

PAC de Desarrollo UF3

CFGS DAW

Módulo 07: Desarrollo web en entorno servidor

2S 2022/2023



1. Normativa

Requisitos necesarios que debe cumplir el trabajo:

- La PAC de Desarrollo se debe entregar únicamente a través de CodeGrade (encontraréis el enlace en la tarea), y dentro de los plazos (día y hora máximo) establecidos en la guía didáctica. En caso de no cumplir dichos plazos, no se podrán enviar de forma posterior. No se evaluarán entregas por mensajes o comentarios.
- La entrega se hará en un único fichero .zip que contendrá en el directorio raíz (sin subcarpetas) los siete ficheros que se describen abajo: articulos.php, conexion.php, consultas.php, formarticulos.php, funciones.php, index.php y usuarios.php
- Para realizar esta actividad se debe consultar los contenidos del material didáctico y hacer uso de fuentes bibliográficas y recursos web.
- Cualquier código detectado como plagio supondrá un suspenso (0) para todas las partes implicadas, independientemente de la fecha de entrega. La plataforma CodeGrade cuenta con un detector de plagio preciso y que arroja un informe vinculante que se tendrá en cuenta a la hora de anular dos o más trabajos. En ningún caso se corregirán aquellos trabajos involucrados en una detección de plagio.
- La entrega del archivo en la plataforma es responsabilidad del alumno/a, debe asegurarse de haber subido en la plataforma el fichero correcto, ya que en ningún caso el profesor revisará el contenido antes del periodo de corrección.
- El programa debe realizarse con PHP y HTML exclusivamente. También contendrá las sentencias SQL necesarias para llevar a cabo la gestión de la base de datos.
- La calificación oscilará entre 0 y 10. Si no se entrega una PAC de desarrollo, la calificación equivaldrá a un 0.
- El/La alumno/a es conocedor de esta normativa en el momento de la entrega de la PAC de desarrollo.

2. Enunciado de la actividad

Esta actividad consiste en la realización de una aplicación web con las siguientes características:

2.1 Descripción

Se quiere desarrollar una aplicación web que permita a sus usuarios actualizar la información de los productos que tiene actualmente en una tienda. Entre las funcionalidades principales encontramos:

- Un usuario superadmin podrá permitir o no si se añaden, editan o borran los productos, además de ver a todos los usuarios de la aplicación.
- Una serie de usuarios autorizados que podrán acceder al listado de productos y, en caso de estar permitido por el superadmin, añadir productos, editarlos o borrarlos.

2.2 Tecnologías

Solo se podrá utiliza PHP y HTML para el desarrollo, así como una base de datos MySQL que se proporciona. Para desarrollarlo se utilizará el Modelo – Vista – Controlador, por lo que todo el código deberá estar separado en base a los objetivos que tiene.

Se adjuntan todos los ficheros necesarios para la realización de esta actividad y las funciones que se han de implementar y utilizar. No se podrán utilizar otros ficheros y deberá hacerse uso de las funciones propuestas para implementar el funcionamiento.

2.3 Restricciones

Para la correcta evaluación de tus ejercicios, tendrás que tener en cuenta algunas restricciones en la programación:

1. **No puedes hacer uso de los estilos, no se puede utilizar código CSS**
2. **Todos los botones de envío de formulario deben ser `INPUT type="submit"`**
3. **Cuidado con los acentos y las ñ. Recuerda que se pueden mostrar en texto, pero no se deben utilizar ni en bases de datos ni en enlaces. Solamente se deben utilizar a la hora de mostrar un texto.**
4. **Has de cumplir al detalle con los requisitos mostrados. Si se pide que se muestre el nombre del usuario, se debe mostrar el nombre del usuario. Si se pide que se comente que el usuario no está registrado o no tiene permisos, debe mostrarse el mensaje correspondiente. Si se pide que aparezca un enlace, ese enlace debe estar en la página.**
5. **Los campos de los formularios que se describen en cada fichero deben ser los que se mencionan, ni añadir ni eliminar.**
6. **El listado de productos u usuarios debe mostrarse a modo de tablas, tal y como se indica y con las columnas que se indica.**
7. **Revisad muy bien las rutas. Deben ser rutas relativas para encontrar los recursos que se necesitan. En especial, cuidado con los IDEs con liveserver, ya que tienden a generar rutas que no existen.**

2.4 Objetivos

Objetivo principal: Desarrollar un sistema que permita la gestión de los productos de una tienda y los permisos para poder hacerlo.

Objetivos específicos:

- Desarrollar el acceso para los diferentes tipos de usuarios al sistema.

- Desarrollar un sistema para que el superadmin pueda acceder al listado de usuarios y cambiar los permisos sobre la manipulación de productos.
- Desarrollar el sistema de gestión de los productos (añadir, editar y borrar) y su consulta.

