

Aplicación de laboratorio

Tal y como se ha comentado en clase, esta aplicación será el tronco fundamental del desarrollo del módulo. En ella aplicaremos los conocimientos adquiridos en HTML, CSS, PHP y MySQL. La realización de la práctica por parte del alumno es **obligatoria y necesaria** para conseguir aprobar la asignatura.

Dividiremos el desarrollo en cuatro fases:

1ª fase: Creación de la parte “visual” en HTML

2ª fase: Sustitución total (salvo en el caso de algunas tablas muy específicas) de los elementos de estilo HTML por sus homólogos CSS. Una vez terminada esta fase **NO se podrá incluir** en el documento otro tipo de estilo que el definido en un fichero CSS.

3ª fase: Inclusión de PHP, tanto para la generación dinámica de HTML como la posibilidad de conexión a una base de datos MySQL.

4ª fase: Creación-modificación de las estructuras de datos en la base de datos.

Estas fases se seguirán siempre que se añada una nueva opción en la aplicación.

La utilización de “referencias a fichero” absolutas (del tipo /home/alumno/php/fichero.inc) están totalmente prohibidas.

Será motivo de suspenso la utilización de nombres o adjetivos soeces en los ejemplos de la aplicación.

El estilo de la edición del código debe ser legible y claro. La utilización de comentarios es obligatoria en: Separación de bloques, explicación de los algoritmos, el nombre de variables y constantes. En el resto, aunque no obligatoria, sí muy recomendable.

Todas las aplicaciones deben contener, al menos, las siguientes subcarpetas:

/css: se guardarán los archivos de estilo.

/includes: los archivos con extensión .inc.

/script: los archivos php que no generan código html.

/img: las imágenes de la aplicación.

Dentro de la subcarpeta includes ya hemos creado (11-12-2007) los siguientes ficheros:

configuracion.inc

Se guardarán las constantes que se utilicen en la aplicación. El nombre de la constante debe comenzar por C, continuar por un sustantivo en minúsculas. En el caso de utilizar más de una palabra, la segunda y siguientes comenzarán por mayúscula y continuarán con minúsculas. Ejemplo: CnombreUsuario.

Al lado de cada constante existirá un comentario aclaratorio del porqué de dicha constante. Utilizaremos // para los comentarios de una sola línea y /* */ para los restantes.

cabecera.inc

Código php que generará el código html contenido entre <html> y <body>, así como la caja de principio del documento.

pie.inc

Código php que generará la caja del final del documento, así como el elemento </html>

menu.inc

Contendrá el código php que genere el menú de la aplicación. Dicho menú constará de las siguientes opciones:

- Peticiones
- Carga de trabajo
- Resultados
- Configuración
- Acerca de
- Salir

Los diferentes scripts contendrán parte de esta relación de includes, así como una parte, variable en cada uno de ellos, denominada ***contenido*** que caracterizará a la opción. Esta parte será generada (en una fase más avanzada del curso) por una función que a partir de parámetros cargará un contenido u otro.

Los scripts desarrollados hasta el momento (11-12-2007)

login.php

Constará de:

- configuracion.inc
- cabecera.inc
- Un contenido, que será un formulario con un campo de texto con nombre *usuario* y un campo password con nombre *contrasenia*
- Los valores se pasarán por POST al menu.php
- pie.inc

menu.php

- configuracion.inc
- cabecera.inc
- menu.inc
- Un contenido, de momento el nombre del usuario y si la contraseña y usuario están autorizados para la ejecución de la aplicación.
- pie.inc

acercade.php

- configuracion.inc
- cabecera.inc
- menu.inc
- un contenido en el que aparecerán los autores de la aplicación, así como otros datos que se crean de interés.
- pie.inc

peticiones1.php

- configuracion.inc
- cabecera.inc
- menu.inc
- un contenido con un formulario con el campo de texto con nombre *nhc* que se pasará por POST a *peticiones2.php*.
- pie.inc

peticiones2.php

- configuracion.inc
- cabecera.inc
- un contenido con dos formularios:
 - en el primero aparecerán los datos demográficos del paciente.
 - en el segundo se solicitará un número de muestra, la fecha, el tipo de muestra y hasta cuatro análisis que se seleccionarán de la tabla correspondiente.
 - un botón grabar que, en un futuro, guardará los datos en la base de datos y volverá a *peticiones1.php*.

- un botón Salir que no grabará nada y volverá al menú principal.
- un botón modificar paciente.

cargatrabajo.php

- configuracion.inc
- cabecera.inc
- menu.inc
- un contenido con una tabla que mostrará las hojas de trabajo, las muestras que contiene dicha hoja, la fecha de actualización y un checkbox.
- un botón salir y otro actualizar.

