

21/8/2013



DIVISIÓN DE
CIENCIAS
BÁSICAS E
INGENIERÍA

PROYECTO TERMINAL DE INGENIERÍA EN
COMPUTACIÓN 2

TITULO DEL PROYECTO:

IMPLEMENTACIÓN DEL CATÁLOGO DE PROBLEMAS PARA EL JUEZ EN LÍNEA DE LA
UAM AZCAPOTZALCO

ASESOR: RODRIGO ALEXANDER CASTRO CAMPOS

REPORTE DE PROYECTO TERMINAL 2 | Daniel Capistrán Rodríguez

Resumen

El sistema implementado en el siguiente proyecto es un juez en línea que a diferencia de los jueces en línea que existen, este fue diseñado pensando en los profesores de la UAM-A que impartan materias de programación, a diferencia de los jueces en línea existentes, en este sistema el profesor podrá gestionar problemas para tareas, exámenes y no solo está diseñado para satisfacer las necesidades académicas también se pueden agendar concursos virtuales, otro beneficio que tenemos en este sistema es que se agregó una herramienta donde se facilita la selección de problemas conforme el tema que este viendo el profesor o a las distintas técnicas de programación usadas para resolver los programas, el profesor podrá agregar un problema de su elección especificando las técnicas con las cuales quiere que se resuelvan y dándole un nivel de dificultad para que el usuario se de una idea de lo que se está enfrentando.

El catalogo implementado permite a los usuarios consultar problemas, visualizar estadísticas generadas con respecto a los intentos de solución a los mismos y permite a los profesores realizar búsquedas específicas sobre éstos de modo de que se facilite encontrar un problema que cumpla con ciertos requerimientos o bien, que puedan agregar sus propios problemas.

Tabla de contenido

Resumen	2
Tabla de contenido	3
Objetivos	4
Introducción	4
Desarrollo del Proyecto	
Semana 1, 2 Y 3 – Implementación Del Módulo Que Muestre Diversos Datos Estadísticos En Base A Los Intentos De Solución De Un Problema	5
Semana 4, 5, 6, 7 y 8 – Implementación del módulo que proponga una lista de problemas en base a ciertas restricciones.....	9
Semana 9 y 10 – Integrar los módulos en una interfaz gráfica (sistema)	11
Bibliografía	11
Apéndices	12
Index.html	12
Index.php.....	13
Problemas.html.....	13
Problemas.php	16
Mostrar_enviado.html.....	17
Mostrar_enviado.php	18
Modificar.html.....	18
Modificar.php	22
Queries.html	24
Queries.php	25
Recomendación.html	27
Recomendación.php.....	28
Db_final.sql.....	31

Objetivos

Objetivo general:

Diseñar e implementar un catalogo de problemas de programación para el juez en línea de la UAM Azcapotzalco

Objetivos específicos:

- Diseñar e implementar un módulo que permita crear una lista de problemas de programación
- Diseñar e implementar un módulo que muestre diversos datos estadísticos en base a los intentos de solución de un problema.
- Diseñar e implementar un algoritmo que recomiende una lista de problemas en base a ciertas restricciones

Introducción

La evaluación de exámenes y tareas para cursos de programación generalmente consiste en la resolución de un problema en donde los alumnos deberán diseñar e implementar un algoritmo adecuado. Existen sistemas web llamados jueces en línea que contienen una lista de problemas de programación y permiten la evaluación automática de programas que los resuelvan.

Esta propuesta forma parte de la implementación de un juez en línea para la UAM Azcapotzalco. Dicho proyecto consistirá en varias partes donde una de ellas es la presente propuesta.

En este proyecto se diseñará e implementará la parte que gestiona el catálogo de problemas disponibles en el juez en línea. Además de permitir especificar varios datos y restricciones para cada problema, se implementará un algoritmo que recomiende una lista de problemas en base a los criterios dados.

Desarrollo del proyecto

El presente reporte menciona los siguientes puntos mencionados en el calendario de actividades para realizar la Propuesta de Proyecto Terminal de la UEA Proyecto Terminal 2

Proyecto Terminal 2											
Trimestre 12 – P											
Actividad/Semana	1	2	3	4	5	6	7	8	9	10	11
Implementación del módulo que muestre diversos datos estadísticos en base a los intentos de solución de un problema	■	■	■								
Implementación del módulo que proponga una lista de problemas en base a ciertas restricciones				■	■	■	■	■			
Integrar los módulos en una interfaz gráfica (sistema)									■	■	
Pruebas a cada uno de los módulos desarrollados, aplicaciones del sistema	■	■	■	■	■	■	■	■	■	■	
Elaboración de documentación y Reportes	■	■	■	■	■	■	■	■	■	■	■

Las actividades que se realizaron en el trimestre en curso se desarrollaron en 11 semanas que corresponden a un trimestre en la UAM – AZC en estas 11 semanas se fue desarrollando el Proyecto Terminal 2, y se divide en las actividades que son descritas enseguida:

SEMANA 1, 2 y 3 – Implementación del módulo que muestre diversos datos estadísticos en base a los intentos de solución de un problema.

En estas semanas se implemento el modulo descrito anteriormente, el cual está dividido en las siguientes partes:

- Antes de realizar este modulo se modifico el modulo anterior del proyecto terminal 1 ya que le hacían falta unas modificaciones para que el sistema funcionara de manera optima.
- Re implementación del portal de registro de problemas

El sistema de registro de problemas fue realizado para que el usuario, ya sea administrador o profesor pueda registrar un problema que será útil para algún concurso de programación, tarea de alguna UEA, etc. El cual se almacena en un formulario que previamente se realizo y en el cual se tiene una conexión a una base de datos pública.

