

# Desarrollo de Aplicaciones Web con PHP

## Laboratorio :: Requerimientos Curso 2025

### 1. Información Administrativa

La tarea comienza el **jueves 24 de abril** y finaliza el **domingo 15 de junio**. Para la evaluación se tendrá en cuenta el producto final logrado, el avance de los grupos en los monitoreos, la documentación y la presentación final (presentación, demo y defensa) el día **lunes 23 de junio**.

### 2. Requerimientos Funcionales

Se pide realizar **todas** las funcionalidades descritas en la visión del laboratorio.

### 3. Requerimientos No Funcionales

A continuación se brinda una lista de requerimientos no funcionales, algunos serán obligatorios y otros opcionales.

#### **Requerimientos Obligatorios**

##### **Diseño responsivo para múltiples dispositivos**

El sitio deberá adaptarse correctamente a diferentes tamaños de pantalla y dispositivos, permitiendo que tanto compradores como rematadores puedan participar o seguir las subastas desde computadoras, tablets o smartphones, sin perder funcionalidad ni claridad visual.

##### **Transmisión en vivo embebida**

Cada subasta será transmitida en vivo desde la misma página de la subasta, utilizando servicios como YouTube Live, Vimeo, Twitch, etc.

##### **Notificaciones automáticas por email**

El sistema debe ser capaz de enviar correos automáticos para confirmar registros, avisar del inicio de subastas favoritas, informar pujas superadas, confirmar a los ganadores.

##### **Panel de estadísticas para casas de remate**

Cada casa podrá acceder a un panel de visualización de datos donde consultar estadísticas sobre sus subastas por categoría, monto, artículos vendidos, participación, etc. Este panel debe incluir gráficos simples y filtros de análisis por rango de tiempo o tipo de producto.

##### **Alertas automáticas al superar ciertos montos**

Durante una subasta activa, la casa de remate podrá definir umbrales de interés para determinados artículos o lotes. El sistema deberá enviar una notificación interna si la

puja supera ese monto, ayudando a anticipar si un artículo tendrá buena rentabilidad o si conviene extender el tiempo.

### **Interactividad en tiempo real con WebSockets**

El sistema debe implementar comunicación en tiempo real mediante WebSockets para mejorar la experiencia interactiva durante las subastas. Esto se aplica a dos funcionalidades fundamentales: puja en tiempo real y chat directo con la casa de remate.

## **Requerimientos Opcionales**

### **Transformación en Aplicación Progresiva (PWA) [2 puntos]**

Se espera que la plataforma pueda ser instalada como una app en dispositivos móviles. Los usuarios registrados podrán iniciar sesión con su cuenta, y participar de subastas desde la pantalla de inicio de su teléfono.

### **Autenticación con redes sociales (OAuth) [2 puntos]**

La plataforma debe ofrecer la posibilidad de autenticarse usando cuentas de Google o Facebook. Esto facilitará el ingreso a usuarios nuevos, agilizando el registro para poder participar rápidamente en una subasta.

### **Visualización geográfica de productos [2 puntos]**

Cada artículo o lote subastado podrá estar asociado a una ubicación geográfica, permitiendo visualizar la distribución de productos sobre un mapa embebido. Los usuarios podrán aplicar filtros geográficos para encontrar productos cercanos o ver de dónde provienen.

### **Colas de procesamiento para pujas y notificaciones [3 puntos]**

Dado que las pujas deben registrarse en tiempo real y con precisión, el sistema debe contar con un sistema de colas como Redis para manejar múltiples eventos simultáneos. Esto asegurará que cada puja sea procesada correctamente en el orden recibido, sin solapamientos.

### **Base de datos NoSQL complementaria para logs e historiales [2 puntos]**

Además de la base de datos relacional central, se podrá emplear una base de datos NoSQL para guardar registros de logs del sistema e historial de interacción.

### **Agenda y recordatorios personalizados para rematadores [2 puntos]**

Los rematadores deberán tener acceso a una agenda digital dentro del sistema, donde puedan gestionar subastas futuras, recibir notificaciones de eventos confirmados o pendientes de aprobación, y obtener recordatorios con anticipación.

### **Automatización puja basada en presupuesto [2 puntos]**

El sistema debe permitir a los compradores establecer un máximo que están dispuestos a pagar por un artículo, dejando que el sistema realice automáticamente pujas mínimas que superen al actual mejor postor hasta alcanzar su límite.

### **Manejo de husos horarios [1 punto]**

Dado que los usuarios y rematadores pueden estar en distintas regiones, la plataforma deberá mostrar las fechas y horas de subastas ajustadas al huso horario del usuario, para evitar confusiones en el inicio o fin de eventos.

**Dockerización del sistema para despliegue portable [3 puntos]**

El desarrollo podrá ser encapsulado en contenedores Docker, permitiendo que el sistema se despliegue de manera fácil y uniforme en distintos entornos de prueba o producción, incluyendo web, base de datos y servicios complementarios.

**Integración con pasarelas de pago o simulación de cobros [2 puntos]**

La plataforma debe contemplar la posibilidad de integrar una pasarela de pago real (Stripe, PayPal, etc.) o un mecanismo simulado, para permitir la realización de pagos y emisión de facturas una vez finalizada la subasta.

**Publicación en la nube (con soporte HTTPS) [2 puntos]**

La plataforma podría desplegarse en un entorno de nube gratuito que permita acceso público con HTTPS (como Render, Vercel, Firebase Hosting, Railway o servicios de AWS/GCP).

**Arquitectura distribuida con frontend desacoplado [3 puntos]**

El sistema podrá organizarse en una arquitectura distribuida, separando el frontend (desarrollado en Angular, React o Vue) del backend que expone una API RESTful. El frontend deberá comunicarse exclusivamente con la API mediante solicitudes HTTP.

Para aprobar el Laboratorio, los grupos de 3 estudiantes deberán completar como mínimo 12 puntos de los Requerimientos No Funcionales. Los grupos de 4 deberá completar 16 puntos.