Las constantes básicas de la aplicación serán:

<i>CtituloAplicacion</i>	El title de la aplicación
CestiloFichero	apuntará al css elegido, si se incluyeran más css las constantes que las referencien pasarían a llamarseCestiloFichero1, CestiloFichero2, etc.
Copcion1	Peticiones
Copcion2	Carga de trabajo
Copcion3	Resultados
Copcion4	Configuración
Copcion5	Acerca de
Copcion6	Salir
CtextoLogin	El texto que aparece en el formulario login.php, campo usuario.
CtextoPassword	El texto que aparece en el formulario login.php campo contrasenia.
CtextoNhc	El texto que aparece en el campo peticiones1.php campo nhc
Ctextomuestra	El texto que aparece en el campo peticiones2.php campo muestra.
Ctextofechamuestra	El texto que aparece en el campo peticiones2.php campo fechamuestra.
Ctextotipomuestra	El texto que aparece en el campo peticiones2.php campo tipomuestra.
Ctexto analisis1	El texto que aparece en el campo peticiones2.php campo analisis1
Ctexto analisis2	El texto que aparece en el campo peticiones2.php campo analisis2
Ctexto analisis3	El texto que aparece en el campo D peticiones2.php campo analisis3
CbaseDatos	Nombre de la base de datos

CrootBaseDatos	Nombre del usuario de la base de datos
CpassBaseDatos	Password del usuario de la base de datos

A fecha 11-12-2007, y hasta que conectemos la aplicación a la base de datos, existirán las constantes:

CnombreUsuario	El usuario autorizado.
CclaveUsuario	La clave del usuario autorizado.

Todos los textos que vayan apareciendo en la aplicación se incluirán en configuracion.inc como constantes. Al añadir un nuevo texto, se decidirá el nombre de la constante.

Ejercicios a realizar en el periodo de vacaciones de Navidad 2007-08

Ejercicio 1: Todas las aplicaciones deben tener la estructura expuesta anteriormente.

Ejercicio 2: El usuario decidirá cómo hacer la pantalla de configuración. En ella debe accederse a la gestión de análisis, gestión de hojas de trabajo y gestión de usuarios.

Ejercicio 3: script de creación de la base de datos MySQL que contendrá las siguientes tablas:

Tusuario:

id_usuario: integer, autoincrement, Not Null y Primary Key.
 nombre_usuario: varchar(20) Not Null.
 pass_usuario: varchar(30)
 grupo_usuario: varchar(2)

Tgrupos_usuario:

grupo_usuario: varchar(2) Primary Key.
 descrip_grupo_usuario: varchar(20)

Tpacientes:

id_paciente: integer, autoincrement, Not Null y Primary Key.
 nombre_paciente: varchar(20)
 apellido1: varchar(20)
 apellido2: varchar(20)
 tfno: varchar(15)
 domicilio: varchar(30)
 codpost: varchar(10)
 cod_provincia: varchar(2)

Tmuestras:

id_muestra: integer, autoincrement, Not Null y Primary Key.

id_paciente: integer
num_mue: varchar(10)
fecha_mue: date
id_usuario:integer

Tanal_mues:

id_muestra:integer
cod_anal: integer //estos dos campos serán primary key.
resultado: varchar (50)

Tanalisis:

cod_anal:integer, autoincrement, Not Null y Primary Key.
des_anal: varchar(20)

El motor de base de datos debe ser MyISAM.

Para acceder a MySQL desde MAX:

1º.- abre un terminal.

2º.- teclea *mysql -u root -p*

3º.- Solicitará un password que, en este caso, es cmadrid.

Si la conexión ha sido correcta aparecerá el siguiente prompt: ***mysql>***

Para crear la base de datos:

`CREATE DATABASE <nombrebasedatos>;`

Para borrar la base de datos:

`DROP DATABASE <nombrebasedatos>;`

Para trabajar con la base de datos:

`USE <nombrebasedatos>;`

A partir de aquí ya puedes crear las tablas indicadas.

Un ejemplo de creación de tabla MyISAM:

```
create table usuarios (  
    id_usuario integer NOT NULL PRIMARY KEY  
    auto_increment,  
    des_usuario varchar(25),  
    clave varchar(15),  
    activo char (1)  
    ) Type= MyISAM;
```

Para salir de mysql: **exit;**

También puedes instalar, en tu casa, el software phpmyadmin.

Ejercicio 4: Crea las siguientes tablas, y añádelas al script:

THojasdetrabajo:

id_ht: integer, autoincrement, Not Null y Primary Key.
nombre_ht: varchar(20) Not Null.

TanalHT:

id_ht: integer, Not Null
cod_anal: integer, Not Null
ámbas formarán una única primary key.

Ejercicio 5: Crea el script **compruebalogin.php**. En este script se realizará, de momento, un include de analisis.inc (ubicado en el directorio de includes) que contendrá un array denominado analisis con dos dimensiones: la primera será escalar e indicará el ind_usuario, y la segunda, asociativa, con los siguientes campos: usuario y clave.

Este script recogerá los datos de login.php, comprobará si el usuario y el password se encuentra en el array y, de ser así, pasará por sesión el id_usuario a menu.php.

En caso de no ser un usuario autorizado pasará el control a **nologin.php** que, en contenido, tendrá el mensaje de usuario no autoizado y un href hacia login.php.

Realiza los cambios necesarios en el resto de script para que puedan acceder al id_usuario.