2.5 Base de datos

Se proporciona una base de datos que contiene toda la información para el funcionamiento de la aplicación. Las características de las tablas y las columnas que son necesarias para el desarrollo de esta aplicación son:

- **category:** contiene las categorías de productos que tiene disponibles en la tienda.
 - **id:** Identificador de cada categoría.
 - **name:** Nombre de la categoría.
- **product:** contiene todos los productos almacenados en la tienda.
 - **id:** Identificador de cada producto.
 - **name:** Nombre del producto.
 - **cost:** Coste del producto.
 - **price:** Precio final del producto.
 - **category_id:** Identificador de la categoría a la que pertenece el producto.
- **setup:** contiene los datos relacionados con la administración del sitio.
 - **management:** Indica los permisos actuales que tienen los usuarios registrados y autorizados.
 - 1: Pueden gestionar productos.
 - 0: No pueden gestionar productos.
 - **superadmin_id:** Indica el identificador del usuario que tiene permisos de superadmin.
- **user:** contiene los datos de los usuarios registrados.
 - **id:** Identificador de cada usuario.
 - **email:** Correo electrónico de cada usuario.

- full_name: Nombre del usuario.
- enabled: Indica si el usuario registrado tiene o no autorización para acceder a la gestión de los productos.
 - 1: usuario autorizado.
 - 0: usuario no autorizado.

2.6 Estructura de ficheros

Para la realización de esta aplicación se proporcionan los ficheros sobre los cuales se deberá implementar:

- index.php: contiene el sistema de acceso a la aplicación mediante el nombre de usuario y su dirección de correo electrónico, es decir, un formulario de tres campos de tipo input (nombre, email y submit). En este fichero se deberá comprobar qué tipo de usuario es y permitir el acceso a la aplicación:
 - En caso de ser superadmin, mostrará un **mensaje de bienvenida que incluya su nombre** (importante) y **mostrará un enlace para acceder a usuarios.php**
 - En caso de ser un usuario autorizado, mostrará un **mensaje de bienvenida que incluya su nombre** (importante) y un **enlace para acceder a articulos.php**
 - En caso de ser un usuario registrado, pero no autorizado, mostrará un **mensaje de bienvenida que incluya su nombre** (importante) y un mensaje advirtiéndole de que **no tiene permisos** de acceso.
 - En caso de que sea un usuario no registrado o se introduzcan unos datos incorrectos, mostrará un mensaje advirtiéndole de que **no está registrado**.
 - Almacenará en una cookie el tipo de usuario que ha intentado registrarse.
- usuarios.php: contiene el sistema de control sobre los permisos de los usuarios autorizados y muestra el listado de todos los usuarios registrados. Solo el superadmin tiene permiso para entrar aquí. Tendrá las siguientes características:
 - Comprobará si el acceso a esta página se ha hecho por un usuario que tiene los permisos suficientes, comprobando la cookie creada en index.php. En

caso de no tener acceso mostrará un mensaje advirtiéndole de que **no tiene permisos** de acceso.

- Indicará el valor almacenado en la base de datos con los permisos actuales de la aplicación (campo management).
- Tendrá un botón que, al pulsar sobre él, cambiará el valor de los permisos de la aplicación (campo management).
- Mostrará una tabla con los datos de todos los usuarios registrados en la aplicación con las siguientes columnas: Nombre (full_name), Email (email) y Autorizado (enabled).
- En caso de usuarios autorizados, el valor de ese campo de la tabla se representará en negrita.
- Tendrá un enlace que permitirá volver a index.php.
- **articulos.php:** contiene el sistema de control sobre los productos de la tienda. Cualquier usuario registrado y autorizado tiene acceso a esta página. Tendrá las siguientes características:
 - Comprobará si el acceso a esta página se ha hecho por un usuario que tiene los permisos suficientes, comprobando la cookie creada en index.php. En caso de no tener acceso mostrará un mensaje advirtiéndole de que **no tiene permisos** de acceso.
 - Mostrará una tabla con los datos de todos los productos almacenados con las siguientes columnas: ID, Nombre, Coste, Precio, Categoría y Acciones.
 - Al pulsar sobre el título de cada columna (excepto Acciones), permitirá ordenar de menor a mayor el contenido de la tabla basándose en el parámetro que se ha pulsado.
 - **En el caso de que estén los permisos de la aplicación (management) activados** (campo management), aparecerán también las siguientes opciones, que no aparecerán en caso de que los permisos de la aplicación (management) estén desactivados:

