

## Práctica 1 UD 5: Desarrollo de una aplicación segura

**Agrupamiento:** Grupal

**Duración:** 8 sesiones

**Objetivos:** Aplicar todos los conocimientos obtenidos durante el curso para la creación de una aplicación web totalmente funcional y segura

### Introducción

En esta última práctica, se pondrán a prueba todos los conocimientos adquiridos a lo largo del curso mediante el desarrollo de una tienda online. Este proyecto se llevará a cabo de manera grupal, lo que permitirá trabajar en equipo y aplicar las habilidades de colaboración de cada alumno.

El objetivo de esta práctica es crear una tienda online funcional utilizando tecnologías web como **PHP**, **HTML**, **JavaScript** y **CSS**. A lo largo del curso, se han mostrado diferentes aspectos de seguridad en aplicaciones web, la autenticación de usuarios, la protección contra vulnerabilidades comunes, entre otros temas. Todos estos conocimientos se deberán de poner en práctica integrándose en el desarrollo de este proyecto.

Durante el desarrollo del proyecto, se emplearán metodologías ágiles como **SCRUM** o **Kanban**, las cuales permiten una gestión flexible y eficiente del desarrollo del software. Estas metodologías se centran en la entrega iterativa y en la colaboración continua entre los miembros del equipo, lo que facilita la adaptación a los cambios y la mejora constante del producto.

Además, se utilizará un sistema de control de versiones como **GIT**, que permitirá gestionar de manera efectiva el código fuente y colaborar de forma simultánea en el desarrollo del proyecto. **GIT** proporciona herramientas para el seguimiento de cambios, la gestión de ramas y la integración de código, lo que facilita el trabajo en equipo y garantiza la integridad del código.

La combinación de metodologías ágiles y un sistema de control de versiones proporcionará un marco de trabajo sólido y eficiente para el desarrollo de la tienda online. Los equipos tendrán la flexibilidad necesaria para adaptarse a los requisitos cambiantes del proyecto y garantizar la entrega oportuna de funcionalidades.

La tienda online deberá tener funcionalidades básicas como la visualización de productos, la selección de productos para comprar, la gestión del carrito de compras, la autenticación de usuarios, entre otras características. Además, se espera que implementen medidas de seguridad adecuadas para proteger la información sensible de los usuarios y evitar posibles vulnerabilidades.

Este proyecto brindará la oportunidad de aplicar de manera práctica todos los conceptos aprendidos, así como también les permitirá desarrollar habilidades de trabajo en equipo, resolución de problemas y toma de decisiones en un entorno realista.

## **Metodología**

Para llevar a cabo el desarrollo del proyecto, se emplearán las metodologías Kanban y SCRUM, que se adaptan de manera eficaz a proyectos de desarrollo de software al permitir una gestión visual y flexible del trabajo. La metodología se dividirá en dos etapas principales: **Desarrollo y Control de Calidad (QA)**.

Cada alumno se encargará de una parte específica de la aplicación que ha sido asignada por sorteo. Cada parte de la aplicación tendrá requisitos obligatorios que deben cumplirse para garantizar la coherencia y el funcionamiento adecuado del proyecto en su conjunto. Estos requisitos obligatorios pueden incluir aspectos como la funcionalidad básica, la seguridad, la usabilidad y la integración con otras partes del sistema.

Una vez cumplidos los requisitos obligatorios, los alumnos tienen total libertad para la implementación y personalización de sus respectivas partes de la aplicación. Esto significa que pueden utilizar su creatividad y habilidades para diseñar y desarrollar características adicionales, mejorar la interfaz de usuario, optimizar el rendimiento o añadir cualquier otro elemento que consideren relevante para mejorar la calidad y la experiencia del usuario.

## **Repositorio GIT**

El repositorio Git utilizado en el proyecto proporciona un espacio centralizado para el almacenamiento y gestión del código fuente del proyecto. En este repositorio, los alumnos pueden colaborar, realizar cambios, mantener un historial de versiones y coordinar el trabajo de manera efectiva.

- Repositorio: <https://github.com/adbteacher/eshop-pps>

## **Metodología SCRUM**

Se aplicará la metodología Scrum para gestionar el desarrollo del proyecto. Se llevarán a cabo **reuniones semanales** donde se revisará el estado del proyecto y se analizará la evolución de cada parte de la aplicación. Estas reuniones, conocidas como "Sprint Reviews", permitirán al equipo evaluar el progreso realizado durante la semana anterior, identificar posibles problemas o impedimentos y planificar las tareas a realizar en la próxima semana.

