



Tecnicatura Universitaria en Programación

Programación II

Propuesta de Trabajo Práctico

Introducción

En el marco de la asignatura Programación II de la Tecnicatura Universitaria en Programación, este trabajo práctico tiene como objetivo integrar los conocimientos adquiridos en el desarrollo frontend, aplicando las mejores prácticas y tecnologías modernas. Los estudiantes trabajarán en equipo para crear una aplicación web funcional, responsiva y accesible, demostrando su capacidad para resolver problemas reales mediante soluciones técnicas robustas.

Modalidad de Trabajo

- Trabajo grupal obligatorio: Los equipos deberán estar conformados por 3 a 4 personas. Esta dinámica fomenta la colaboración, el uso de sistemas de control de versiones (Git) y la división de tareas bajo una metodología ágil.
- Fecha límite de entrega: 12 de septiembre de 2025 (Semana 7). Se evaluará puntualidad, cumplimiento de requisitos y calidad técnica.

Objetivos Ampliados

El proyecto busca que los estudiantes:

1. Implementen arquitecturas frontend escalables usando HTML5 semántico.
2. Dominen técnicas avanzadas de maquetación con CSS Grid/Flexbox.
3. Apliquen principios de UX/UI en formularios complejos con validación en tiempo real.
4. Integren APIs mediante Fetch API/Async-Await con manejo de estados asíncronos.
5. Desarrollen estrategias responsive-first con dark/light mode persistente.
6. Experimenten con animaciones CSS/JS para mejorar la interactividad.

Tecnologías Requeridas

- HTML5 semántico
- CSS3 (Flexbox/Grid/CSS Custom Properties)
- JavaScript ES6+ (Módulos, Fetch API, Async/Await)

Estructura del Proyecto

Requisitos Mínimos (Base)

- 4 páginas HTML con:
 - Estructura semántica: header, nav, main, footer
 - Maquetación con Flexbox y Grid (combinados)
 - Favicon personalizado e imágenes optimizadas (WebP format)

Formulario Avanzado

- Formulario con:
 - 6-8 campos (combinando diferentes tipos: text, email, tel, date, select, checkbox group, radio group)
 - Validación en tiempo real con JavaScript (obligatoriedad + formato + validación cruzada)
 - Feedback visual avanzado (mensajes contextuales, iconos de estado)
 - Simulación de envío con Fetch API a un mock API (JSON Server o similar)

Diseño Responsive

- 4 breakpoints:
 - PC (>1200px) - Diseño desktop completo
 - Laptop (992px) - Adaptaciones menores
 - Tablet (768px) - Menú hamburguesa, reorganización
 - Celular (<480px) - Stack vertical, tamaños ajustados
- Dark/Light mode toggle con persistencia en localStorage

Efectos y Animaciones (elegir al menos 2)

- Animaciones @keyframes complejas (secuencias)
- Micro-interacciones (hover, focus, active)
- Transformaciones 3D (perspective, rotateX/Y)
- Scroll animations (Intersection Observer)
- Loaders y transiciones entre páginas

Extras Opcionales Valorados (Desafíos Adicionales)

- Implementación de un carrito de compras con localStorage
- Gráficos o visualización de datos con Chart.js
- Uso de Web Components (Custom Elements)
- Internacionalización (i18n) básica
- Implementación de un buscador con filtrado en tiempo real
- Sistema de rating con estrellas interactivo
- Galería de imágenes con lightbox

Estructura de Archivos Recomendada

```
/proyecto
├── README.md
├── index.html
├── /css (archivos de estilos)
├── /js (archivos .js)
├── /assets
│   ├── /images
│   ├── /icons
│   └── /fonts
├── /pages (otros archivos .html)
└── /mock (para datos de prueba)
```

Entregables

- Repositorio Git público con código
- README claro con:
 - Descripción del proyecto
 - Tecnologías usadas
 - Instrucciones para ejecutarlo
 - Capturas de pantalla
- Lista de funcionalidades implementadas
- Lista de extras implementados (si aplica)



Algunas ideas para Proyectos:

1. Plataforma de cursos online (con carrito)
2. Galería de fotos interactiva
3. Clima app con API pública
4. Task manager avanzado
5. E-commerce frontend (sin backend real)
6. Blog personal con modo oscuro
7. Portfolio interactivo
8. Juego simple en navegador