En este apartado se realizo el portal web siguiente:



Figura 1. Pagina principal para registro de problemas

Al dar clic a ir a registro de problema, y se desplegara el formulario donde se podrá registrar un nuevo problema por algún profesor o administrador

Formulario	
Título	<input type="text"/>
Descripcion	<input type="text"/>
Limite_tiempo (s)	<input type="text"/>
Limite_memoria (kb)	<input type="text"/>
Dificultad	<input type="text"/>
Tecnicas	<input type="checkbox"/> Cálculos Aritméticos <input type="checkbox"/> Búsquedas <input type="checkbox"/> Programación Dinámica <input type="checkbox"/> Algoritmos Geométricos <input type="checkbox"/> Algoritmo Gloton <input type="checkbox"/> Otros
SubirCodigo	Pincha para adjuntar codigo: <input type="button" value="Seleccionar archivo"/> No se ha seleccionado ningún archivo
Agregar Caso(s) de prueba	<input type="button" value="Agrega"/>
Envia	<input type="button" value="Envia"/>

Figura 2. Formulario para registrar un nuevo problema

En la figura 2 podemos observar como el usuario (administrador, profesor) da de

alta un nuevo problema para su conveniencia ya sea para un concurso o una tarea, en el formulario se muestran los campos ya mencionados anteriormente

Si damos clic a Mostrar lo enviado nos manda a la siguiente pantalla



Figura 3. Lista de problemas que están almacenados en la base de datos pública

En la figura 3 se muestra una lista de problemas que se han agregado a la base de datos pública, y que sirven para los concursos de programación o tareas, ejercicios de diferentes UEA's, se agregaron más problemas para hacer pruebas al módulo que muestra datos en base a ciertas restricciones.

Al dar clic a alguno de los problemas agregados anteriormente podemos editar el mismo problema, y nos manda a la siguiente pantalla.

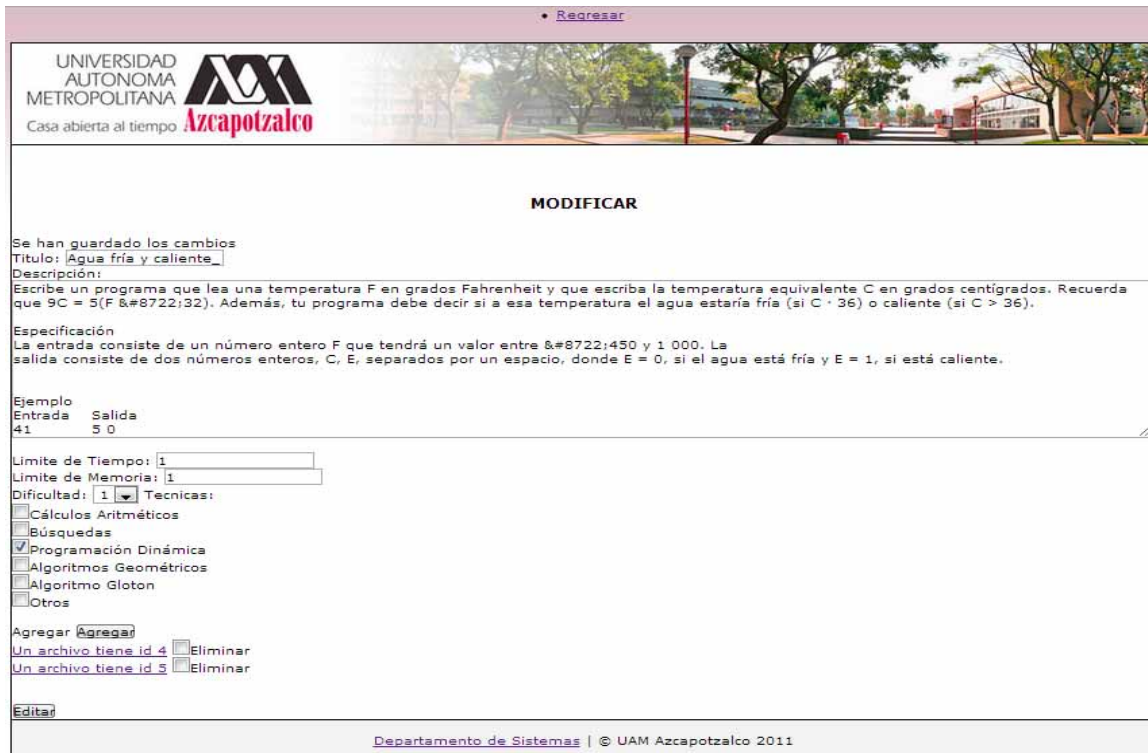


Figura 4. Modificar

En la figura 4 se muestra una pantalla similar al formulario en el cual se agregan los problemas, solo que aquí tenemos lo ya cargado previamente y lo podemos editar, también podemos eliminar el código fuente o agregar uno más.