Durante estas reuniones, cada miembro del equipo presentará el trabajo realizado en su parte de la aplicación, destacando los avances, los problemas encontrados y las posibles mejoras propuestas. Se revisarán los requisitos obligatorios cumplidos y se discutirán las decisiones de diseño y las funcionalidades implementadas.

## **Metodología Kanban**

La metodología Kanban para el proyecto establece un proceso visual y transparente para la gestión del flujo de trabajo. En Kanban, se utilizan tableros divididos en columnas que representan diferentes etapas del proceso, desde la planificación hasta la implementación y la revisión. Cada tarea o elemento de trabajo se representa mediante tarjetas que se mueven a través de las columnas según avanza su progreso.

- Departamento de desarrollo:
  - Trello Board 1:  
<https://trello.com/invite/b/FPaEGzzH/ATTI8d50fe83826f2c86284a6cc14bc0df0cC1B01A0E/departamento-de-desarrollo>
  - Trello Board 2:  
<https://trello.com/invite/b/2aRbKGrY/ATTI2151b83ba6746cc73764fc3eccc7156124082CF/departamento-de-desarrollo>
    - En estos tableros, los equipos de desarrollo podrán visualizar las tareas pendientes, en progreso y completadas, así como asignarlas y moverlas según el progreso del trabajo.
- Departamento de control de calidad (QA):
  - Trello Board 1:  
<https://trello.com/invite/b/y5VdrMqa/ATTIe54d344fcfd2c292257bf4beb01a3df1C2A347C6/departamento-de-qa>
  - Trello Board 2:  
<https://trello.com/invite/b/yjuNNkX7/ATTI1e19eb3d35d83311e0cad5708d238617F6BC23B3/departamento-de-qa>
    - En estos tableros, el equipo de QA podrá gestionar las actividades relacionadas con la revisión y prueba de las funcionalidades desarrolladas, así como reportar y hacer seguimiento de los problemas encontrados.

## Código

Es fundamental que el código esté completamente comentado y bien organizado para facilitar la comprensión por parte de otros miembros del equipo. Esto permite que cualquier miembro del equipo pueda entender rápidamente cómo funciona cada parte del sistema y realizar contribuciones de manera eficiente

```
1 <?php
2 /**
3  * Este script PHP se encarga de procesar el formulario de registro de usuarios.
4  * Recibe los datos del formulario, los valida y los guarda en la base de datos.
5  */
6
7 // Incluir el archivo de configuración de la base de datos
8 require_once 'config.php';
9
10 // Definir variables y establecer valores predeterminados
11 $name = $email = $password = '';
12 $errors = [];
13
14 // Verificar si se ha enviado el formulario
15 if ($_SERVER["REQUEST_METHOD"] == "POST") {
16     // Validar y sanitizar los datos del formulario
17     $name = htmlspecialchars(trim($_POST['name']));
18     $email = htmlspecialchars(trim($_POST['email']));
19     $password = htmlspecialchars(trim($_POST['password']));
20
21     // Verificar si los campos están vacíos
22     if (empty($name)) {
23         $errors[] = "El nombre es obligatorio.";
24     }
25     if (empty($email)) {
26         $errors[] = "El correo electrónico es obligatorio.";
27     }
28     if (empty($password)) {
29         $errors[] = "La contraseña es obligatoria.";
30     }
31
32     // Si no hay errores, procesar el formulario
33     if (empty($errors)) {
34         // Insertar los datos en la base de datos
35         $sql = "INSERT INTO users (name, email, password) VALUES (?, ?, ?)";
36         $stmt = $pdo->prepare($sql);
37         $stmt->execute([$name, $email, $password]);
38         // Redirigir al usuario a la página de inicio de sesión
39         header("Location: login.php");
40         exit();
41     }
42 }
43 ?>
```

## **Proyecto (10 puntos)**

A continuación, se detallan cada una de las partes de la aplicación, junto con los requisitos obligatorios que deben cumplirse:

### **Parte PRE**

#### **1. Login**

- **Autenticación segura:** Utilizar un algoritmo de hash robusto para almacenar las contraseñas de los usuarios en la base de datos, como bcrypt o Argon2.
- **2FA (Autenticación de dos factores):** Disponer de un segundo método de autenticación opcional, como un código generado por una aplicación de autenticación o enviado por mensaje de texto, además de la contraseña estándar.
- **Registro de actividad de inicio de sesión:** Registrar y monitorizar los intentos de inicio de sesión, incluyendo la dirección IP y otros detalles relevantes, para detectar y responder a posibles intentos de acceso no autorizados.

