**Tarea 2 – UD3**

**CARRITO DE COMPRA con AUTENTICACIÓN**

**Descripción de la Práctica:**

La práctica consiste en la implementación de un sistema de autenticación de usuario mediante un formulario propio en PHP, con contraseñas almacenadas de manera segura utilizando la función password\_hash() y verificación mediante password\_verify().

Con esta autenticación, se busca restringir el acceso a la aplicación solo a usuarios autenticados, y algunas partes específicas como la gestión de stock, solo a usuarios específicos.

Se utiliza session\_start() para manejar las variables de sesión y rastrear la autenticación del usuario.

El código se estructura en archivos separados para realizar tareas específicas, como mostrar formularios, cerrar sesiones y verificar la autenticación.

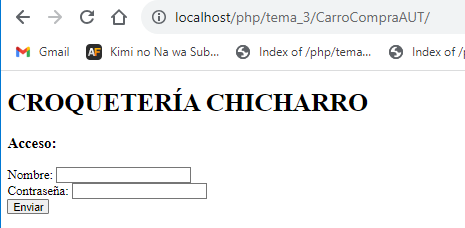
Y luego una segunda opción, teniendo guardada la información de usuarios y resúmenes (hash) de contraseñas en una tabla de base de datos.

***\*\*Indicar que las explicaciones de este documento están un poco escuetas, sobre todo las cosas técnicas, porque está todo explicado/comentado en el código de todos los archivos php, todos los pasos y líneas del código están explicados con comentarios.***

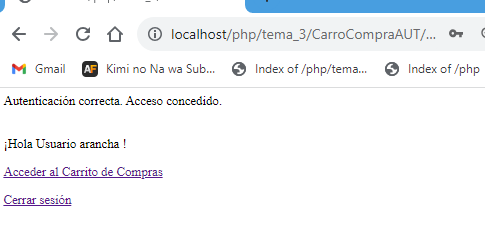
**Pasos Realizados y Estructura del código:**

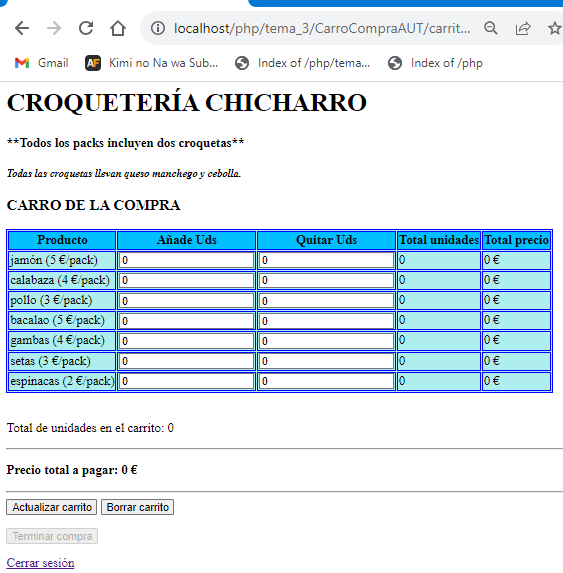
1. **Creación usuarios y contraseñas:**
   * **Stock.php**
   * Creo array usuarios para almacenar los usuarios y sus contraseñas, a modo de clave-valor.
   * Como tenemos que crear contraseñas seguras, usamos la función *password\_hash()*, y ese código hash es el que almacenamos en el array. *(las líneas de generación de hash las comentamos para que se puedan visualizar pero no deberían de estar ahí como protección).*
2. **Implementación de la Autenticación:**
   * **index.php**
     1. Se creó un formulario sencillo de autenticación en index.php, para que sea lo que aparezca en el navegador al querer ir a cualquiera de las páginas de mi aplicación.
     2. Include de stock.php.
     3. Comprueba si existe un indicador de redirección (redir) en la solicitud. Si existe, muestra un mensaje de error de *Nombre o la contraseña son incorrectos.*.
     4. Aquí creamos una función *muestraFormulario* al que, al llamarla, paso una url para que se redirija la página, en este caso le pasamos la url *auth.php.*
   * **Auth.php**
     1. Se inicia la sesión y carga el archivo stock.php con include.
     2. Creo variables para obtener el nombre de usuario y la contraseña de la solicitud.
     3. El código de auth.php verifica las credenciales. Para comprobar el nombre se usa *array\_key\_exists* para verificar si el nombreUsuario está en la clave del array y para la contraseña, se usa *password\_verify*, para verificar que la contraseña introducida coincide con la almacenada en el array usuarios. Y se redirige según el tipo de usuario.
     4. Si es admin, puede acceder a todas las páginas de la aplicación, por lo que se le muestra enlaces al carrito de compras, a la gestión de stock y a la info del stock. Además de un mensaje de bienvenida y otro enlace para cerrar sesión.
     5. Si no es admin, solo puede acceder a la página de carrito de compras, por lo que se muestra enlace a esta página además de un mensaje de bienvenida y enlace de cierre de sesión
     6. Si la autenticación es correcta, en la variable de sesión “authok” almacenamos el valor 1, indicando así que el usuario está autenticado..
     7. Si la autenticación falla, se redirige a la página inicial (index.php) con un header y con Locatión, y parámetro redir , haciendo que ésta muestre el mensaje de error correspondiente al tener el parámetro redir.
3. **Restricción de Acceso:**
   * **check-auth.php**
   * Se crea check-auth.php para verificar si el usuario ha iniciado sesión y si no ha iniciado sesión, se redirige a la página de inicio.
   * Si la variable de sesión authok no está establecida o su valor no es igual a 1, significa que el usuario no está autenticado.
   * Este código asegura que el usuario solo tenga acceso a las partes protegidas de la aplicación si ha iniciado sesión correctamente. Si la variable de sesión 'authok' no está establecida o su valor no es igual a 1, el usuario es redirigido a la página de inicio con un indicador de redirección. En resumen, garantiza la autenticación del usuario antes de permitir el acceso a ciertas áreas de la aplicación.
4. **Cierre de Sesión:**
   * **logout.php**
   * Creamos un logout.php para cerrar la sesión con *sesión\_destroy()* y redirigir al usuario a la página de inicio con un *header(Location).*
   * Los enlaces a esta página, los hemos implementado en las páginas carritoCompras, gestiónStock, infoStock y en auth, para que el usuario siempre tenga la opción de cerrar sesión.