- ❖ Creación de la tabla *archivos* en la base de datos temp ya diseñada previamente se agrego la tabla archivos donde se almacenaran los casos de prueba de cada uno de los problemas que se encuentren en la base de datos.

```
DROP TABLE IF EXISTS `archivos`;
CREATE TABLE `archivos` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `contenido` varchar(255) NOT NULL,
  `problema` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`problema`) REFERENCES `problemas` (`id`) ON DELETE
  CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

The screenshot shows the phpMyAdmin interface for the 'temp' database. The table 'archivos' is selected, and its structure is displayed in a table format. Below the structure table, the 'indices' section shows two keys: a PRIMARY key on the 'id' column and a key on the 'problema' column.

Columna	Tipo	Null	Predeterminado	Enlaces a	Comentarios
id	int(11)	No			
contenido	varchar(255)	No			
problema	int(11)	No		problemas -> id	

Nombre de la clave	Tipo	Unico	Empaquetado	Columna	Cardinalidad	Cotejamiento	Null	Comentario
PRIMARY	BTREE	Si	No	id	4	A		
problema	BTREE	No	No	problema	4	A		

- ❖ Se diseño de formulario para consulta de datos en base a ciertos criterios, este formulario se realizo en los lenguajes HTML y PHP para que el administrador, o profesor, puedan consultar ciertos datos como son:
 - Cuántos envíos hay en el problema que tiene un id ingresado por el usuario.
 - Cuántos usuarios diferentes tienen estado 10 con un id ingresado por el usuario.
 - Calcula el promedio de longitud de código con el id ingresado por el usuario.

En la pantalla siguiente se muestra el formulario donde se muestran datos en base a lo anterior.

UNIVERSIDAD AUTÓNOMA METROPOLITANA Azcapotzalco

• Registrar

UNIVERSIDAD AUTÓNOMA METROPOLITANA Azcapotzalco

Casa abierta al tiempo

Consultas en base a:

¿Cuántos envíos son del problema con el siguiente id?

Enviar

¿Cuántos usuarios diferentes tienen estado 10 con el siguiente id?

Enviar

Calcula el promedio de longitud de código con el siguiente id?

Enviar

Departamento de Sistemas | © UAM Azcapotzalco 2012

Figura 5. Consulta en base a:

En la figura 5 se muestra el formulario donde se pide una serie de consultas en base a los criterios que se mencionaron anteriormente

SEMANA 4, 5, 6, 7 y 8 – Implementación del módulo que proponga una lista de problemas en base a ciertas restricciones.

En estas semanas se implemento el modulo descrito anteriormente, el cual está dividido en las siguientes partes:

- ❖ Diseño de formulario para proponer problemas: este formulario se realizo en los lenguajes HTML y PHP para que el administrador o profesor, puedan tener un modulo que les permita generar una lista de problemas de programación en base a ciertas restricciones que son delimitadas por el usuario.
- ❖ El formulario fue diseñado con los siguientes requerimientos:
 - ¿Cuántos problemas quieres?

En esta pregunta el sistema se delimito en un intervalo de 1 a 10 donde como máximo puedes tener 10 problemas y mínimo 1.
 - ¿Promedio de dificultad?

En esta pregunta se toma una dificultad de entre 1y 2, 2y 3, 3y 4 y como máximo en dificultad seria entre 4 y 5 para que al momento que se pide el nivel de dificultad se haga un promedio entre estos intervalos.
 - Y por último, se realizo una serie de preguntas en base a las diferentes técnicas que existen en el mismo formulario de generar un problema.

- A. ¿Cuántos problemas que se resuelva con Cálculos Aritméticos?
- B. ¿Cuántos problemas que se resuelva con Búsquedas?
- C. ¿Cuántos problemas que se resuelva con Programación Dinámica?
- D. ¿Cuántos problemas que se resuelva con Algoritmos Geométricos?
- E. ¿Cuántos problemas que se resuelva con Algoritmo Glotón?

No se mostro la opción “otros” ya que si el usuario está indicando que quiere una lista de problemas debe tener bien definido lo que pide y no puede poner, quiero una lista con otros problemas de programación, ya que el sistema fue acotado solo a las técnicas mencionadas en la propuesta de proyecto terminal.

The screenshot shows a web page header for Universidad Autónoma Metropolitana Azcapotzalco. Below the header is a form titled "Recomendacion de problemas en base a:". The form contains several input fields and dropdown menus:

- ¿Cuántos problemas quieres?: A dropdown menu with the value "1" selected.
- ¿Promedio de dificultad?: A dropdown menu with the value "Entre 1 y 2" selected.
- ¿Cuántos problemas que se resuelva con Cálculos Aritméticos?: An empty text input field.
- ¿Cuántos problemas que se resuelva con Búsquedas?: An empty text input field.
- ¿Cuántos problemas que se resuelva con Programación Dinámica?: An empty text input field.
- ¿Cuántos problemas que se resuelva con Algoritmos Geométricos?: An empty text input field.
- ¿Cuántos problemas que se resuelva con Algoritmo Glotón?: An empty text input field.

At the bottom of the form is an "Enviar" button. The footer of the page reads "Departamento de Sistemas | © UAM Azcapotzalco 2012".

Figura 6.
Recomendacion de
problemas en base a:

En la figura 6 se muestra la interfaz donde el usuario solicitara la lista de problemas de programación en base a las restricciones que se muestran.

En los campos donde se piden cuantos problemas de cada técnica Cálculos Aritméticos, Búsquedas, Programación Dinámica, Algoritmos Geométricos y Algoritmo Glotón se utilizo un script de java para que al momento de ingresar los datos se ingresaran solo números, la función utilizada es: `onKeyPress = "return RANGO (this, 0, null, event)`

Y se necesita meter el script siguiente:

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

SEMANA 9 y 10 – Integrar los módulos en una interfaz gráfica (sistema)

En las últimas semanas se realizaron las pruebas necesarias para que el sistema funcionara como se describió en la propuesta, estas y cada una de las pruebas se hicieron al término de cada uno de los módulos y cuando se concluyo el proyecto, para que al momento de llegar a la fase final, no hubiera ningún problema en el funcionamiento del sistema y así se pudieran integrar todos y cada uno de los módulos a la interfaz grafica.

Bibliografía

- Pavón, J., 2005, Navegar en internet: creación de un portal con PHP y MySQL, Alfaomega.
- López, J. 2008. Domine HTML y DHTML, Alfaomega.
- López, J., 2008, Domine PHP 5, Alfaomega.
- Welling, L., 2005, Desarrollo Web con PHP y MySQL, Anaya.
- <http://php.net/index.php>, sitio principal de PHP.
- <http://www.MySQL.com>, sitio principal de MySQL.

Apéndices

Listado de código fuente

Index.html

```
<html>
  <head>
    <title>Juez en linea</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/index.css">
  </head>
  <body>
    <div class="cuerpo">
      <div class="header">
        <a href="http://www.azc.uam.mx"></a>
      </div>
      <br>
      <br>
      <div ALIGN=center class="bienvenida">
        <p> Bienvenidos al juez en linea </p>
      </div>
      <br>
      <br>
      <div ALIGN=center class="agregar">
        <li><a href="$RAIZ/problemas.php">Ir registro de problema</a>
      </div>
      <div class="footer">
        <a href="http://sistemas.azc.uam.mx">Departamento de Sistemas</a> | © UAM
      </div>
    </div>
  </body>
</html>
```

Index.php

```
<?php
require_once('includes/entorno.php');
main::inicia(__FILE__, './', 'html');

main::termina( );
?>
```

Problemas.html

```
<html>
  <head>
    <title>Juez en linea</title>
    <style>
      table {
        border: 1px solid black;
      }

      tr {
        width: 200px;
        height: 20px;
        border: 1px solid black;
      }
    </style>
    <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/index.css">
  </head>
  <br><li><a href="$RAIZ">Regresar</a>
<li><a href="$RAIZ/mostrar_enviado.php">Mostrar lo enviado</a><br>
  <li><a href="$RAIZ/queries.php">Consultas en base a :</a><br>
  <li><a href="$RAIZ/recomendacion.php">Recomendacion</a><br>
  <body>
  <div class="cuerpo">
  <div class="header">
    <a href="http://www.azc.uam.mx"></a>
  </div>
```