#### **2. Recuperar contraseña**

- **Proceso seguro de restablecimiento de contraseña:** El proceso de restablecimiento de contraseña debe ser seguro y totalmente funcional a través de correo para los usuarios normales y los vendedores.
- **Verificación adicional en caso de intentos fallidos:** Si un usuario intenta restablecer la contraseña varias veces sin éxito, se debe requerir una verificación adicional de identidad antes de permitir más intentos.
- **Registro de actividades de restablecimiento:** Registrar y monitorizar las solicitudes de restablecimiento de contraseña, incluyendo detalles como la dirección IP del solicitante y la hora de la solicitud, para detectar y responder a posibles intentos de acceso no autorizados.

#### **3. Registro**

- **Formulario de registro seguro:** Se deberá implementar un formulario de registro seguro para los dos tipos de cuenta que existen: usuarios normales y vendedores. En el caso del usuario normal se deberá de solicitar nombre, apellidos, número de teléfono, dirección física, dirección de correo electrónico y contraseña. En el caso del usuario vendedor se deberá de solicitar, nombre de la empresa, CIF, dirección comercial, número de teléfono de contacto, dirección web de su tienda, dirección de correo electrónico y contraseña.
- **Verificación de identidad para los vendedores:** Implementar un proceso adicional de verificación de identidad para los vendedores, que puede incluir la presentación de documentos comerciales o información fiscal (ambos documentos en formato PDF).

- **Verificación de correo electrónico:** Enviar un correo electrónico de verificación a la dirección proporcionada por el usuario para confirmar la autenticidad de la cuenta y activarla.
- **Autenticación segura:** Utilizar un algoritmo de hash robusto para almacenar las contraseñas de los usuarios en la base de datos, como bcrypt o Argon2.
- **Política de contraseñas:** Establecer una política de contraseñas robusta que requiera una combinación de letras, números y caracteres especiales, y que exija una longitud mínima para garantizar la seguridad de las cuentas de usuario.

## **Parte POST**

### **1. Mi perfil**

- **Edición de información personal:** Permitir al usuario editar su información personal, como nombre, dirección de envío, dirección de correo electrónico y número de teléfono.
- **Cambio de contraseña:** Proporcionar una opción para que el usuario cambie su contraseña de inicio de sesión.
- **Gestión de direcciones:** Permitir al usuario gestionar múltiples direcciones de envío, editar las existentes y agregar nuevas direcciones según sea necesario.
- **Métodos de pago:** Permitir al usuario agregar, editar o eliminar métodos de pago asociados a su cuenta, como tarjetas de crédito, PayPal u otros métodos de pago aceptados.

### **2. Rol vendedor**

- **Gestión de productos:** Permitir al vendedor ver y administrar los productos que ha listado en la tienda, incluyendo la capacidad de agregar nuevos productos, editar información existente y eliminar productos obsoletos.
- **Gestión de inventario:** Proporcionar herramientas para que el vendedor pueda realizar un seguimiento del inventario de sus productos, incluyendo la cantidad disponible, los niveles de stock y la capacidad de realizar ajustes cuando sea necesario
- **Estadísticas de ventas:** Mostrar al vendedor estadísticas detalladas sobre el rendimiento de sus productos, incluyendo datos como el número de ventas, ingresos generados, productos más vendidos y tendencias de ventas a lo largo del tiempo.
- **Gestión de clientes:** Permitir al vendedor ver una lista de clientes que han comprado sus productos. Proporcionar información básica sobre los clientes, como el nombre y la dirección de correo electrónico.

### 3. Rol usuario normal

- **Estructura troncal:** Definir e implementar la estructura troncal de la aplicación.
- **Búsqueda y navegación:** Permitir a los usuarios normales buscar productos por categoría, precio o palabras clave, y navegar por las distintas secciones de la tienda de manera intuitiva.
- **Formulario de tickets de soporte:** El usuario tendrá a disposición en la aplicación web un formulario de incidencias donde podrá exponer sus dudas o problemas técnicos. Este formulario estará disponible para cualquier usuario autenticado en la aplicación (rol usuario normal o vendedor).

