

Práctica 12 - Gestión de almacén

1 Requisitos

Nuestro cliente, una fábrica de muebles, desea informatizar algunos procesos de su negocio para lo que nos solicita que diseñemos una aplicación WEB. La información que pretende gestionar gira en torno a los muebles y las piezas que los componen. Los datos a tener en cuenta para los muebles son su código de mueble (único), su nombre y su precio. Todo mueble está formado por una o más piezas. Cada pieza tiene un identificador único, un nombre y una descripción. Cada pieza puede formar parte de varios muebles. Interesa saber cuántas unidades de cada pieza componen cada mueble. **Por ejemplo, la mesa de televisión está formada por 3 piezas: tablero en 1 unidad, patas en 4 unidades y ruedas en 4 unidades.** Todas las unidades de una pieza se encuentran en uno o más estantes del almacén. No se permite mezclar distintas piezas en cada estante. El estante viene determinado de forma única por dos valores: pasillo y altura. Además de en qué estantes están las piezas, interesa conocer cuantas unidades de la pieza hay almacenadas en cada estante. **Por ejemplo, en el pasillo uno altura uno hay almacenadas 50 unidades de la pieza "Bisagra metal". De dicha pieza hay un total de 173 unidades almacenadas en cuatro estantes de acuerdo con la siguiente distribución: 50, 50, 50 y 23.** La aplicación WEB debe permitir:

- Solicitar un listado con todos los muebles. Para cada mueble se ha de mostrar su nombre y precio. Este listado debe estar disponible para todos los usuarios.
- La consulta de una pieza mostrando su nombre y unidades disponibles de esa pieza en el almacén. Este listado debe estar disponible sólo para usuarios identificados.

2 Base de datos

El equipo de bases de datos ha diseñado la base de datos de acuerdo al siguiente esquema relacional:

```
Mueble (#cod, nombre, precio);
Pieza (#cod, nombre, descrip);
Formado (#cod_mueble, #cod_pieza, unidades);
Estante (#pasillo, #altura, cod_pieza, unidades);
Formado.cod_mueble ⊆ Mueble.cod;
Formado.cod_pieza ⊆ Pieza.cod;
Estante.cod_pieza ⊆ Pieza.cod;
```

A partir de dicho esquema se ha creado la base de datos. Dicha base de datos se puede importar desde un SGBD a partir del archivo "Muebles.sql" disponible en:

<https://app.box.com/s/5u5rr9mb1gcncszpi47jboh4df7sazyw>.

3 Presentación

El equipo de diseño WEB ha diseñado las páginas estáticas de la aplicación. El esquema general de las mismas contiene un menú situado a la izquierda. Cada menú de cada página incluye opciones que deben ser restringidas o no de acuerdo con los requisitos que se van a detallar. Las páginas que entrega el equipo de diseño WEB son:

- ☒ index.php -> Página de inicio accesible para todos los usuarios.
- ☐ listado.php -> Listado de muebles disponible para todos los usuarios.
- ☒ login.php -> Página de acceso sólo accesible para usuarios anónimos. Esta página envía por método POST las variables user y pass.
- ☒ user_page.php -> Página de bienvenida a usuarios identificados. No accesible desde menú.
- ☐ form_existencias.php -> Página con desplegable de selección de pieza sólo accesible a usuarios identificados.
- ☐ existencias.php -> Página no accesible desde menú con información de existencias de la pieza seleccionada en form_existencias. Dicha pieza se recoge por método POST de la variable "pieza".
- ☒ La opción del menú "Cerrar sesión" sólo estará disponible para usuarios identificados.

Las páginas estáticas se suministran en el archivo "www Muebles.zip" disponible en la URL:

<https://app.box.com/s/k2fpf8ru1zyjtfv1pstxs0gqeohqwo1>.

4 Se pide

Las tareas a realizar son:

- Establecer plan de pruebas de validación.
- Desplegar la base de datos en la máquina Ubuntu que tiene LAMP y phpMyAdmin ya instalados.
- Desarrollar la aplicación WEB con consultas a la base de datos a partir de las páginas estáticas.
- Desplegar la aplicación.
- Pasar el plan de pruebas.
- Una memoria de todas las tareas realizadas.