

Portales y Comercio Electrónico

Primer Parcial

Profesor: Santiago Gallino.

Consigna:

Realizar una web dinámica de tema libre (excepto política o religión) que ofrezca un blog / novedades / noticias y un producto / servicio a la venta o un servicio para contratación.

Sitio:

La web debe componerse de dos partes: una parte para los usuarios comunes ("el sitio") y otra parte para la administración ("el admin").

El sitio, al menos, debe:

- Ofrecer a los usuarios algún servicio o producto, como, por ejemplo: un servicio de hosting, servicio de auditoría, servicio de desarrollo, producto con suscripción (ej: un antivirus, una app online como Figma), un videojuego, etc. No es necesaria la implementación de un carrito de compras.
- Incluir una sección de blog / novedades / noticias donde se hable del servicio / producto o de temas relacionados.
- Incluir una home que presente el producto.

El admin, al menos, debe:

- Requerir de una autenticación para acceder.
- Proveer de un ABM para administrar las entradas del blog / novedades / noticias.

Ambos deben incluir una semántica y estructura de HTML correcta, que será motivo de evaluación. Deben ofrecer además estilización en CSS

personalizada por el alumno/a, pudiendo usar un framework de CSS como Bootstrap o Tailwind.

Base de Datos

Debe constar de, al menos, 3 tablas, *usuarios* y otras dos tablas para el blog / novedades / noticias y para los servicios / productos.

La tabla de *usuarios* debe estar constituida de *al menos* 3 campos: uno para el id, uno para el email y otro para el password. Puede utilizarse la que ya viene incluida en el framework.

Al menos una de las otras tablas estar constituida de al menos 5 campos (sin contar PK y los campos de fechas de Laravel).

Toda la creación de tablas, y la carga inicial de datos, deberá estar realizada con **migrations** y **seeders**.

PHP

Deberá usarse la versión actual del framework Laravel aplicando los principios de la programación orientada a objetos, y aprovechando los mecanismos de trabajo ofrecidos por el mismo, siguiendo sus prácticas recomendadas.

Las vistas deberán utilizar el motor de template **Blade** para su renderizado.

Todos los ingresos de datos deberán estar validados, e informar los errores ocurridos, en caso de haberlos.

Debe proveerse de mensajes de feedback al usuario sobre lo ocurrido para facilitar la comprensión de la web.

Se evaluará y tendrá impacto en la nota también:

- Complejidad de la tarea realizada.
- Tablas de relaciones extras en la base de datos.
- Correcta implementación de las herramientas de la Orientación a Objetos (encapsulamiento, herencia, etc).
- Implementación de buenas prácticas para la Programación Orientada a Objetos (ej: Principio de Responsabilidad Única).
- Coherencia en los nombres de variables, clases, métodos, etc.
- **Uso correcto de las etiquetas semánticas de HTML.**
- Documentación apropiada usando phpDoc.
- Estilización del sitio.
- Buena aplicación de los distintos tipos de componentes vistos en clase (Models, Controllers, Middlewares, etc).
- Prolijidad en el código.
- Prolijidad en la organización de la carpeta del proyecto.

Modalidad de entrega

La entrega se realizará de manera digital, subiendo un zip/rar en el Classroom / Campus (según corresponda) en la tarea asignada. Deberá llamarse:

- **"apellido-nombre.[rar|zip]"** (ej: gallino-santiago.rar) en caso de ser un integrante.
- **"apellido-nombre_apellido2-nombre2.[rar|zip]"** (ej: gallino-santiago_noto-federico.rar).

El zip/rar debe contener el proyecto completo, más un archivo **datos.txt** con todos los datos del o los estudiantes:

Carrera, materia, cuatrimestre, año, turno, comisión, apellido y nombre, docente, carácter de entrega (1er parcial).

El incumplimiento de **cualquiera** de las condiciones de entrega estipuladas puede incurrir en una reducción de la nota de al menos 1 punto.