

PONTE A PRUEBA

CFGS DAW

Módulo 07: Desarrollo web en entorno servidor

1S 2024/2025



1. ¿EN QUÉ CONSISTE EL EJERCICIO?

¡Te doy la bienvenida a este ejercicio extra!

- Este ejercicio es NO EVALUABLE y OPTATIVO (no es obligatorio), por lo que lo puede hacer cualquier persona que quiera aceptar el desafío independientemente del transcurso del módulo oficial.
- Al no ser evaluable, no hay que entregarlo ni por mensaje ni por tarea en la plataforma.
- Es muy sencillo comprobar que funciona.
- Es un ejercicio individual con trasfondo colaborativo. Le acompañará un foro donde poder plantear las dudas que os surjan y donde podéis participar tanto para preguntar como para responder.
- El profesor no atenderá mensajes sobre este ejercicio, aunque sí participará eventualmente en el foro para responder algunas dudas.
- Al no ser evaluable, podéis darle vuestro toque personal: ampliar o recortar funcionalidad, cambiar requisitos, tomar decisiones que creáis convenientes, etc.
- En principio está diseñado para hacerse con la tecnología vista en el módulo, pero sois libres, como dice el punto anterior, de hacerlo con otra/s tecnologías.
- Como consejo, realízalo para poner a prueba tus conocimientos, pero que no te suponga una sobrecarga durante este o sucesivos semestres, lo mejor es llevarlo a cabo sin prisa, cuando tengas tiempo para poder disfrutar de su desarrollo.

2. ENUNCIADO DE LA ACTIVIDAD

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

2.1 Descripción

Se trata de un periódico online donde los usuarios pueden añadir o editar noticias. La aplicación web debe cumplir con los siguientes requisitos:

- La **página inicial** consiste en un listado de noticias de las que se muestra el título, la fecha y el autor. Se debe poder **ordenar** el listado en base a cualquiera de esos campos (fecha, título o autor) tanto en orden ascendente como en orden descendente. También contará con un formulario en la parte superior que permita iniciar sesión directamente en la propia página inicial (mediante un email, una contraseña y un botón submit). También debe aparecer **un botón** que permita acceder a la página para **registrarse**.
- **Todos los visitantes y los usuarios registrados** podrán hacer clic en cada noticia y visualizarla por completo en una página diferente a la inicial. En el detalle de cada noticia (se muestra todo lo relativo a dicha noticia) se debe de poder volver al listado mediante un botón **VOLVER**. En esta página de detalle de la noticia también se debe añadir el **formulario de inicio de sesión** y el **botón de registro**, igual que en la página inicial o, en caso de sesión iniciada, el **nombre del usuario** y el botón de **logout** (no aparecerían, por tanto, en caso de una sesión iniciada, ni el formulario de login ni el botón de registro). Ningún usuario no registrado debe poder añadir, editar o borrar noticias.

- **La base de datos debe ser capaz de guardar las noticias.** Cada noticia contará con, al menos, un título, un cuerpo (desarrollo de la noticia), una fecha y un autor (usuario que debe estar registrado previamente).
- A través del botón de **registro** de la pantalla inicial se accede al formulario de registro en una página diferente. El **registro** consistirá en guardar, al menos, el **email, nombre y contraseña** del usuario en una base de datos. No es necesario cifrar la contraseña. Tras registrarse, se volverá a la pantalla inicial **SIN** hacer un login automático (es decir, no se inicia la sesión con el usuario recién registrado automáticamente, sino que simplemente, se vuelve a la pantalla inicial después del registro).
- Al hacer **login** en la pantalla inicial, se entra en la aplicación como un **usuario registrado** (iniciando una **sesión** que guarde sus datos). En la parte superior de la página principal desaparecerá el formulario y se sustituirá por el nombre del usuario y un botón de **logout** (finalizar sesión). En el detalle de cada noticia también aparecerá este encabezado (nombre de usuario y logout). Recuerda que en esta página inicial se **listan todas las noticias SIN el cuerpo**, es decir, se listarán las noticias mostrando su **título, fecha y autor**, y no se debe mostrar el cuerpo de la noticia.
- En esta página inicial que lista las noticias, tras iniciar una sesión, debe haber un botón para **AÑADIR** una nueva noticia al final del listado. También deben haber dos botones, **EDITAR** y **BORAR**, junto a **cada noticia añadida por el usuario actual**.
- El botón **AÑADIR** nos llevará a un **formulario** donde se pedirán las características de la nueva noticia (al menos, título, cuerpo y fecha), y guardarlo. También se debe guardar en la base de datos el usuario que ha añadido la noticia, de forma que sea solo ese usuario quien lo pueda editar.
- El botón **EDITAR** nos llevará al **formulario** que **recuperará todos los campos** de esa noticia, permitiendo su modificación.
- El botón **BORRAR** eliminará una noticia de la base de datos.