```

                <div ALIGN=center class="bienvenida">
                    <p> Formulario </p>
                </div>
<si condicion="$formulario == true">
    <div ALIGN=center>
        <form method="post" enctype="multipart/form-data">
            <table cellspacing=5 border=3 border align = "1" valign="center" width="30%" height="60%"
>
                <tr><td>Titulo</td><td><input type="text" name="titulo"></td>
            <tr><td>Descripcion</td> <td><textarea name="descripcion" rows="10"
cols="50"></textarea></td>
            <tr><td>Limite_tiempo (s)</td> <td><input type="text" name="limite_tiempo"
onKeyPress="return RANGO(this, 0, null, event)"></td>
            <tr><td>Limite_memoria (kb)</td> <td><input type="text" name="limite_memoria"
onKeyPress="return RANGO(this, 0, null, event)"></td>
            <tr>
                <td>Dificultad</td>
                <td>
                    <select name="dificultad">
                        <option value="1">1
                        <option value="2">2
                        <option value="3">3
                        <option value="4">4
                        <option value="5">5
                    </select>
                </td>
            </tr><!-- select -->
            <tr>
                <td>Tecnicas</td>
                <td>
                    <input type="checkbox" name="tecnicas[]" value="1">Cálculos Aritméticos<br>
                    <input type="checkbox" name="tecnicas[]" value="2">Búsquedas<br>
                    <input type="checkbox" name="tecnicas[]" value="3">Programación Dinámica<br>
                    <input type="checkbox" name="tecnicas[]" value="4">Algoritmos Geométricos<br>
                    <input type="checkbox" name="tecnicas[]" value="5">Algoritmo
                    <input type="checkbox" name="tecnicas[]" value="6">Otros<br>
                </td>
            </tr>
        </form>
    </div>
</si>

```

```

        <tr>
        <td><h1>SubirCodigo</h1></td>
        <!--<td> <input type="button" value="Agregar codigo" onClick = "agrega( )" >
    </td>-->
        <td>Pincha para adjuntar codigo:<br><br><input type="file" name="codigo"
/> </td>
        <!--<td><li><a href="$RAIZ/mostrar_enviado.php ">Mostrar lo
enviado</a></td>-->
    </tr>
    <tr>
    <td>Agregar Caso(s) de prueba</td>
    <td> <input type="button" value="Agregar" onClick="agrega( )"><br> </td>
</tr>
<tr>
<td></td>
<td>
    <table id="formas">
    </table>
</td>
</tr>
<tr><td COLSPAN=0> <INPUT type="submit" value="Enviar"></td></tr>
</table>
</form>
<si>
    Agregamos el problema a la base de datos
</si>
<script type="text/javascript" src="$RAIZ/javascript/input.js"></script>
<div class="footer">
    <a href="http://sistemas.azc.uam.mx">Departamento de Sistemas</a> | © UAM
Azcapotzalco 2012
    </div>
</div>
</div>
</body>
<script>
    // http://www.w3schools.com/jsref/met_table_insertrow.asp
    // http://www.w3schools.com/jsref/met_table_row_insertcell.asp

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

```

```

// onKeyPress="return RANGO(this, 0, 200, event)"

function agrega( )
{
    var tabla = document.getElementById("formas")
    var fila = tabla.insertRow(-1)
    var celda = fila.insertCell(-1)

    celda.innerHTML = "<input type='file' name='archivo[]'>"
}
</script>
</html>

```

Problemas.php

```

<?php
require_once('includes/entorno.php');
main::inicia(__FILE__, './', 'html');

    entrada::filtra($_POST, array('titulo', dato::CADENA), array('descripcion', dato::CADENA),
array('limite_tiempo', dato::NATURAL), array('limite_memoria', dato::NATURAL), array('dificultad',
dato::RANGO(dato::ENTERO, 1, 5)), array('tecnicas', dato::ARREGLO(dato::RANGO(dato::ENTERO, 1, 6)), array(
)));
    entrada::filtra($_FILES, array('codigo', dato::ARCHIVO));
    entrada::filtra(upload::desempaca('archivo'), array('archivo', dato::ARREGLO(dato::ARCHIVO), array(
)));
    entrada::examina('insertar');

    entrada::termina('mostrar_formulario');
?>

<?php
function insertar( )
{
    db::query("INSERT INTO problemas (titulo, descripcion, limite_tiempo, limite_memoria, dificultad)
VALUES ('", $_REQUEST['titulo'], "', '", $_REQUEST['descripcion'], "', '", $_REQUEST['limite_tiempo'], "', '",
$_REQUEST['limite_memoria'], "', '", $_REQUEST['dificultad'], "')");
    $id = db::query("SELECT LAST_INSERT_ID( )->fetch_value( );

```



```

foreach ($_REQUEST['tecnicas'] as $tecnica) {
echo "el problema ", $id, " se puede resolver con la tecnica ", $tecnica, '<br>';
db::query("INSERT INTO tecnicas (problema, tecnica) VALUES ($id, $tecnica)");
}
//-----
foreach ($_REQUEST['archivo'] as $archivo) {
echo "el problema ", $id, " tiene el siguiente archivo(s)<br>";
db::query("INSERT INTO archivos (contenido, problema) VALUES (", file_get_contents($archivo-
>ruta( )), ", ", $id, ")");
}
//-----

rename($_REQUEST['codigo']->ruta( ), "problemas/$id.cpp");
$GLOBALS['formulario'] = false;
}

function mostrar_formulario( )
{
    $GLOBALS['formulario'] = true;
}
?>

```

Mostrar_enviado.html

```

<html>
<head>
<title>Muestra Datos</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/index.css">
<br><li><a href="$RAIZ/problemas.php">Regresar</a><br>
</head>
<body>
<div class="cuerpo">
<div class="header">
<a href="http://www.azc.uam.mx"></a>
</div>

```

```

<br>
<br>
    <div ALIGN=center class="Bienvenida">
    <br><br><p>MUESTRA LOS DATOS</p><br>
        <ciclo var="$problemas">
            <a href="modificar.php?id=$id">$titulo</a><br>
        </ciclo>
    </div>
    <br>
    <br>
    <div class="footer">
        <a href="http://sistemas.azc.uam.mx">Departamento de Sistemas</a> | © UAM
Azcapotzalco 2011
    </div>
</div>
</body>
</html>

```

Mostrar_enviado.php

