# Aplicaciones Enriquecidas

## Final

**Profesor: Santiago Gallino.**

**Consigna:**

Realizar un sitio dinámico de tipo e-commerce de tema libre (excepto política y religión). Debe tener un contenido principal ("sitio") para ser consumido por los visitantes, y un panel de administración ("panel").

El "sitio" debe contener:

* una home que sirva como presentación de la web.
* un listado de productos, que incluya la posibilidad de ingresar a cada uno a ver más detalles y agregarlos a su carrito de compras personal.
* un sistema de registro y autenticación de usuarios "comunes" (no adminsitradores) que de acceso a un perfil y el carrito de compras.

El "admin" debe contener:

* un formulario de autenticación que requiera un usuario administrador.
* un ABM de los productos del sitio.
* un historial de pedidos realizados.
* un panel de estadísticas, que debe estar testeado (más sobre esto después).

Los usuarios "comunes" debe poder:

* registrarse y autenticarse en el sitio.
* acceder a su perfil, donde puedan editar sus datos.
* agregar o eliminar productos de su carrito, o modificar su cantidad.
* ver sus pedidos pasados.
* completar sus pedidos con la interfaz de prueba de MercadoPago vista en clase, que deje asentado en la base de datos que fue "pagado".

Las estadísticas deben:

* mostrar información en un "dashboard" (tablero) sobre diferentes estadísticas del sitio, como pueden ser: Productos más comprados, cantidad de pedidos realizados en un X periodo de tiempo, importe promedio de cada pedido, meses con mayor actividad, etc. Por supuesto, pueden haber pedidos ya cargados a través de seeders en la base que ayude a demostrar la funcionalidad del mismo.
* estar cubiertos con tests los métodos que obtengan las diferentes estadísticas, para asegurar su correcto funcioamiento.
* opcionalmente, los usuarios "comunes" pueden tener acceso a estadísticas personales.

El trabajo debe estar realizado íntegramente utilizando el framework Laravel versión 8. El sitio debe utilizar una estructura de HTML5 correcta, y presentar estilización en CSS (no se aceptarán trabajos en HTML pelado).

Esto debe cumplir:

Base de Datos

La base de datos debe estar normalizada, y diseñada correctamente siguiendo las buenas prácticas típicas.

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

PHP

Deberá usarse el framework Laravel 8 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.

La tabla de los productos debe poder hacer alta/modificación de imágenes.

Para la verificación del usuario administrador, debe realizarse un Middleware personalizado que haga tal verificación.

Testing

Deben incluirse tests unitarios (unit tests) que cubran, al menos, los métodos que sea usados para obtener estadísticas.

Deberá además aplicarse cualquier corrección realizada por el docente sobre el segundo TP.

Se evaluará y tendrá impacto en la nota también:

* Complejidad de la tarea realizada.
* Tablas de relaciones extras en la base de datos.
* Implementación de los principios de la Orientación a Objetos (encapsulamiento, herencia, etc).
* Coherencia en los nombres de variables, clases, métodos, etc.
* **Uso correcto de las etiquetas semánticas de HTML (puede ser causa de desaprobación).**
* 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.
* Prolijidad en la organización de la carpeta del proyecto.
* Cantidad y utilidad de las estadísticas.
* Manejos especiales de distintos resultados de la API de MercadoPago (ej: si se "Aprobó" o no un pago, en base a los estados que la API proporciona).

***Importante***

El final puede requerir una defensa oral por parte del o los/as alumnos/as. Esto puede incluir:

* Preguntas teóricas sobre los contenidos de la materia.
* Preguntas teóricas sobre materias anteriores relacionadas a esta (HTML, CSS, JS, php, etc).
* Preguntas sobre la realización del trabajo.

Modalidad de entrega

La entrega se realizará de manera digital, subiendo el trabajo en el panel de Da Vinci antes o al comienzo de la mesa de examen, según lo indicado por la Escuela. Debe presentarse en un zip/rar, cuyo nombre debe tener el formato (**requisito para aprobar**):

**apellido-nombre\_final\_.rar**

Por ejemplo: gallino-santiago\_final\_.rar

Este zip/rar debe incluir:

* Una carpeta con el proyecto entero (**debe** incluir la carpeta **vendor**, para asegurarse de que la app pueda probarse sin requerir descargar nada adicional vía Composer u otro método).
* Archivo **datos.txt** con todos los datos del o los estudiantes:

Apellido y nombre, profesor, materia, carácter de entrega (final), cuatrimestre, año, carrera.

El incumplimiento de cualquiera de las condiciones de entrega va a recibir una penalización en la nota que va desde *al menos* un punto menos hasta la desaprobación del mismo.