- Antes del listado de productos, un enlace para añadir un producto que lleva a formarticulo.php con la opción de añadir producto. Debe llamarse a la página mediante el método GET.
 - Ya en la lista de productos, un enlace junto a cada producto (en la columna Acciones) que permita editarlo y lleve a formarticulo.php. Debe llamarse a la página mediante el método GET.
 - También en la lista de productos, un enlace junto a cada producto (en la columna Acciones) que permita borrarlo y lleve a formarticulo.php. Debe llamarse a la página mediante el método GET.
- Tendrá un enlace que permitirá volver a index.php.
- formarticulo.php: contiene el formulario que permitirá realizar gestiones sobre los productos. Cualquier usuario registrado y autorizado tiene acceso a esta página. El formulario debe tener estrictamente los campos que corresponden a los atributos de los productos (con los inputs id, nombre, coste, precio, y un select para la categoría) y un input type submit para añadir el producto. El input id siempre debe estar deshabilitado, ya que es información que no se modificará.
 - Comprobará si el acceso a esta página se ha hecho por un usuario que tiene los permisos suficientes, comprobando la cookie creada en index.php. En caso de no tener acceso mostrará un mensaje advirtiéndole de que **no tiene permisos** de acceso.
 - En el caso de venir al formulario para Añadir, mostrará un formulario vacío. Al completar todos los datos (menos el ID) y pulsar sobre el botón, se añadirá el producto a la base de datos y mostrará un mensaje de que el producto se ha añadido.
 - En el caso de venir al formulario para Editar, mostrará el formulario completo con los datos del producto que se va a editar (con el ID incluido pero inhabilitado) y el botón de Editar. Al realizar las modificaciones y pulsar sobre el botón, se modificará el producto en la base de datos y mostrará un mensaje de que el producto se ha modificado.

- En caso de venir al formulario para Borrar, mostrará el formulario completo con los datos del producto que se va a borrar y el botón de Borrar. Al pulsar sobre el botón, se eliminará el producto de la base de datos y mostrará un mensaje de que le producto se ha borrado.
- Tendrá un enlace que permitirá volver a `articulos.php`.
- `conexion.php`: contiene la declaración de las funciones para crear una conexión con la base de datos y para cerrar la conexión.
- `consultas.php`: contiene la declaración de las funciones para realizar todas las consultas a la base de datos necesarias de la aplicación.
- `funciones.php`: contiene la declaración de las funciones para realizar acciones auxiliares que se necesitan para implementar la funcionalidad en cada una de las páginas de la aplicación.

2.7 Funciones a desarrollar

Para la realización de esta aplicación es necesario implementar una serie de funciones que están almacenadas en `conexion.php`, `consultas.php` y `funciones.php`. En caso de ser necesario se podrán desarrollar funciones auxiliares, respetando el MVC propuesto.

Conexión.php

crearConexion()

- Devuelve un manejador de conexión con la base de datos.

cerrarConexion(\$conexion)

- Cierra la conexión indicada por el manejador `$conexion`.

Consultas.php

tipoUsuario(\$nombre, \$correo)

- Recibe un \$nombre de usuario (full_name) y un \$correo electrónico de usuario (email).
- Comprobará si el usuario está registrado, y en caso de estarlo, qué permisos tiene.
- Devuelve una cadena de caracteres indicando el tipo de usuario que corresponda:
 - Si corresponde con el superadmin => "superadmin"
 - Si corresponde con un usuario registrado y autorizado => "autorizado"
 - Si corresponde con un usuario registrado, pero no autorizado => "registrado"
 - Si no corresponde con ninguno que esté en la base de datos => "no registrado"

esSuperadmin(\$nombre, \$correo)

- Recibe un \$nombre de usuario (full_name) y un \$correo electrónico de usuario (email).
- Comprobará si el usuario está identificado como superadmin. Devuelve un valor booleano: True, si es superadmin, False si no lo es.

getListaUsuarios()

- Devuelve una tabla virtual con los datos (full_name, email, enabled) de todos los usuarios almacenados en la tabla user.

getPermisos()

- Devuelve el valor almacenado en la columna management de la tabla setup.

cambiarPermisos()

- Cambia el valor almacenado en la columna management de la tabla setup. Si vale 0 cambia a 1 y si vale 1 cambia a 0.

getCategorias()

- Devuelve una tabla virtual con los datos (id, name) de todas las categorías almacenadas en la tabla category.

getProduct(\$ID)

- Recibe un \$ID que corresponde al identificador de un producto.
- Devuelve una tabla virtual con todos los datos que corresponden al producto cuyo identificador es \$ID.

getProductos(\$orden)

- Recibe el \$orden por el que deben ordenarse los productos.
- Devuelve una tabla virtual con el contenido de todos los productos de la base de datos, ordenados por el valor de \$orden. Esta tabla debe contener los siguientes valores por cada producto:
 - De la tabla product: id, name, cost, price
 - De la tabla category: name

anadirProducto(\$nombre, \$coste, \$precio, \$categoria)