```

<?php
require_once('includes/entorno.php');
main::inicia(__FILE__, './', 'html');

$GLOBALS['problemas'] = db::query("SELECT * FROM `problemas`")->fetch_all_assoc( );
main::termina( );
?>

```

Modificar.html

```

<html>
    <head>
        <title>Modificar</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/index.css">
        <li><a href="$RAIZ/mostrars_enviado.php">Regresar</a>
    </head>

```

```

<body>
  <div class="cuerpo">
    <div class="header">
      <a href="http://www.azc.uam.mx"></a>
    </div>
    <div ALIGN=center class="Bienvenida">
      <br><br><p>MODIFICAR</p><br>
    </div>
    <si condicion="isset($actualizado)">
      Se han guardado los cambios
    </si>
    <div ALIGN=left>
      <tr><td> <form method="post" enctype="multipart/form-data">
        Titulo: <input type="text" name = "titulo" value="$problema->titulo"><br>

        Descripción: <br> <textarea name="descripcion" rows="10" cols="50">$problema-
>descripcion</textarea><br>
        $problema->autor<br>
        Limite de Tiempo: <input type="text" name="limite_tiempo" value="$problema-
>limite_tiempo"><br>
        Limite de Memoria: <input type="text" name="limite_memoria" value="$problema-
>limite_memoria"><br>
        Dificultad:
        <select name="dificultad" id="dificultad">
          <option value="1">1
          <option value="2">2
          <option value="3">3
          <option value="4">4
          <option value="5">5
        </select>

        Tecnicas:<br>
        <input type="checkbox" id="c1" name="tecnicas[]" value="1">Cálculos Aritméticos<br>
        <input type="checkbox" id="c2" name="tecnicas[]" value="2">Búsquedas<br>
        <input type="checkbox" id="c3" name="tecnicas[]" value="3">Programación Dinámica<br>
        <input type="checkbox" id="c4" name="tecnicas[]" value="4">Algoritmos
Geométricos<br>
        <input type="checkbox" id="c5" name="tecnicas[]" value="5">Algoritmo Gloton<br>
        <input type="checkbox" id="c6" name="tecnicas[]" value="6">Otros<br><br>

```

```

<!--<si condicion="$problema->archivos > 0">
  <ciclo var="range(1, $problema->archivos)">
    <!-- -----se acaba de hacer-----
    se quito el checkbox de esta parte y se movio adelante del boton agregar
    <!-- -----
----- -->

    <li><a href="$RAIZ/problemas/$PADRE->problema->id/ent$ACTUAL.txt">$ACTUAL</a>
  </ciclo>
</si>
</tr>
  <!-- -----se acaba de hacer-----
->
<!-- aqui agregamos un archivo nuevo y ponemos el checkbos para eliminar el archivo existente-> se acaba de
hacer-->
    <tr>
      <td>Agregar</td>
      <td> <input type="button" value="Agregar" onClick="agrega ( )"><br>      </td>
    <ul>
      <ciclo var="$archivos">
        <li><a href="archivo.php?id=$id">Un archivo tiene id $id</a>
<input type="checkbox" id="c1" name="eliminar[]" value="$id">Eliminar<br>
        <!-- -->
      </ciclo>
      <!--<li><a href="$RAIZ/problemas.php">Ir registro de problema</a>--
>
    </ul>
  </tr>
<tr>
  <td></td>
  <td>
    <table id="formas">
    </table>
  </td>
</tr>
<!-- -----
- -->

<script>
  <ciclo var="$tecnicas">

```

```
                document.getElementById("c$tecnica").checked = true
            </ciclo>
        </script>
        <br><br><tr><td colspan=0> <input type="submit" value="Editar"></td></tr>
        <br>
    </form>
    </div>
    <div class="footer">
        <a href="http://sistemas.azc.uam.mx">Departamento de Sistemas</a> | © UAM
Azcapotzalco 2011
    </div>
</div>
</body>
<script>
    document.getElementById("dificultad").selectedIndex = $problema->dificultad - 1
</script>
    <!-- ***** -->

    <!-- con este script funciona el boton agregar -> se acaba de hacer -->
<script>
    function agrega ( )
    {
        var tabla = document.getElementById("formas")
        var fila = tabla.insertRow(-1)
        var celda = fila.insertCell(-1)

        celda.innerHTML = "<input type='file' name='archivo[]'"
    }
</script>
    <!-- ***** -->

</html>
```

Modificar.php

```
<?php
require_once('includes/entorno.php');
main::inicia(__FILE__, './', 'html');

entrada::requiere($_GET, array('id', dato::NATURAL));
entrada::filtra($_POST,
    array('titulo', dato::CADENA),
    array('descripcion', dato::CADENA),
    array('limite_tiempo', dato::NATURAL),
    array('limite_memoria', dato::NATURAL),
    array('dificultad', dato::RANGO(dato::ENTERO, 1, 5)),
    array('tecnicas', dato::ARREGLO(dato::RANGO(dato::ENTERO, 1, 6)), array( )),
    array('eliminar', dato::ARREGLO(dato::ENTERO), array(  ))
);
entrada::filtra(upload::desempaca('archivo'), array('archivo', dato::ARREGLO(dato::ARCHIVO), array(
)));
//var_dump($_REQUEST);
//var_dump(entrada::$tabla);
entrada::examina('editar', 'mostrar_formulario');

entrada::termina('mostrar_formulario');
?>

<?php
// http://sites.google.com/site/rccuam

function editar( )
{
    db::query('DELETE FROM tecnicas WHERE problema = ', $id);

    db::query("UPDATE problemas SET titulo = ", $_REQUEST['titulo'], ", descripcion = ",
$_REQUEST['descripcion'], ", limite_tiempo = ", $_REQUEST['limite_tiempo'], ", limite_memoria = ",
$_REQUEST['limite_memoria'], ", dificultad = ", $_REQUEST['dificultad'], " WHERE id = ", $_REQUEST['id']);
    $id = $_REQUEST['id'];
}
```