### 4. Rol usuario soporte técnico

- **Gestión de tickets de soporte:** Permitir al usuario de soporte técnico recibir, gestionar y responder a tickets de soporte generados por los clientes. Habilitar la clasificación y priorización de los tickets según la gravedad del problema y el tiempo de respuesta requerido.
- **Comunicación con clientes:** Facilitar la comunicación bidireccional entre el equipo de soporte técnico y los clientes, permitiendo responder a consultas, proporcionar actualizaciones sobre el estado de los problemas y ofrecer soluciones o asistencia técnica.
- **Registro de actividades:** Registrar todas las interacciones y acciones realizadas por el equipo de soporte técnico en relación con los tickets de soporte, incluyendo notas, cambios de estado y resolución de problemas.

### 5. Rol administrador

- **Gestión de usuarios:** Permitir al administrador crear, modificar y eliminar cuentas de usuario dentro del sistema (cualquier tipo de cuenta). Habilitar la asignación de roles y permisos específicos a cada usuario según sus responsabilidades y funciones dentro de la plataforma.
- **Gestión de productos:** Facilitar la creación, edición y eliminación de productos en el catálogo de la tienda, incluyendo la definición de atributos, precios, existencias y categorías. Permitir la importación y exportación masiva de datos de productos mediante archivos CSV.
- **Análisis y reporting:** Ofrecer herramientas de análisis y generación de informes para el seguimiento del rendimiento de la tienda, incluyendo métricas de ventas, inventario, tráfico web y comportamiento del usuario.

### 6. Proceso de compra

- **Carrito de compras:** Proporcionar un carrito de compras intuitivo y fácil de usar que permita a los usuarios agregar, modificar y eliminar productos, así como ajustar las cantidades y ver el resumen de la compra en todo momento.
- **Métodos de pago:** Ofrecer opciones de pago seguras y variadas, incluyendo métodos de pago en línea como tarjetas de crédito, transferencias bancarias y plataformas de pago electrónicas. (Obviamente no tienen que funcionar, en



el último paso del proceso de compra, cuando se pulse pagar, se comprara el artículo).

- **Confirmación de la compra:** Proporcionar a los usuarios un resumen detallado de su compra, incluyendo los artículos seleccionados, el total de la compra, los costos de envío y cualquier impuesto aplicable. Requerir que los usuarios confirmen su compra antes de proceder al pago, mostrando claramente los términos y condiciones de la compra.
- **Finalización de la compra:** Permitir a los usuarios revisar y confirmar los detalles de envío y facturación antes de completar la transacción, ofreciendo la opción de guardar esta información para futuras compras. Enviar una confirmación de compra por correo electrónico al usuario después de que se haya completado con éxito la transacción, incluyendo un resumen de la compra y un número de seguimiento del pedido si corresponde.
- **Cupones de descuento:** Integrar un campo de entrada para que los usuarios ingresen códigos de cupón de descuento durante el proceso de pago. Validar automáticamente los códigos de cupón ingresados para garantizar su autenticidad y aplicar el descuento correspondiente al total de la compra si el cupón es válido. Mostrar mensajes claros de confirmación o error para informar a los usuarios sobre el resultado de aplicar el cupón de descuento.

## **7. Artículos**

- **Catálogo de productos:** Mostrar una lista completa de todos los productos disponibles en la tienda, organizados por categorías y subcategorías para facilitar la navegación. Permitir a los usuarios realizar búsquedas rápidas y avanzadas para encontrar productos específicos según criterios como nombre, marca, precio o características.
- **Detalles del producto:** Proporcionar información detallada sobre cada producto, incluyendo imágenes de alta calidad, descripción, precio, disponibilidad, opciones de tamaño/color y especificaciones técnicas. Permitir a los usuarios ver reseñas y calificaciones de otros clientes, así como agregar sus propias opiniones sobre los productos.
- **Ofertas y promociones:** Resaltar productos en oferta o promociones especiales con etiquetas visibles y llamativas, junto con detalles sobre el descuento aplicable y el período de validez. Permitir a los usuarios filtrar productos por ofertas y promociones para encontrar fácilmente las mejores ofertas disponibles en la tienda en línea.
- **Funcionalidades de interacción:** Permitir a los usuarios agregar productos a su carrito de compras directamente desde la página de detalles del producto, con opciones para seleccionar la cantidad y las variantes disponibles.

