

Portales y Comercio Electrónico

Final

Profesor: Santiago Gallino.

Consigna:

Realizar un sitio dinámico de tema libre (excepto política y 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, o algún tipo de e-commerce.
- Incluir una sección de blog / novedades / noticias donde se hable del servicio / producto / e-commerce o de temas relacionados.
- Incluir una home que presente el producto.
- Tener un registro y autenticación de usuarios. **Está prohibido el uso de la interfaz de autenticación y controllers que Laravel provee.**
- Tener un perfil para los usuarios autenticados, donde puedan ver / editar su información.
- Debe incluir la correspondiente pasarela de pago con el modo sandbox de MercadoPago para poder completar la contratación o compra.

Si el sitio es para la venta de un servicio o suscripción:

- **No** es necesaria la implementación de un carrito de compras.
- Los usuarios deben poder administrar su plan de suscripción, incluyendo cancelarlo, cambiarlo, etc, y ver desde cuándo está suscripto. Esta información debe ya estar cargada desde los Seeders para, al menos, algunos usuarios.

Si el sitio es un e-commerce:

- **Sí** es necesaria la implementación de un carrito de compras.
- Los usuarios deben poder ver su carrito de compras, realizar el pedido, y ver un historial de los pedidos realizados.

El admin, al menos, debe:

- Requerir de una autenticación de un usuario administrador para acceder.
- Proveer de un ABM para administrar las entradas del blog / novedades / noticias.
- Poder ver los usuarios, y de tener uno, el servicio que tienen contratado, en el caso del servicio para contratar, o el historial de compras realizada, en el caso del e-commerce.
- Tener un "Dashboard" inicial que muestre algunas estadísticas relevantes (ej: plan con más suscriptores, productos más comprados, mes con mayor facturación, etc).

Ambos **deben incluir una semántica y correcta estructura de HTML**, 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 tener como nombre:

- Si es un solo integrante: **apellido_nombre**
- Si es un grupo: **apellido1_apellido2**

Debe constar de las tablas necesarias para modelar todos los requerimientos.

Debe haber una tabla de usuarios constituida de al menos 3 campos: uno para el id, uno para el nombre de usuario, otro para el password. Puede utilizarse la que ya viene incluida en el framework.

Al menos una de las otras tablas debe 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.

Al menos un usuario debe tener un servicio contratado, cargado mediante seeders.

Debe haber relaciones entre tablas.

PHP:

Debe 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 **deben** estar validados, e informar los errores ocurridos, en caso de haberlos. **No puede usarse la validación de HTML.**

La tabla del blog / novedades / noticias debe poder hacer alta/modificación de imágenes.

Debe haber al menos 2 roles para los usuarios: usuario común y administrador. Para la verificación del usuario administrador, debe realizarse un Middleware personalizado que haga tal verificación.

Debe haber modelos relacionados, según las relaciones que existan en la base de datos.

Se evaluará y tendrá impacto en la nota también:

- Complejidad de la tarea realizada.
- Accesibilidad, navegabilidad y usabilidad del sitio.
- Tablas de relaciones extras en la base de datos.
- Implementación de los principios de la Orientación 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).
- Utilización adecuada de Ajax para mejorar la performance del sitio.
- Prolijidad en el código y carpeta del proyecto.

Modalidad de entrega:

La entrega se realizará de manera digital, en el DV Panel según sea indicado por la Escuela, en un archivo con los siguientes requerimientos:

- Debe tener el nombre con el formato:
 - **apellido-nombre_final_.[rar|zip]** en caso de ser un solo integrante.
 - **apellido-nombre_apellido-nombre_final_.[rar|zip]** en caso de ser 2 o más integrantes.
 - Ej:
 - gallino-santiago_final_.rar

- **gallino-santiago_noto-federico_final_.rar**
- El archivo de la entrega debe contener:
 - **Todos** los archivos necesarios para correr el proyecto (incluyendo carpetas como **vendor** para acelerar el proceso de corrección).
 - Un archivo **datos.txt** con la siguiente información:
 - Carrera, materia, cuatrimestre, año, turno, comisión, apellido y nombre (todos los integrantes), docente, carácter de entrega (final).

El incumplimiento de la modalidad de entrega **en cualquiera de sus puntos estipulados** puede incurrir en una penalización en la nota de *al menos* un punto.

De detectarse un trabajo copiado, tanto copiador como copiado recibirán automáticamente una calificación de 1 (uno).

Quedará a discreción del profesor si es necesario realizar preguntas al alumno, ya sean con respecto a cómo se encaró la entrega, como teóricas de la materia o materias relacionadas (ej: php, HTML, CSS).