```

//var_dump($_REQUEST);

foreach ($_REQUEST['archivo'] as $archivo) {
echo "el problema ", $id, " tiene el siguiente archivo(s)<br>";
db::query("INSERT INTO archivos (contenido, problema) VALUES (", file_get_contents($archivo-
>ruta( )), ", ", $id, ")");
}

foreach ($_REQUEST['tecnicas'] as $tecnica) {
echo "el problema ", $id, " se puede resolver con la tecnica ", $tecnica, '<br>';
db::query("INSERT INTO tecnicas (problema, tecnica) VALUES ($id, $tecnica)");
}

foreach ($_REQUEST['eliminar'] as $temp) {
echo "el elemento $i del arreglo es", $_REQUEST['eliminar'], "<br>";
db::query("DELETE FROM archivos WHERE id = ", $temp);
}
/*for ($i = 0; $i < alumnos_profesor_ibfk_1($_REQUEST['eliminar']); ++$i) {
echo "el elemento $i del arreglo es", $_REQUEST['eliminar'][$i], "<br>";
}*/

$GLOBALS['actualizado'] = true;
}

function mostrar_formulario( )
{
$GLOBALS['problema'] = db::query("SELECT id, titulo, autor, descripcion, limite_tiempo,
limite_memoria, dificultad FROM problemas WHERE id = ", $_REQUEST['id'])->fetch_required_assoc( );
$GLOBALS['tecnicas'] = db::query("SELECT tecnica FROM tecnicas WHERE problema = ",
$_REQUEST['id'])->fetch_all_assoc( );
$GLOBALS['archivos'] = db::query("SELECT id FROM archivos WHERE problema = ", $_REQUEST['id'])->
>fetch_all_assoc( );
}
}
?>

```

Queries.html

```
<html>
  <head>
    <title>Juez en linea</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/index.css">
    <br><li><a href="$RAIZ/Problemas.php">Regresar</a>
  </head>
  <body>
    <div class="cuerpo">
      <div class="header">
        <a href="http://www.azc.uam.mx"></a>
      </div>
      <br>
      <br>
      <div ALIGN=center class="bienvenida">
        <p> Consultas en base a: </p>
      </div>
      <tr><td> <form method="post" enctype="multipart/form-data">
      <div ALIGN=center><table align="center" bottom="middle" cellspacing=2 border=1
width="38%" height="10%" >
      <tr><td>¿Cuántos envíos son del problema con el siguiente id?: </td><td><input
type="text" name="id0"><!--"Cuantos_problemas"--></td>
      <tr><td colspan=0> <INPUT type="submit" value="Enviar"></td></tr>
      </table></div>
      <br><br>
      <si condicion="isset($q0)">
      <p>1.- Dado el problema $id0, ¿cuántos envíos son de ese problema?</p>$q0
      </si>
      </form>

      <tr><td> <form method="post" enctype="multipart/form-data">
      <div ALIGN=center><table align="center" bottom="middle" cellspacing=2 border=1
width="38%" height="10%" >
```



```

        <tr><td>¿Cuántos usuarios diferentes tienen estado 10 con el siguiente id?:
</td><td><input type="text" name="id1">
        <tr><td colspan="2"> <input type="submit" value="Enviar"></td></tr>
        </table></div>
        <br><br>
        <si condicion="isset($q1)">
            <p>2.- Dado el problema $id1, ¿cuántos usuarios diferentes tienen estado = 10?
UNIQUE</p>$q1
        </si>
        </form>

        <tr><td>
        <form method="post" enctype="multipart/form-data">
        <div align="center"><table align="center" bottom="middle" cellspacing=2 border=1
width="38%" height="10%" >
        <tr><td>Calcula el promedio de longitud de código con el siguiente id?:
</td><td><input type="text" name="id2"><!--"Cuantos problemas"--></td>
        <tr><td colspan="2"> <input type="submit" value="Enviar"></td></tr>
        </table></div>
        <br><br>
        <si condicion="isset($q2)">
            <p>1.- Dado el problema $id2, ¿Calcula el promedio de longitud de
codigo?</p>$q2
        </si>
        </form>
        <div class="footer">
        <a href="http://sistemas.azc.uam.mx">Departamento de Sistemas</a> | © UAM
Azcapotzalco 2012
        </div>
        </div>
        </body>
</html>

```

Queries.php

```

<?php
require_once('includes/entorno.php');
main::inicia(__FILE__, './', 'html');

```

```

entrada::filtra($_POST, array('id0', dato::ENTERO));
entrada::examina('peticion0');

entrada::filtra($_POST, array('id1', dato::ENTERO));
entrada::examina('peticion1');

entrada::filtra($_POST, array('id2', dato::ENTERO));
entrada::examina('peticion2');

entrada::termina( );
/*
dado el problema I, ¿cuántos envíos son de ese problema?(cuantos envios tienen el mismo problema)
dado el problema I, ¿cuántos usuarios diferentes tienen estado = 10? UNIQUE
dado el problema I, calcular el promedio de longitud_codigo AVERAGE AVG
*/

function peticion0( )
{
    $GLOBALS['q0'] = db::query("SELECT alumnos_profesor_ibfk_1(*) FROM envios WHERE problema = ",
$_REQUEST['id0'])->fetch_value( );
    $GLOBALS['id0'] = $_REQUEST['id0'];
}

function peticion1( )
{
    $GLOBALS['q1'] = db::query("SELECT alumnos_profesor_ibfk_1(*) FROM (SELECT DISTINCT usuario FROM
envios WHERE estado = 10 AND problema = ",$_REQUEST['id1'], ") AS temp")->fetch_value( );
    $GLOBALS['id1'] = $_REQUEST['id1'];
}

function peticion2( )
{
    $GLOBALS['q2'] = db::query("SELECT AVG(longitud_codigo) FROM envios WHERE problema = ",
$_REQUEST['id2'])->fetch_value( );
    $GLOBALS['id2'] = $_REQUEST['id2'];
}

/*
    $GLOBALS['q1'] = db::query("SELECT alumnos_profesor_ibfk_1(*) FROM (SELECT DISTINCT usuario FROM
envios WHERE estado = 10 AND problema = ",$_REQUEST['id1'], ") AS temp")->fetch_value( );

```

```

        $GLOBALS['q2'] = db::query("SELECT AVG(longitud_codigo) FROM envios WHERE problema = ",
$_REQUEST['id2']->fetch_value( );

