

2018 Parcial HTML, CSS, Programación Web

Disponible desde: Saturday, 5 de May de 2018, 09:30

Límite de entrega: Wednesday, 23 de May de 2018, 19:15

Ficheros requeridos: index.php, estilos.css, scripts.js, business.class.php, presentation.class.php, data_access.class.php, login.php, addUser.php, listUsers.php
(Descargar)

Número máximo de ficheros: 40

Tipo de trabajo: Individual

Las tareas a realizar en el examen son las del ENUNCIADO en PAPEL y NO estas que se suministran como Información de referencia.

Organización

Esta práctica se divide en dos partes: una primera de desarrollo de una aplicación web en el lado del servidor usando sólo PHP y una segunda que consiste en añadirle programación del lado del cliente en JavaScript.

Descripción general

Esta práctica consiste en crear una aplicación web. Se desea desarrollar una aplicación web, para dar soporte a una empresa denominada Artencuentro. Artencuentro se dedica a la intermediación entre profesionales de las artes gráficas y empresas que requieren sus servicios. Los profesionales de artes gráficas se especializan en la elaboración del diseño e imágenes empleadas en carteles, rótulos, logos, marcas, envases, etc. La aplicación de Artencuentro suministrará un lugar en internet donde los profesionales de las artes gráficas pueden presentar trabajos previos para mostrar sus habilidades. Las empresas con interés en contratar un profesional en este campo podrán contactar con ellos por medio de Artencuentro o presentar una oferta de trabajo para que sean los profesionales los que se pongan en contacto con ellas. Los trabajos expuestos podrán ser calificados con "Me gusta" por los visitantes.

Como la mayoría de las aplicaciones web se tienen distintos tipos de usuarios (roles): administrador, autor, empresa y visitante.

El administrador es el encargado de crear, ver, modificar y borrar usuarios.

Los autores pueden publicar obras, ver y dialogar sobre ofertas de empresas.

Las empresas se encargan de hacer ofertas de trabajos y dialogar con los autores.

Los visitantes pueden buscar y ver las obras. El resto de roles también pueden hacer todo lo que puede hacer un visitante.

Casos de uso según tipo de usuario

Visitante (Usuario no registrado)

Buscar obras por tipo y/o título aproximado. Como resultado se genera un listado mostrando autor, título, fecha y tipo.

Listar las obras de un autor. Desde el listado anterior se tiene enlaces desde el autor a listo de todas sus obras.

Mostrar una obra concreta con todos sus detalles, y enlazando al listado de obras del autor. Para llegar a esta información en los listados anteriores se tienen enlaces para mostrar los datos completos de cada obra.

Identificarse como usuario registrado.

Autores

Listado de obras propias.

Añadir/modificar/eliminar una obra desde el listado.

Elegir me gusta una obra.

Listar propuestas dirigidas al autor.

Ver/responder a mensajes de ofertas de realización de trabajos.

Empresas

Listar propuestas propias realizadas.

Ver/responder a mensajes de autor.

Proponer trabajo a un autor. [NO HACER]

Elegir me gusta una obra. [NO HACER]

Administrador

Listar usuarios. [NO HACER]

Crear, ver, modificar, borrar usuarios. [NO HACER]

Para el desarrollo de esta práctica se debe emplear PHP como lenguaje de programación en el servidor, SQLite3 como gestor de Bases de Datos, y para la programación del lado del cliente se empleará JavaScript y opcionalmente JQuery y JQuery-UI.

Como entorno de desarrollo se puede utilizar la plataforma aulaga u otro entorno de desarrollo que se prefiera. En cualquier caso la actividad debe entregarse y ser funcional en el

entorno suministrado por aulaga. Para usar la plataforma aulaga no es necesario realizar ninguna configuración. Se debe tener en cuenta que en caso de edición simultánea de la entrega en grupo se almacenará la última guardada.

Como ya se ha dicho. la base de datos a emplear, por su sencillez de configuración y uso, será SQLite accediendo a ella empleando como abstracción de acceso a bases de datos PHP Data Objects (PDO) .