- Recibe el \$nombre, \$coste, \$precio e identificador de \$categoria de un producto.
- Añade ese producto a la base de datos.
- Devuelve el resultado de la consulta realizada.

borrarProducto(\$id)

- Recibe el \$identificador de un producto.
- Elimina ese producto de la base de datos.
- Devuelve el resultado de la consulta realizada.

editarProducto(\$id, \$nombre, \$coste, \$precio, \$categoria)

- Recibe el \$identificación, \$nombre, \$coste, \$precio e identificador de \$categoria de un producto.

- Actualiza la información de ese producto en la base de datos.
- Devuelve el resultado de la consulta realizada.

funciones.php

pintaCategorias(\$defecto)

- Representa en HTML las opciones para poder seleccionar las categorías de los productos según la tabla category para un input de tipo select.
- El contenido de cada opción deberá ser el nombre (name) de la categoría y el valor asociado a la opción el identificador de la categoría (id).
- Deberá mostrar seleccionada la opción indicada por \$defecto.

pintaTablaUsuarios()

- Representa en HTML el contenido de la tabla de user indicando como encabezados Nombre (full_name), Email (email) y Autorizado (enabled).
- En aquellos casos en que el usuario esté autorizado (su valor de enabled sea igual a 1), el valor se representará en negrita (etiquetas y)

pintaProductos(\$orden)

- Representa en HTML la tabla con todos los productos almacenados en la tabla product, indicando como encabezados ID (id), Nombre (name), Coste (cost), Precio (price), Categoría (el valor name asociado al category_id según la tabla category) y Acciones.
- Se mostrará el resultado ordenado según el \$orden que se pasa como parámetro.
- En cada producto, en la columna de Acciones se deberán incluir dos enlaces que permitan Editar y Borrar ese elemento. Esos enlaces utilizarán para el envío de información el método GET. **Cada uno de ellos debe mostrar solamente ese texto en el enlace (Editar y Borrar respectivamente) de forma que se sepa qué hace cada enlace.**

3. Evaluación

La evaluación de esta actividad se realizará siguiendo la siguiente rúbrica, teniendo en cuenta que será necesario superar los requisitos de un nivel inferior para optar a la calificación del superior:

1. Acceso como usuario no registrado (0,5)
2. Acceso como usuario con credenciales incorrectas (1)
3. Acceso como usuario con credenciales correctas, pero no está autorizado (1)
4. Acceso como usuario superadmin al panel de usuarios y cambio de permisos (1)
5. Acceso como usuario autorizado al panel de productos con los permisos de la aplicación desactivados y ordenando los productos por el nombre del producto (1,5)
6. Acceso como usuario autorizado al panel de productos con los permisos de la aplicación activados y añadir un nuevo producto (2)
7. Acceso como usuario autorizado al panel de productos con los permisos de la aplicación activados y editar un producto existente (1,5)
8. Acceso como usuario autorizado al panel de productos con los permisos de la aplicación activados y borrar un producto existente (1,5)

Criterios	Pruebas	Puntos
Acceso como usuario no registrado	Usuario y correo que no aparecen en base de datos	0.2
	Intentar entrar en las páginas sin hacer login	0.3
Acceso con credenciales incorrectas	Acceso con nombre correcto pero email incorrecto	0.3
	Acceso con nombre incorrecto pero email correcto	0.3
	Acceso con nombre y email correctos pero de usuarios distintos	0.4
Acceso como usuario registrado pero no autorizado	Login correcto pero aviso de que no tiene permisos	0.4
	Intentar entrar en las páginas tras hacer login pero sin permisos	0.6
Acceso como superadmin y cambio de permisos de la aplicación (management)	Acceso con usuario superadmin.	0.2
	Acceso correcto a la página de usuarios y cambio de permisos para habilitar o deshabilitar el manejo de productos	0.8
Acceso como usuario autorizado al panel de productos con los permisos desactivados y ordenando los productos	Acceso con usuario registrado y autorizado	0.2
	Acceso correcto a la página de productos con los permisos inhabilitados	0.6
	Acceso correcto a la página de productos ordenados	0.7
Acceso como usuario autorizado los productos y añadir uno nuevo	Acceso al listado y al formulario de artículos, correctamente implementado	0.5
	Añadimos un nuevo producto	1.5
Acceso como usuario autorizado los productos y editar uno	Acceso al listado y al formulario de artículos y editamos uno	1.5
Acceso como usuario autorizado los productos y borrar uno	Acceso al listado y al formulario de artículos y borramos uno	1.5