        $GLOBALS['peticiones'] = true;
    }*/
?>

```

Recomendación.html

```

<?php
require_once('includes/entorno.php');
main::inicia(__FILE__, './', 'html');

entrada::filtra($_POST, array('id0', dato::ENTERO));
entrada::examina('peticion0');

entrada::filtra($_POST, array('id1', dato::ENTERO));
entrada::examina('peticion1');

entrada::filtra($_POST, array('id2', dato::ENTERO));
entrada::examina('peticion2');

entrada::termina( );
/*
    dado el problema I, ¿cuántos envios son de ese problema?(cuantos envios tienen el mismo problema)
    dado el problema I, ¿cuántos usuarios diferentes tienen estado = 10? UNIQUE
    dado el problema I, calcular el promedio de longitud_codigo AVERAGE AVG
*/

function peticion0( )
{
    $GLOBALS['q0'] = db::query("SELECT alumnos_profesor_ibfk_1(*) FROM envios WHERE problema = ",
$_REQUEST['id0']->fetch_value( );
    $GLOBALS['id0'] = $_REQUEST['id0'];
}

function peticion1( )
{

```

```

        $GLOBALS['q1'] = db::query("SELECT alumnos_profesor_ibfk_1(*) FROM (SELECT DISTINCT usuario FROM
envios WHERE estado = 10 AND problema = ",$_REQUEST['id1'], ") AS temp")->fetch_value( );
        $GLOBALS['id1'] = $_REQUEST['id1'];
    }

    function peticion2( )
    {
        $GLOBALS['q2'] = db::query("SELECT AVG(longitud_codigo) FROM envios WHERE problema = ",
$_REQUEST['id2'])->fetch_value( );
        $GLOBALS['id2'] = $_REQUEST['id2'];
    }

    /*
        $GLOBALS['q1'] = db::query("SELECT alumnos_profesor_ibfk_1(*) FROM (SELECT DISTINCT usuario FROM
envios WHERE estado = 10 AND problema = ",$_REQUEST['id1'], ") AS temp")->fetch_value( );
        $GLOBALS['q2'] = db::query("SELECT AVG(longitud_codigo) FROM envios WHERE problema = ",
$_REQUEST['id2'])->fetch_value( );

        $GLOBALS['peticiones'] = true;
    */
?>

```

Recomendación.php

```

<?php
require_once('includes/entorno.php');
main::inicia(__FILE__, './', 'html');

    entrada::filtra($_POST,
        array('cuantos', dato::ENTERO),
        array('dificultad', dato::ENTERO),
        array('tecnica1', dato::ENTERO, 0),
        array('tecnica2', dato::ENTERO, 0),
        array('tecnica3', dato::ENTERO, 0),
        array('tecnica4', dato::ENTERO, 0),
        array('tecnica5', dato::ENTERO, 0)
    );

```

```

entrada::examina('busqueda');
entrada::termina( );

function busqueda( )
{
    $GLOBALS['cuantos'] = $_REQUEST['cuantos'];
    $GLOBALS['dificultad'] = $_REQUEST['dificultad'];
    $GLOBALS['technical'] = $_REQUEST['technical'];
    $GLOBALS['tecnica2'] = $_REQUEST['tecnica2'];
    $GLOBALS['tecnica3'] = $_REQUEST['tecnica3'];
    $GLOBALS['tecnica4'] = $_REQUEST['tecnica4'];
    $GLOBALS['tecnica5'] = $_REQUEST['tecnica5'];
    $GLOBALS['problemas_db'] = db::query("SELECT id, dificultad FROM problemas")->fetch_all_assoc(
);
    $GLOBALS['tecnicas_db'] = db::query("SELECT problema, tecnica FROM tecnicas")->fetch_all_assoc(
);
    $GLOBALS['problemas_candidatos'] = array( );
    $GLOBALS['promedio'] = 0;

    function sugiere($i)
    {
        if (count($GLOBALS['problemas_candidatos']) == $GLOBALS['cuantos']) { // ya juntó
suficientes candidatos
            $suma = 0;
            foreach($GLOBALS['problemas_candidatos'] as $valor){
                $suma = $suma + $valor['dificultad'];//sacar el promedio de dificultad,
sumando de la var dificultad y dividiendo / numero_candidatos
            }
            $promedio = $suma / $GLOBALS['cuantos'];

            if (floor($promedio) < $GLOBALS['dificultad'] || ceil($promedio) >
$GLOBALS['dificultad'] +1 ){
                return;
                //si cumple esta restriccion ahora verificar las tecnicas
            }
            $tecnicas = array(
                1 => $GLOBALS['technical'],
                2 => $GLOBALS['tecnica2'],

```

```

        3 => $GLOBALS['tecnica3'],
        4 => $GLOBALS['tecnica4'],
        5 => $GLOBALS['tecnica5']
    );

    for ($i = 0; $i < count($GLOBALS['problemas_candidatos']); ++$i){
        $id = $GLOBALS['problemas_candidatos'][$i]['id'];
        for ($j = 0; $j < count($GLOBALS['tecnicas_db']); ++$j){
            if ($GLOBALS['tecnicas_db'][$j]["problema"] == $id) {
                $tec = $GLOBALS['tecnicas_db'][$j]['tecnica'];
                if ($tec != 6){
                    $tecnicas[$tec] -= 1;
                }
            }
        } // */ // terminar esto checa bien lo de las tecnicas una por una
        //
        //y de ahi ver si esta en el rango de 0 y numeros negativos

    }

    for($t = 1; $t < 6 ; ++$t){
        if($tecnicas[$t] > 0)
        {
            return;
        }
    }

    // checar si los candidatos cumplen las tecnicas regresa y evalua todas las tecnicas
    //ya teniendo la evaluacion y ver que se cumplieron sal del ciclo,si no funcion
recursiva
    //llama a un nuevo arreglo de problemas candidatos
    var_dump($GLOBALS['problemas_candidatos']);
    exit;
}
else if ($i < count($GLOBALS['problemas_db'])) { // nos faltan cadidatos, pero
todavía hay de donde agarrar
    array_push($GLOBALS['problemas_candidatos'], $GLOBALS['problemas_db'][$i]); // lo
consideramos
    sugiere($i + 1); // vemos qué pasa después

```