Tablas empleadas

Para más detalle ver el código de creación de las tablas que se muestra más adelante. Tenga en cuenta que en aulaga las tablas se crean a su contenido inicial en cada ejecución. Además de crear las tablas se añaden datos iniciales, destacando los usuarios adm, aut1, aut2, aut3, emp1 y emp2 cuyas claves son respectivamente "1", "2", "3", "4", "5" y "6". Como se puede deducir por su nombre el primero es administrador, los tres siguientes son autores y los dos últimos empresas.

usuarios

Campo	Descripción	Tipo
id	Identificador numérico único del usuario	Entero auto
cuenta	Nombre de cuenta	Ristra
clave	Clave del usuario (cifrada con MD5)	Ristra
nombre	Nombre completo del usuario	Ristra
tipo	1 administrador, 2 autor, 3 empresa	Entero
poblacion	Población del local o usuario	Ristra
direccion	Calle, número, piso, etc.	Ristra
telefono	Teléfono de contacto	Ristra

obras

Campo	Descripción	Tipo
id	Identificador numérico único de la obra	Entero
idautor	Identificador numérico único del usuario autor	Entero
titulo	Título de la obra	Ristra
tipo	Tipo de la obra	Ristra
fecha	Fecha de realización	Ristra
descripcion	Descripción de la obra	Ristra

imagen Imagen en formato jpg Blob

propuestas

Campo	Descripción	Tipo
-------	-------------	------

id	Identificador numérico único de la propuesta	Entero
----	--	--------

idautor	Identificador numérico único del usuario autor	Entero
---------	--	--------

idempresa	Identificador numérico único del usuario empresa	Entero
-----------	--	--------

hora	Hora de propuesta en formato epoch	Entero
------	------------------------------------	--------

descripcion	Descripción de la propuesta	Ristra
-------------	-----------------------------	--------

presupuesto	Propuesta de presupuesto en euros	Entero
-------------	-----------------------------------	--------

acuerdo	0: no se ha producido acuerdo, 1: hay acuerdo	Entero
---------	---	--------

mensajes

Campo	Descripción	Tipo
-------	-------------	------

id	Identificador numérico único de mensajes	Entero
----	--	--------

idpropuesta	Identificador de la propuesta	Entero
-------------	-------------------------------	--------

idusuario	Identificador del usuario que envía el mensaje	Entero
-----------	--	--------

hora	Hora de envío en formato epoch	Entero
------	--------------------------------	--------

mensaje	Contenido del mensaje	Ristra
---------	-----------------------	--------

megusta

Campo	Descripción	Tipo
-------	-------------	------

idobra	Identificador de la obra	Entero
--------	--------------------------	--------

idusuario	Identificador del usuario que dice me gusta	Entero
-----------	---	--------

.

Restricciones a comprobar en al servidor para aceptar crear/actualizar los siguientes tipos de registros

obras:

titulo: mínimo 2 caracteres y máximo 32 caracteres

tipo: mínimo 5 caracteres y máximo 16 caracteres, solo minúsculas y espacios

fecha: formato correcto [d]d/[m]m/aaaa y fecha posible

descripción: mínimo 12 y máximo 1024 caracteres

propuestas:

descripcion: mínimo 16, máximo 64 caracteres

presupuesto: número entero no negativo

mensajes:

mensaje: mínimo 1 caracter y máximo 256 caracteres

Organización del código

El código compartido se distribuirá entre distintos ficheros php según su propósito:

presentation.class.php: contendrá el código encargado de genera código html comunes de las páginas.

data_access.class.php: contendrá el código de acceso a los datos almacenados en la base de datos.

business.class.php: contendrá el código de gestión, modificación y comprobación de la información (usando data_access.class.php).

Para depuración del manejo de la base de datos se dispone del fichero a.php que visualiza su contenido.

Formato de hora

Los diversos campos de las tablas anteriores que almacenan una hora o momento, almacenan este dato en forma de entero. Este entero representa el número de segundos desde el 1 de enero de 1970 (epoch). Para generar este entero se debe usar la función time(), para convertir el número de segundos (epoch) en una fecha y hora en forma textual se usa la función date().

Imágenes

Deberán estar en formato jpg, y se incrustan en la correspondiente tabla como blobs.

Parte 1

Como ya se ha indicado la práctica se divide en dos partes. En la primera sólo se debe emplear HTML+CSS+PHP y hacer peticiones desde enlaces o desde formularios para realizar los casos de uso. No se permite el uso de JavaScript.

