

# Taller 1 – Rúbrica y Condiciones de Entrega

Taller Frontend con IA, JSON y Servidor Apache

Asignatura: Desarrollo Web y Móvil

## Introducción

Durante la tercera semana del curso de Desarrollo Web y Móvil, los estudiantes se enfrentan a un desafío orientado a la aplicación práctica de los conceptos fundamentales del desarrollo frontend sin el apoyo de frameworks de alto nivel. El propósito de esta actividad es que los alumnos comprendan y experimenten con las bases del desarrollo web, fortaleciendo su capacidad de construir soluciones funcionales a partir de herramientas esenciales como HTML, CSS y JavaScript, además del uso de un servidor Apache como entorno de ejecución.

En esta etapa, los estudiantes deben transformar ideas en productos digitales concretos. A partir de casos propuestos —como la creación de un clon simplificado de Amazon, una plataforma de encuestas estudiantiles o un sistema de reservas— se espera que sean capaces de idear, diseñar y validar aplicaciones que simulen escenarios reales. Cada grupo debe elaborar previamente un conjunto de *mockups* en herramientas como Figma, los cuales representarán la estructura visual y la interacción prevista en sus aplicaciones. Dichos prototipos constituyen la base para la posterior implementación en código, asegurando coherencia entre la planificación y el producto final.

Un aspecto central de esta semana es el uso de la inteligencia artificial como apoyo en la transición desde los diseños iniciales hasta la implementación en HTML, CSS y JavaScript. Los estudiantes deberán documentar los *prompts* utilizados y reflexionar sobre su efectividad, desarrollando así una mirada crítica acerca de las ventajas y limitaciones de las herramientas de IA en el contexto del desarrollo web. Al mismo tiempo, se enfatiza el uso de archivos JSON como mecanismo para manejar datos de manera local, permitiendo simular interacciones dinámicas sin necesidad de un backend.

La evaluación no se centra únicamente en la capacidad técnica, sino también en la organización, claridad y responsabilidad en la gestión del trabajo. Por ello, los estudiantes deben mantener su proyecto en un repositorio de GitHub, donde se evidencie el proceso de construcción mediante commits descriptivos, ramas de trabajo y un archivo README que detalle las instrucciones de despliegue y uso. Finalmente, la presentación de los proyectos permitirá valorar no solo el resultado final, sino también la capacidad de los grupos para comunicar sus decisiones de diseño, los aprendizajes obtenidos y la forma en que abordaron los desafíos planteados.

Con esta actividad, los estudiantes desarrollan competencias técnicas y comunicacionales, adquiriendo la experiencia necesaria para comprender los fundamentos del desarrollo frontend y sentar las bases para la incorporación de frameworks más avanzados en etapas posteriores del curso.

## 1. Condiciones específicas de la Semana 3

- **Sin framework frontend ni backend:** No se permite usar frameworks de frontend (p. ej., React, Angular, Vue) ni un backend (Django, FastAPI, Node, etc.). Se evalúa la **lógica en JavaScript puro**.
- **Servidor web Apache:** Deben servir la app desde **Apache** (XAMPP/WAMP/LAMP). Los datos se consumen con **fetch** sobre **http://**.
- **Datos en JSON:** Carga y persistencia simulada con archivos **.json** (catálogos, encuestas, reservas, etc.) y **localStorage** para estado de sesión (carrito, favoritos, respuestas, etc.).
- **UI responsiva y validaciones:** Uso consistente de **Bootstrap o Tailwind**, **modales** para confirmaciones y **mensajes de validación** visibles en formularios.
- **Casos propuestos:** Implementar los casos definidos para el curso (p. ej., Mini-Marketplace, Encuestas, Citas Médicas, Canchas, Libro de Clases Kinder) con **4–6 vistas** según el enunciado de cada uno.
- **Versionamiento:** Repositorio en **GitHub** con README (setup, estructura, decisiones), ramas y commits descriptivos.

## 2. Entregables obligatorios

1. **Mockups** (Figma o similar) de todas las vistas implementadas (4–6). Exportar enlace/imagenes.
2. **Documento de Prompts de IA** usados para generar/ajustar HTML, CSS y JS (incluye contexto y justificación).
3. **Proyecto Frontend** servido por Apache: HTML, CSS, JS, **data/\*.json**, y README en GitHub.

### 3. Rúbrica de evaluación (ponderación total = 100 %)

Criterio (Peso)	Descriptorios de logro
1. Entrega de Mockups (12.5 %)	<p><b>Excelente (100 %):</b> Mockups completos de todas las vistas (4-6), diseño coherente y responsivo.</p> <p><b>Bueno (75 %):</b> Mayoría de vistas; diseño entendible con faltas menores.</p> <p><b>Aceptable (50 %):</b> Incompletos (&lt;4 vistas) o poco detallados.</p> <p><b>Insuficiente (25 %):</b> No presenta mockups o irrelevantes.</p>
2. Prompts de IA (12.5 %)	<p><b>Excelente:</b> Documento claro con todos los prompts usados, justificados y organizados.</p> <p><b>Bueno:</b> La mayoría entregados; organización parcial.</p> <p><b>Aceptable:</b> Pocos o desordenados; difícil seguimiento.</p> <p><b>Insuficiente:</b> No entrega prompts o irrelevantes.</p>
3. Requerimientos Funcionales (12.5 %)	<p><b>Excelente:</b> Cumple todo: 4-6 vistas, validaciones visibles, modales, toasts, JSON y <code>localStorage</code>.</p> <p><b>Bueno:</b> Falta 1-2 funcionalidades.</p> <p><b>Aceptable:</b> 50 % de funcionalidades; errores de validación/persistencia.</p> <p><b>Insuficiente:</b> &lt;50 % o app no funcional.</p>
4. Calidad de Código (12.5 %)	<p><b>Excelente:</b> Código organizado/modular; buen uso de utilidades CSS; funciones JS claras.</p> <p><b>Bueno:</b> Orden general; algunas redundancias.</p> <p><b>Aceptable:</b> Desorden/poca modularidad; funcional con malas prácticas.</p> <p><b>Insuficiente:</b> Incompleto/desordenado con errores graves.</p>
5. Validaciones y Errores (12.5 %)	<p><b>Excelente:</b> Todas las validaciones funcionan con mensajes claros.</p> <p><b>Bueno:</b> Mayoría funciona; faltan casos/mensajes.</p> <p><b>Aceptable:</b> Solo básicas; muchas no funcionan.</p> <p><b>Insuficiente:</b> Sin validaciones o no operativas.</p>
6. UI/UX (12.5 %)	<p><b>Excelente:</b> Interfaz limpia, responsiva y coherente; componentes bien usados.</p> <p><b>Bueno:</b> Responsiva y clara; inconsistencias menores.</p> <p><b>Aceptable:</b> Básica; problemas de responsividad/consistencia.</p> <p><b>Insuficiente:</b> Pobre o sin estilos coherentes.</p>
7. GitHub (12.5 %)	<p><b>Excelente:</b> Commits descriptivos, ramas feature/PRs, issues; README completo.</p>

*Nota:* Los estudiantes deben adjuntar enlaces/archivos de mockups y un documento con los **prompts** utilizados para IA. El repositorio GitHub debe contener instrucciones para desplegar con **Apache** y ejemplos de **fetch** a JSON.