## 8. Mis pedidos

- **Visualización de pedidos:** Permitir a los usuarios acceder a una lista completa de sus pedidos anteriores, mostrando detalles como el número de pedido, fecha de compra, estado del pedido y detalles del artículo. Proporcionar opciones de filtrado y búsqueda para que los usuarios puedan encontrar fácilmente pedidos específicos según criterios como fecha, estado o número de pedido.
- **Detalles del pedido:** Permitir a los usuarios ver información detallada sobre cada pedido, incluyendo los artículos comprados, la cantidad, el precio unitario, el subtotal, los costos de envío y cualquier impuesto aplicable. Mostrar el estado actual del pedido, como "En Proceso", "Enviado" o "Entregado", junto con cualquier información de seguimiento disponible para los envíos.
- **Seguimiento de pedidos:** Permitir a los usuarios realizar un seguimiento del estado de sus pedidos en tiempo real, desde la confirmación de la compra hasta la entrega final, proporcionando actualizaciones y notificaciones oportunas sobre cualquier cambio en el estado del pedido.
- **Gestión de devoluciones y reembolsos:** Facilitar el proceso de devolución y reembolso de productos, ofreciendo políticas claras y transparentes, así como un sistema automatizado para gestionar las solicitudes de devolución y procesar los reembolsos de manera oportuna.
- **Historial de transacciones:** Mantener un historial completo de todas las transacciones realizadas por el usuario, incluyendo compras exitosas, cancelaciones de pedidos, cambios en los pedidos y solicitudes de devolución. Permitir a los usuarios acceder y revisar su historial de transacciones en cualquier momento para realizar un seguimiento de sus actividades de compra y mantener un registro de sus interacciones con la tienda en línea.

## 9. Administrador de sistemas

- **Configuración del servidor:** Crear e instalar el servidor web y la base de datos necesarios para el proyecto, utilizando tecnologías como Apache, Nginx, MySQL, PostgreSQL u otros según los requisitos del sistema.
- **Gestión de bases de datos:** Diseñar y crear las bases de datos necesarias para almacenar la información del sitio web, incluyendo tablas para usuarios, productos, pedidos, cupones, entre otros.
- **Monitoreo y mantenimiento del sistema:** Implementar herramientas de monitoreo del sistema para supervisar el rendimiento del servidor, la disponibilidad del sitio web y la integridad de los datos en tiempo real.



## Control de Calidad (QA)

### 1. Tester 1

- **Pruebas de caja blanca:** Desarrollar casos de prueba utilizando PHPUnit para evaluar la funcionalidad de los componentes individuales y las clases del sistema, asegurando que cada método y función cumpla con sus especificaciones. Crear pruebas unitarias que cubran todos los caminos posibles de ejecución dentro del código, incluyendo casos de éxito y casos de error para garantizar una cobertura exhaustiva. Implementar assertions para verificar que el comportamiento esperado de las funciones y métodos se cumple en todas las circunstancias, utilizando afirmaciones específicas de PHPUnit como assertEquals, assertTrue, assertFalse, entre otras.
- **Cobertura:** La cobertura de las pruebas del Tester 1 será toda la parte pre-autenticada y la función post-autenticadas de “Mi perfil”.
- **Colaboración:** Colaborar con el equipo de desarrollo para implementar soluciones y parches de seguridad para corregir las vulnerabilidades identificadas, asegurando que se apliquen las mejores prácticas de seguridad en todo momento.

### 2. Tester 2

- **Pruebas de caja blanca:** Desarrollar casos de prueba utilizando PHPUnit para evaluar la funcionalidad de los componentes individuales y las clases del sistema, asegurando que cada método y función cumpla con sus especificaciones. Crear pruebas unitarias que cubran todos los caminos posibles de ejecución dentro del código, incluyendo casos de éxito y casos de error para garantizar una cobertura exhaustiva. Implementar assertions para verificar que el comportamiento esperado de las funciones y métodos se cumple en todas las circunstancias, utilizando afirmaciones específicas de PHPUnit como assertEquals, assertTrue, assertFalse, entre otras.
- **Cobertura:** La cobertura de las pruebas del Tester 2 será las funcionalidades de la parte post-autenticada: Rol vendedor, Rol usuario normal, Rol usuario soporte técnico y Rol administrador.
- **Colaboración:** Colaborar con el equipo de desarrollo para implementar soluciones y parches de seguridad para corregir las vulnerabilidades identificadas, asegurando que se apliquen las mejores prácticas de seguridad en todo momento.