**Capturas de pruebas:**

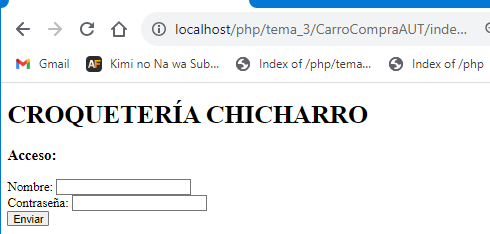
-Página nada más acceder a la carpeta de la aplicación: index.php:  


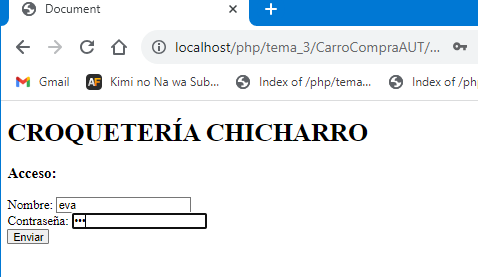
-Probamos con el usuario “arancha” (contraseña la misma):  

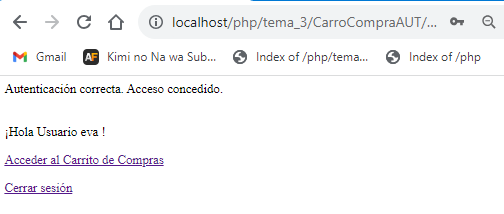

- y damos a enviar y nos redirige a auth.php. Vemos que se muestra mensaje de autenticación correcta (que se imprime en el if de la verificación) y enlace para acceder al carrito de compra, ya que se ha detectado que no es admin y por lo tanto solo puede acceder ahí, también muestra enlace de cerrar sesión:  
*Al iniciar sesión, aparece advertencia del navegador avisando de contraseña poco segura, pues en nuestro caso la hemos puesto igual que el nombre.*  


