos de texto.

El algoritmo de subsecuencia mas larga consta en resolver dos secuencias de ítems y se desea encontrar la secuencia más larga de ítems que se presenta las dos secuencias originales en el mismo orden. Esto es, se quiere encontrar una nueva secuencia que pueda obtenerse de la primera secuencia eliminando algunos ítems, y de la segunda secuencia eliminando otros ítems. Se quiere también que esta secuencia sea tan larga como sea posible.

Durante las semanas 5 y 6 fue dedicada a una investigación sobre el algoritmo para poder entender el funcionamiento adecuado y posteriormente pasar a la implementación.

diff.php

```
<?php
```

```
function diff($old, $new){
    $maxlen=0;
```

```

$matrix = array( );
foreach($old as $soindex => $sovalue){
    $nkeys = array_keys($new, $sovalue);
    foreach($nkeys as $nindex){
        $matrix[$soindex][$nindex] = isset($matrix[$soindex - 1][$nindex - 1]) ?
            $matrix[$soindex - 1][$nindex - 1] + 1 : 1;
        if($matrix[$soindex][$nindex] > $maxlen){
            $maxlen = $matrix[$soindex][$nindex];
            $omax = $soindex + 1 - $maxlen;
            $nmax = $nindex + 1 - $maxlen;
        }
    }
}
if($maxlen == 0) return array(array('d'=>$old, 'i'=>$new));
return array_merge(
    diff(array_slice($old, 0, $omax), array_slice($new, 0, $nmax)),
    array_slice($new, $nmax, $maxlen),
    diff(array_slice($old, $omax + $maxlen), array_slice($new, $nmax +
$maxlen)));
}

```

```

function htmlDiff($old, $new){
    $diff = diff(explode(' ', $old), explode(' ', $new));
    $ret = "";
    foreach($diff as $k){
        if(is_array($k))
            $ret .= (!empty($k['d'])?"<del>".implode(' ', $k['d'])."</del> ":"").
                (!empty($k['i'])?"<ins>".implode(' ', $k['i'])."</ins> ":"");
        else $ret .= $k . ' ';
    }
}

```

```
    }  
    return $ret;  
}  
?>
```

```
<?php  
    require_once('includes/entorno.php');  
    main::inicia(__FILE__, './', 'html');  
    entrada::filtra($_POST,array('texto1', dato::CADENA),array('texto2',  
dato::CADENA));  
    entrada::examina('calcular');  
  
    entrada::termina('mostrar_formulario');  
  
?>
```

```
<?php  
function calcular ( )  
{  
    $f1=$_REQUEST['texto1'];  
    $f2=$_REQUEST['texto2'];  
    $GLOBALS['f1']=$f1;  
    $GLOBALS['f2']=$f2;  
    $GLOBALS['s1']=htmlDiff( $f1, $f2);  
    $GLOBALS['s2']=htmlDiff( $f2, $f1);  
    $GLOBALS['formulario1'] = true;
```

```

        $GLOBALS['formulario'] = false;
    }
    function mostrar_formulario( )
    {
        $GLOBALS['formulario1'] = false;
        $GLOBALS['formulario'] = true;

    }
?>

```

El script esta diseñado para poder hacer la comparación de dos códigos poniendo ambos códigos en los cuadros de texto que se presentan a continuación en el código HTML.

Posteriormente este script puede extenderse a elegir archivos específicos desde el equipo de cómputo del usuario, delimitándolo al manejo de las extensiones de cada uno de los lenguajes de programación (c, c++, c#, java , etc.).

diff.html

```

<html>

    <head>
        <style>
            del {
                color: red;
            }

            ins {
                color: green;
            }

        </style>

```

```

        <html>

</html>
    </head>
    <body>
        <head>
            <title>Detector de Copias</title>
            <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
            <link rel="stylesheet" type="text/css" href="$RAIZ/css/estilo.css">
            <link rel="stylesheet" type="text/css" href="$RAIZ/css/panel.css">
            <script type="text/javascript" src="$RAIZ/javascript/formulario.js"></script>
        </head>
        <body>
            <div class="cuerpo">
                <div class="header">
                    <a href="http://www.azc.uam.mx"></a>
                </div>
                <div class="central">
                    <ul class="navegar">
                        <li><a href="$RAIZ">Regresar</a>
                    </ul>
                    <si condicion="$formulario == true">
                        <form action="diff.php" method="post">
                            <table cellpadding=10 border align = "l">
                                <tr><td>primer codigo</td><td>segundo codigo</td> <tr><td><textarea
                                name="texto1" rows="20" cols="60"></textarea>
                                <td><textarea name="texto2" rows="20" cols="60"></textarea></td>

```

```

</table>

<P> <INPUT type="SUBMIT" value="ver diferencias">
<P> <INPUT type="reset" value="Limpiar">

</form>
<si>
  <si condicion="$formulario1 == true">
    <form>
      <table cellpadding=10 border align = "l">
        <tr><td>primer codigo</td>
          <td>segundo codigo</td>
        <tr><td><pre>$f1</td>
          <td><pre>$f2</td>
        <tr><td>comparacion primer codigo contra segundo</td>
          <td>comparacion segundo codigo contra primero</td>
        <tr><td><pre> ${str_replace(array('&lt;del&gt;', '&lt;/del&gt;', '&lt;ins&gt;', '&lt;/ins&gt;'),
array('<del>', '</del>', '<ins>', '</ins>'), $s1)}
          <td>
        <tr><td><pre>${str_replace(array('&lt;del&gt;', '&lt;/del&gt;', '&lt;ins&gt;', '&lt;/ins&gt;'),
array('<del>', '</del>', '<ins>', '</ins>'), $s2)}
          <td>
        </tr>
      </table>
    </form>
  </si>
</si>
</body>
</html>

```

Las pruebas realizadas a este modulo fueron exitosas después de observar que la forma de llamar el texto del formulario debido a que realizar las primeras pruebas se observo que el resultado lo mandaba como nulo cuando el resultado era otro.

SEMANA 9 – 10 – 11: INTEGRACION DE LOS MODULOS DEL SISTEMA

Al finalizar la implementación de todos los scripts se decidió darle un formato en común para fomentar una buena presentación, esto fue con la finalidad de dar un poco de atracción al usuario y a si mismo se viera de una forma llamativa y orientada a la Universidad Autónoma Metropolitana Azcapotzalco.