

# Portales y Comercio Electrónico

## Segundo Parcial

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. 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.
- Tener un registro y autenticación de usuarios. **Está prohibido el uso de la interfaz de autenticación y controllers que Laravel provee.**

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 un listado de los usuarios. Debe poder entrar a verse un detalle de cada usuario que incluya el servicio que tienen contratado o las compras realizadas, en caso de tenerlas.

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, 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 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 deberán 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).

- Prolijidad en el código y 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 (2do parcial).

El proyecto **no debe incluir la carpeta del link simbólico [public/storage]**.

El incumplimiento de la modalidad de entrega **en cualquiera de sus puntos estipulados** puede incurrir en una penalización en la nota de *a/ 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 los exámenes se corrigen en clase o no. También quedará a su discreción si es necesario realizar preguntas al alumno, ya sean con respecto a cómo se encaró la entrega, como teóricas.