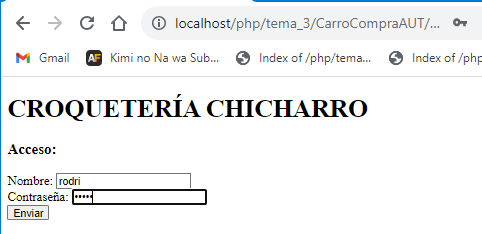
- accedemos al carrito de compras, y efectivamente nos redirige al carritoCompras.php, donde también se muestra enlace de cerrar sesión:  


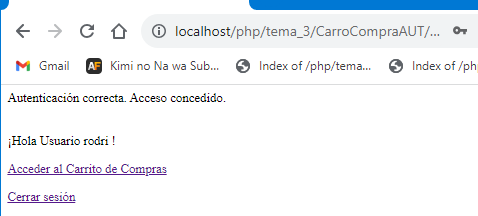
- Probamos a cerrar sesión, y nos redirige nuevamente al formulario, ya que este enlace dirige a logout.php y este manda un Location con header a la página del formulario, index.php:

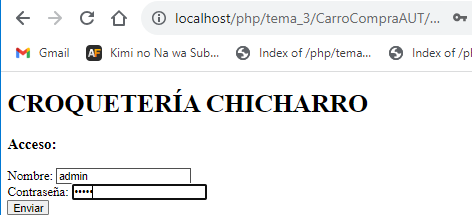


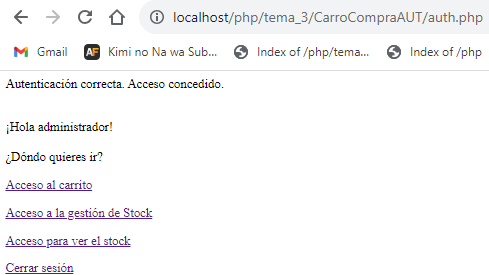
- Ahora probamos con el usuario “eva”, misma contraseña, y también funciona:  




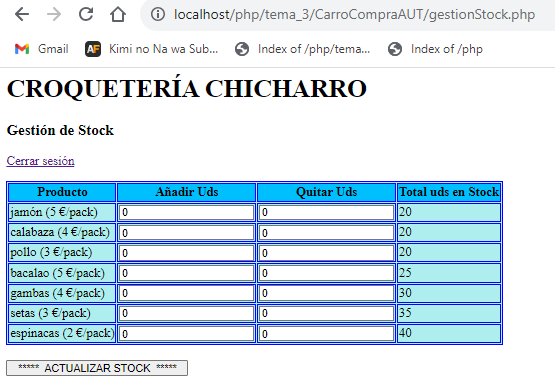
-Y con el usuario rodri:  




-Y por último probamos con el usuario “admin”, que tiene privilegios para acceder a todas las páginas de la aplicación:  


-Vemos que se muestran los enlaces de acceso a todas las páginas, además de enlace para cerrar sesión.  


-Probamos con acceso a la gestión de stock, y nos redirige bien, mostrando además enlace de cerrar sesión:



**PARTE OPCIONAL: usuarios y contraseñas hash en tabla de BBDD**

Para no borrar código de la aplicación, he copiado los archivos y para modificar código sin borrar el de la aplicación inicial. De esta manera ahora hay dos aplicaciones, una con autenticación php y otra con autenticación con BBDD.

**Pasos Realizados y Estructura del código:**

1. **Modificación de stock.php:**
   * Se añadió código para conectar a una base de datos MySQL utilizando la extensión PDO.
   * Se realizaron consultas a la tabla de usuarios en la base de datos para obtener la información de los usuarios y sus contraseñas hash.
   * Se eliminó el código de generación manual de hashes en favor de la obtención de datos desde la base de datos.
2. **Modificación de croqueteria.sql:**
   * Se añadió la creación de una base de datos llamada 'croqueteria'.
   * Se creó una tabla llamada 'usuarios' para almacenar los nombres de usuario y las contraseñas hash.

**Desafíos y Soluciones:**

**Comentarios y conclusiones:**

La implementación de un sistema de autenticación segura es crucial para proteger áreas sensibles de una aplicación. La utilización de funciones de hashing como password\_hash() proporciona una capa adicional de seguridad. La persistencia de sesiones y el manejo adecuado de la información de autenticación son esenciales para una experiencia de usuario fluida y segura. La práctica también resalta la importancia de la gestión segura de contraseñas y la necesidad de implementar medidas de seguridad, como HTTPS, para proteger la transmisión de datos.