```

                                array_pop($GLOBALS['problemas_candidatos']);                                // y luego lo
sacamos                                                                    sugiere($i + 1);                                // lo ignoramos y nos vamos al
siguiente                                                                    }
                                                                    }
                                sugiere(0);
}
//+
                                /* lo mismo pero con if's anidados
                                if ($tec == 1) {
                                    $tecnicas[1] -= 1;
                                }
                                else if ($tec == 2) {
                                    $tecnicas[2] -= 1;
                                }
                                else if ($tec == 3) {
                                    $tecnicas[3] -= 1;
                                }
                                */
*/
?>

```

Db_final.sql

```

/*create database if not exists temp2;
USE temp2;
*/

create database if not exists final1;
USE final1;

DROP TABLE IF EXISTS `usuarios`;
CREATE TABLE `usuarios` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `clave_uam` varchar(45) DEFAULT NULL,
  `nombre` text NOT NULL,
  `apellido_paterno` text NOT NULL,

```

```

`apellido_materno` text NOT NULL,
`email` varchar(254) NOT NULL, /*aleatorio <--*/
`password` text NOT NULL,
`tipo` int(11) NOT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `email_UNIQUE` (`email`),
UNIQUE KEY `clave_uam_UNIQUE` (`clave_uam`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;
/*ya llenados, obtener los ids que existan*/

DROP TABLE IF EXISTS `alumnos_profesor`;
CREATE TABLE `alumnos_profesor` (
  `profesor` int(11) NOT NULL,
  `alumno` int(11) NOT NULL,
  KEY `profesor` (`profesor`),
  KEY `alumno` (`alumno`),
  CONSTRAINT `alumnos_profesor_ibfk_1` FOREIGN KEY (`profesor`) REFERENCES `usuarios` (`id`) ON DELETE
CASCADE,
  CONSTRAINT `alumnos_profesor_ibfk_2` FOREIGN KEY (`alumno`) REFERENCES `usuarios` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `concursos`;
CREATE TABLE `concursos` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` text,
  `organizador` int(11) NOT NULL,
  `password` text,
  `inicio` datetime NOT NULL,
  `duracion` time NOT NULL,
  `intentos` int(11) DEFAULT NULL,
  `estricto` tinyint(1) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `organizador` (`organizador`),
  CONSTRAINT `concursos_ibfk_1` FOREIGN KEY (`organizador`) REFERENCES `usuarios` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `inscritos`;
CREATE TABLE `inscritos` (
  `id` int(11) NOT NULL AUTO_INCREMENT,

```



```

`concurso` int(11) NOT NULL,
`usuario` int(11) NOT NULL,
PRIMARY KEY (`id`),
KEY `concurso` (`concurso`),
KEY `usuario` (`usuario`),
CONSTRAINT `inscritos_ibfk_1` FOREIGN KEY (`concurso`) REFERENCES `concursos` (`id`) ON DELETE CASCADE,
CONSTRAINT `inscritos_ibfk_2` FOREIGN KEY (`usuario`) REFERENCES `usuarios` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `envios` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `usuario` int NOT NULL, /* usuario int NOT NULL, */
  `problema` int NOT NULL, /* problema int NOT NULL, */
  `tiempo` datetime NOT NULL,
  `longitud_codigo` int(11) DEFAULT NULL,
  `estado` int(11) DEFAULT NULL, /* -1 es que no compiló, 10*/
  `bytes_memoria` int(11) DEFAULT NULL,
  `tiempo_ejecucion` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`usuario`) REFERENCES usuarios (id), /* FOREIGN KEY (`usuario`) REFERENCES usuarios (id)
*/
  FOREIGN KEY (`problema`) REFERENCES problemas (id) /*FOREIGN KEY (`problema`) REFERENCES problemas (id)*/
) ENGINE=InnoDB;

```

/* YA TENIENDO LA TABLA LLENA DE USUARIOS HACER EL QUERY DE CUANTOS ID TENGO, DESPUES HACER LO MISMO PARA LA TABLA DE PROBLEMAS y ya teniendo eso hacer que cuando llenemos envios sean usuarios y problemas validos*/

/* llenar la tabla usuarios (hacer un script que la llene posiblemente de basura) y llenar la tabla envios (posiblemente de basura pero en usuario debe ser un usuario valido y en problema debe ser un problema válido
estado es la calificación (-1 es que no compiló, 10 quiere decir que el envío resolvió el problema correctamente)
longitud_codigo debe ser positivo

programar queries que respondan:

- dado el problema I, ¿cuántos envíos son de ese problema?
- dado el problema I, ¿cuántos usuarios diferentes tienen estado = 10? UNIQUE
- dado el problema I, calcular el promedio de longitud_codigo AVERAGE AVG

```

Forma de insertar correcta en SQL--> (INSERT INTO usuarios (nombre, apellido_paterno, apellido_materno,
email, password,
        tipo) VALUES ('daniel','capistran', 'rodriguez', 'reks_dan_capi',2,3)
)
*/

```

```

DROP TABLE IF EXISTS `problemas`;
CREATE TABLE `problemas` (
  `id` int(11) NOT NULL AUTO INCREMENT,
  `titulo` varchar(255) NOT NULL,
  `autor` int(11) DEFAULT NULL,
  `archivos` int(11) NOT NULL,
  `descripcion` text NOT NULL,
  `limite_tiempo` int(11) NOT NULL,
  `limite_memoria` int(11) NOT NULL,
  `dificultad` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `titulo` (`titulo`),
  KEY `autor` (`autor`),
  CONSTRAINT `problemas_ibfk_1` FOREIGN KEY (`autor`) REFERENCES `usuarios` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*ya llenados, obtener los ids que existan*/

```

```

DROP TABLE IF EXISTS `tecnicas`;
CREATE TABLE `tecnicas` (
  `problema` int(11) NOT NULL,
  `tecnica` int(11) NOT NULL,
  KEY `problema` (`problema`),
  CONSTRAINT `tecnicas_ibfk_1` FOREIGN KEY (`problema`) REFERENCES `problemas` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

DROP TABLE IF EXISTS `archivos`;
CREATE TABLE `archivos` (
  `id` int(11) NOT NULL AUTO INCREMENT,
  `contenido` varchar(255) NOT NULL,
  `problema` int(11) NOT NULL,
  PRIMARY KEY (`id`),

```

```
FOREIGN KEY (`problema`) REFERENCES `problemas` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```