Inicialización de la base de datos SQLite

```
DROP TABLE IF EXISTS [usuarios];
```

```
DROP TABLE IF EXISTS [obras];
```

DROP TABLE IF EXISTS [propuestas];

DROP TABLE IF EXISTS [mensajes];

DROP TABLE IF EXISTS [megusta];

/*

Create Tables

***/

CREATE TABLE [usuarios]

(
 [id] INTEGER PRIMARY KEY,
 [cuenta] NVARCHAR(20) NOT NULL,
 [clave] NVARCHAR(32) NOT NULL,
 [nombre] NVARCHAR(200) DEFAULT "",
 [tipo] INTEGER DEFAULT 2,
 [poblacion] NVARCHAR(200) DEFAULT "",
 [direccion] NVARCHAR(200) DEFAULT "",
 [telefono] NVARCHAR(200) DEFAULT ""

);

/*

tipo = 1 => administrador

tipo = 2 => autor

tipo = 3 => empresa

*/

CREATE UNIQUE INDEX IF NOT EXISTS [indexusuario] on [usuarios] ([cuenta]);

CREATE TABLE [obras]

(
 [id] INTEGER PRIMARY KEY,
 [idautor] INTEGER NOT NULL,
 [titulo] NVARCHAR(80),

```
[tipo] NVARCHAR(40),  
[fecha] NVARCHAR(80),  
[descripcion] TEXT,  
[imagen] BLOB,  
FOREIGN KEY ([idautor]) REFERENCES [usuarios] ([id])  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE [propuestas]  
(  
    [id] INTEGER PRIMARY KEY,  
    [idautor] INTEGER NOT NULL,  
    [idempresa] INTEGER NOT NULL,  
    [hora] INTEGER NOT NULL,  
    [descripcion] TEXT "",  
    [presupuesto] INTEGER NOT NULL,  
    [acuerdo] INTEGER Default 0,  
    FOREIGN KEY ([idautor]) REFERENCES [usuarios] ([id])  
        ON DELETE CASCADE,  
    FOREIGN KEY ([idempresa]) REFERENCES [usuarios] ([id])  
        ON DELETE CASCADE  
);
```

```
CREATE TABLE [mensajes]  
(  
    [id] INTEGER PRIMARY KEY,  
    [idpropuesta] INTEGER NOT NULL,  
    [idusuario] INTEGER NOT NULL,  
    [hora] INTEGER NOT NULL,  
    [mensaje] TEXT "",  
    FOREIGN KEY ([idpropuesta]) REFERENCES [propuestas] ([id])
```

```

        ON DELETE CASCADE,
    FOREIGN KEY ([idusuario]) REFERENCES [usuarios] ([id])
        ON DELETE CASCADE
);

```

```

CREATE TABLE [megusta]
(
    [id] INTEGER PRIMARY KEY,
    [idobra] INTEGER NOT NULL,
    [idusuario] INTEGER NOT NULL,
    FOREIGN KEY ([idobra]) REFERENCES [obras] ([id])
        ON DELETE CASCADE,
    FOREIGN KEY ([idusuario]) REFERENCES [usuarios] ([id])
        ON DELETE CASCADE
);

```

```

CREATE UNIQUE INDEX IF NOT EXISTS [indexmegusta] on [megusta] ([idobra],[idusuario]);

```

```

/*****
***

```

Populate Tables

```

*****/

```

```

INSERT INTO [usuarios] ([cuenta], [clave], [nombre], [tipo])
    VALUES ('adm', 'c4ca4238a0b923820dcc509a6f75849b', 'Elma Ndamás', 1);
INSERT INTO [usuarios] ([cuenta], [clave], [nombre], [tipo], [poblacion], [direccion], [telefono])
    VALUES ('aut1', 'c81e728d9d4c2f636f067f89cc14862c', 'Artis Taenun Saco', 2, 'Arucas',
'C/Mayor Nº 33', '928111111');
INSERT INTO [usuarios] ([cuenta], [clave], [nombre], [tipo], [poblacion], [direccion], [telefono])
    VALUES ('aut2', 'eccbc87e4b5ce2fe28308fd9f2a7baf3', 'Elme Jordi Bujante', 2, 'Telde',
'C/San Greorio Nº 44', '928123456');