2.2 Tecnologías

El ejercicio está pensado para tecnologías propias de este módulo, principalmente:

- PHP puro
- Laravel (o cualquier otro framework de PHP)
- Incluso se puede realizar con Python (Django) o con Java (Spring)

2.3 Recomendaciones

Un par de recordatorios a modo de recomendación:

1. **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, ya que pueden dar problemas. Solamente se deben utilizar a la hora de mostrar un texto.
2. **Revisa 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. Mejor probar en un Xampp instalado en tu ordenador para asegurarte de que funciona.

2.4 Objetivos

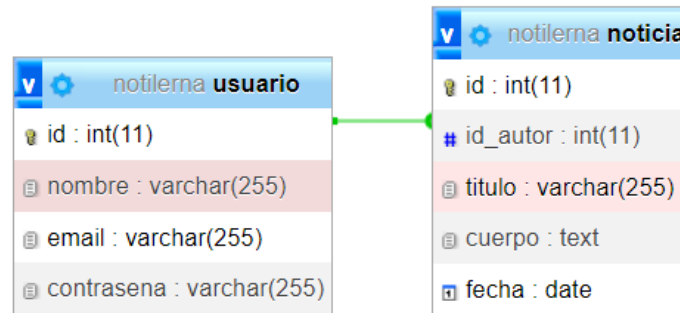
Objetivo principal: Desarrollar un sistema que permita la visualización y gestión de noticias.

Objetivos específicos:

- Desarrollar un portal de noticias visible para todos los visitantes.
- Desarrollar el acceso y registro para usuarios en el sistema.
- Desarrollar un sistema para que cada autor pueda añadir, editar y borrar sus propias noticias (y no las de otros autores)

2.5 Base de datos

El ejercicio funciona con una base de datos con este esquema Entidad-Relación:



- noticia: contiene las noticias
 - id: identificador de la noticia
 - id_autor: identificador del autor de la noticia
 - titulo: título de la noticia
 - cuerpo: desarrollo de la noticia
 - fecha: fecha de la noticia
- usuario: contiene los usuarios
 - id: identificador de usuario
 - nombre: nombre del usuario
 - email: correo electrónico del usuario
 - contraseña: contraseña para iniciar sesión

Si consideráis interesante añadir más campos (fotografía, etiquetas, categorías, etc.) o modificar los existentes, sois libres de rediseñarlo.

Para crear esta base de datos, se adjunta el código en sql.

```

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
  
```

```

/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de datos: `ilernoticias`
--

CREATE DATABASE IF NOT EXISTS `ilernoticias` DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci;
USE `ilernoticias`;

--
-- Estructura de tabla para la tabla `noticia`
--

CREATE TABLE `noticia` (
  `id` int(11) NOT NULL,
  `id_autor` int(11) NOT NULL,
  `titulo` varchar(255) NOT NULL,
  `cuerpo` text DEFAULT NULL,
  `fecha` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Estructura de tabla para la tabla `usuario`
--

CREATE TABLE `usuario` (
  `id` int(11) NOT NULL,
  `nombre` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `contrasena` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Indices de las tablas volcadas
--

ALTER TABLE `noticia`
  ADD PRIMARY KEY (`id`),

```

```
ADD KEY `rel_usuarioid_noticiaidautor` (`id_autor`);

ALTER TABLE `usuario`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT de las tablas volcadas
--
ALTER TABLE `noticia`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
ALTER TABLE `usuario`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

--
-- Restricciones para tablas volcadas
--
ALTER TABLE `noticia`
  ADD CONSTRAINT `rel_usuarioid_noticiaidautor` FOREIGN KEY (`id_autor`) REFERENCES `usuario` (`id`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```