### 3. Tester 3

- **Pruebas de caja blanca:** Desarrollar casos de prueba utilizando PHPUnit para evaluar la funcionalidad de los componentes individuales y las clases del sistema, asegurando que cada método y función cumpla con sus especificaciones. Crear pruebas unitarias que cubran todos los caminos posibles de ejecución dentro del código, incluyendo casos de éxito y casos de error para garantizar una cobertura exhaustiva. Implementar assertions para verificar que el comportamiento esperado de las funciones y métodos se

cumple en todas las circunstancias, utilizando afirmaciones específicas de PHPUnit como `assertEquals`, `assertTrue`, `assertFalse`, entre otras.

- **Cobertura:** La cobertura de las pruebas del Tester 3 será las funcionalidades de la parte post-autenticada: Proceso de compra, artículos y Mis pedidos.
- **Colaboración:** Colaborar con el equipo de desarrollo para implementar soluciones y parches de seguridad para corregir las vulnerabilidades identificadas, asegurando que se apliquen las mejores prácticas de seguridad en todo momento.

○

#### 4. Pentester

- **Pentesting:** Realizar pruebas de penetración en la aplicación web para identificar vulnerabilidades de seguridad, como inyecciones SQL, XSS, CSRF, entre otras, utilizando herramientas como Burp Suite, OWASP ZAP o similares.
- **Auditoría de código:** Analizar el código fuente de la aplicación en busca de posibles vulnerabilidades de seguridad, como la falta de validación de entradas, la exposición de datos sensibles o la ejecución de funciones peligrosas.
- **Generación de informe:** Documentar de manera detallada todas las vulnerabilidades encontradas, incluyendo su descripción, impacto potencial y posibles medidas de mitigación, para que puedan ser abordadas por el equipo de desarrollo.
- **Colaboración:** Colaborar con el equipo de desarrollo para implementar soluciones y parches de seguridad para corregir las vulnerabilidades identificadas, asegurando que se apliquen las mejores prácticas de seguridad en todo momento.

## **NOTAS**

- Para la realización del proyecto no se pueden utilizar frameworks como Laravel o Angular. Únicamente se podrán utilizar librerías como JQuery o Bootstrap.
- Cualquier utilización de una librería o plugin de terceros se deberá de consultar con el profesor.
- Cada alumno deberá de crear más de 4 tareas en el Trello de su parte de la aplicación e ir moviéndolas en las diferentes secciones conforme avance el proyecto.
- Antes de empezar a “picar código” los alumnos deberán de definir los requisitos de la aplicación, así como la estructura.
- Las personas dentro de Control de Calidad (QA) deberán de subir al repositorio toda su parte (código de pruebas, documentos, etc).
- Cada semana se realizará una reunión siguiendo la metodología ágil SCRUM donde el profesor tomará nota de los avances de cada uno en el proyecto, dato que posteriormente se utilizará para definir la nota de cada alumno.
- Los alumnos tienen total libertad para implementar cualquiera de las herramientas vistas en la parte teórica de la asignatura como Jenkins, Ansible, Docker, Kubernetes, etc.

## **FORMATO DE ENTREGA**

Al ser un proyecto grupal no se deberá entregar como las prácticas anteriores. El alumno deberá actualizar el repositorio mediante commits y mover las tareas del Trello constantemente para ver el desarrollo del proyecto.

El día de entrega del proyecto, el profesor realizará un pentesting a la aplicación web y revisará cada parte del proyecto de forma minuciosa. Posteriormente, consultará el repositorio de GitHub y el Trello para ver el continuo proceso de desarrollo.

La forma de evaluación del proyecto será tanto el trabajo realizado cada día en clase como la identificación y explotación de vulnerabilidades por parte del profesor en la aplicación web.

**NOTA:** En caso de que la aplicación funcione correctamente sin vulnerabilidades el día de la corrección, se subirá 0'5 de la nota final a todos los alumnos.