

## **UNIDAD 3: Tarea**

### **Desarrollo de aplicaciones web con PHP**

¿Qué te pedimos que hagas?

#### **ACTIVIDADES**

1. Realiza los siguientes ejercicios:

a) Crea una página de ejemplo con [autenticación HTTP básica](#). A continuación, en dicha página, responde: ¿por qué no es seguro esta autenticación? ¿Cómo podría mejorarse? Para responder, puedes ayudarte con [este material](#).

b) Crea una página de ejemplo con [autenticación HTTP hash/digest](#) con el algoritmo Blowfish/Bcrypt usando [crypt\(\)](#). A continuación, en dicha página, responde: ¿qué es una [función hash](#)? ¿Por qué es tan importante en la seguridad informática? ¿Cuál sería la autenticación HTTP más segura de todas finalmente? Para responder, puedes ayudarte con este [material](#).

2. Crea una cesta de la compra de frutas y verduras y almacénala durante una hora usando [cookies](#). El usuario elegirá una serie de frutas y verduras y mantendrá dichos alimentos escogidos durante una hora aunque cierre el navegador. Mediante [setcookie\(\)](#), sin autenticación ni sesiones, almacena dicha cesta y crea también opciones para modificar, eliminar y comprobar si existe dicha cookie. Puedes ayudarte de los materiales de [Diego Lázaro](#) y de [W3Schools](#).

3. ¿Qué es [Composer](#) y qué [no es](#) según ellos? ¿Con qué [archivo](#) configuramos nuestro proyecto en Composer? ¿Qué es [JSON](#)? Completa el [tutorial de JSON](#). ¿Cómo [instalamos](#) nuestras dependencias en Composer y qué archivo se genera? ¿Cómo [actualizamos](#) nuestras dependencias a sus últimas versiones? ¿Qué es [Packagist](#) y [cómo se usa](#)? ¿Cómo cargamos (específicamente [autoloading](#)) nuestras dependencias en PHP?

4. Crea un balance sencillo de ingresos y gastos. Para obtener el balance, habrá que registrarse como usuario en una base de datos y después iniciar sesión; las contraseñas estarán almacenadas en hash mediante el algoritmo [Bcrypt](#) (usa [password\\_hash](#) y [password\\_verify](#)). Una vez iniciada la sesión correctamente, el usuario introducirá, mediante formularios, una serie de ingresos y gastos, pulsará en Generar y, a continuación, verá un informe similar a [éste](#) en PDF. Usa la librería [MPDF](#) usando Composer para instalarla y cargarla.

5. Realiza la [aplicación web](#) (sistema de test online, reservas online de coches o pizzería online) que no realizaste en la tarea anterior pero esta vez usando (además de bases de datos y sesiones) cookies para guardar la solicitud o pedido y funciones hash para almacenar las contraseñas de los usuarios. Si te ves capaz, genera el informe además en PDF usando MPDF mediante Composer.

## MATERIAL COMPLEMENTARIO

El material complementario se puede encontrar en la Unidad 3 de DWES de la web de Manuel Ignacio López Quintero.

## FORMATO DE ENTREGA

1. La estructura de archivos y carpetas debe quedar [así](#).
2. La página *index.html* permitirá acceder a todas las actividades.
3. El *script* [inicio.sql](#) creará las dos bases de datos con sus tablas, inserciones y restricciones necesarias.
4. El profesor descomprimirá el ZIP en *htdocs*, iniciará XAMPP, entrará en PhpMyAdmin, ejecutará/importará el *script inicio.sql* y abrirá *index.html* desde *localhost*.
5. El **no cumplimiento** del formato de entrega conllevará una calificación de cero en la presente tarea.

## ACTIVIDAD EXTRA PROPUESTA POR EL PROFESORADO (10 %)

- Subir esta tarea a Heroku con la siguiente URL:  
apellidos\_tarea3\_dwes\_2021.herokuapp.com.
- Esta URL se indicará en el archivo de texto *heroku.txt*.
- Si se sube esta tarea a Heroku, el profesor evaluará todo los ítems desde dicha URL excepto el código que lo evaluará en el *zip* entregado.