```



```
INSERT INTO [usuarios] ([cuenta], [clave], [nombre], [tipo], [poblacion], [direccion], [telefono])  
  
VALUES ('aut3', 'a87ff679a2f3e71d9181a67b7542122c', 'Pint Orde Brocha', 2, 'Las Palmas  
de Gran Canaria', 'Avd/Juan XXV Nº 13', '928454545');
```

```
INSERT INTO [usuarios] ([cuenta], [clave], [nombre], [tipo], [poblacion], [direccion], [telefono])  
  
VALUES ('emp1', 'e4da3b7fbbce2345d7772b0674a318d5', 'Construcciones Eternas', 3,  
'Santa Brigida', 'C/Principal', '928252525');
```

```
INSERT INTO [usuarios] ([cuenta], [clave], [nombre], [tipo], [poblacion], [direccion], [telefono])  
  
VALUES ('emp2', '1679091c5a880faf6fb5e6087eb1b2dc', 'Mark Etingcan Arias', 3, 'Gáldar',  
'C/Larga', '928888888');
```

```
INSERT INTO [obras] ([idautor], [titulo], [tipo], [fecha], [imagen], [descripcion])  
  
VALUES (2, 'Programación en Piña', 'cartel', '1/1/2008', readfile('imagenes/cartel1.jpg'),  
  
'Cartel que representa la Programación rodeada de múltiples nueces y piñas que hay que  
abrir');
```

```
INSERT INTO [obras] ([idautor], [titulo], [tipo], [fecha], [imagen], [descripcion])  
  
VALUES (2, 'Portada libro PHP', 'portada libro', '13/1/2012',  
readfile('imagenes/cartel2.jpg'),  
  
'Portada de libro de prácticas de PHP en la asignatura Programación 4 del Grado en  
Ingeniería Informática');
```

```
INSERT INTO [obras] ([idautor], [titulo], [tipo], [fecha], [imagen], [descripcion])  
  
VALUES (3, 'Logo asignatura', 'logo', '14/1/2014', readfile('imagenes/logo2.jpg'),  
  
'Sencillo logo de la asignatura Programación con destino a situarlo en la portada en su  
curso Moodle');
```

```
INSERT INTO [obras] ([idautor], [titulo], [tipo], [fecha], [imagen], [descripcion])  
  
VALUES (3, 'Poster presentación charla PHP', 'poster', '1/4/2017',  
readfile('imagenes/cartel3.jpg'),  
  
'Poster de charla sobre PHP. Un experto en PHP nos visita para contar su experiencia de  
uso del lenguaje más usado en la Web');
```

```
INSERT INTO [obras] ([idautor], [titulo], [tipo], [fecha], [imagen], [descripcion])  
  
VALUES (4, 'Biblioteca', 'mural', '11/11/2028', readfile('imagenes/mural.jpg'),  
  
'Mural para serigrafiar y pegar en puertas de cristal en Biblioteca');
```

```
INSERT INTO [obras] ([idautor], [titulo], [tipo], [fecha], [imagen], [descripcion])  
  
VALUES (4, 'Logo VPL', 'logo', '1/1/2008', readfile('imagenes/logo1.jpg'),  
  
'Logo de VPL Virtual Programming Lab para Moodle');
```

```
INSERT INTO [propuestas] ([idautor], [idempresa], [hora], [descripcion], [presupuesto])  
VALUES (2, 5, 1520326648, "Realización de Logo de empresa Transportes El Águila", 1500);  
INSERT INTO [propuestas] ([idautor], [idempresa], [hora], [descripcion], [presupuesto])  
VALUES (4, 6, 1520337448, "Escuela de Ingeniería Informática: desarrollar letras y logo de  
4 metros de largo a situar en el exterior de edificio", 2500);
```

```
INSERT INTO [mensajes] ([idpropuesta], [idusuario], [hora], [mensaje])  
VALUES (1, 2, 1520344648, "Necesito más detalles");  
INSERT INTO [mensajes] ([idpropuesta], [idusuario], [hora], [mensaje])  
VALUES (1, 5, 1520345648, "Vale, te añado los detalles");  
INSERT INTO [mensajes] ([idpropuesta], [idusuario], [hora], [mensaje])  
VALUES (1, 2, 1520346648, "Sí, me puede interesa");
```

```
INSERT INTO [megusta] ([idobra], [idusuario]) VALUES (1, 2);  
INSERT INTO [megusta] ([idobra], [idusuario]) VALUES (2, 2);  
INSERT INTO [megusta] ([idobra], [idusuario]) VALUES (3, 4);  
INSERT INTO [megusta] ([idobra], [idusuario]) VALUES (1, 3);  
INSERT INTO [megusta] ([idobra], [idusuario]) VALUES (3, 2);  
INSERT INTO [megusta] ([idobra], [idusuario]) VALUES (4, 2);
```