



UNIVERSIDAD AUTÓNOMA DE SINALOA
FACULTAD DE INGENIERIA MOCHIS
LICENCIATURA EN INGENIERIA DE SOFTWARE



PROGRAMA DE ESTUDIO

1. DATOS DE IDENTIFICACIÓN			
UNIDAD DE APRENDIZAJE O MÓDULO:	INTRODUCCIÓN AL DESARROLLO WEB		
Clave:			
Ubicación:	Semestre VI	Área: Profesionalizante	
Horas y créditos:	Teóricas: 40	Prácticas: 40	Estudio Independiente: 80
	Total de horas: 160		Créditos: 10
Competencia(s) del perfil de egreso al que aporta:	CG9. Desarrolla nuevos enfoques interdisciplinarios y construye propuestas innovadoras a partir de la transdisciplina. CG10. Asume con responsabilidad y ética el manejo de las Tecnologías del Aprendizaje y el Conocimiento y es capaz de reconducir las Tecnologías de la Información y Comunicación para la adquisición y actualización del conocimiento de manera permanente para su vida y su profesión. CE12. Seleccionar estándares de desarrollo de las etapas de diseño, codificación, pruebas y mantenimiento para resolver un caso de estudio, analizando sus características. CE13. Aplica conocimientos teóricos y prácticos, en conjunto con las metodologías para la buena construcción de programas y sistemas de software, considerando su análisis y diseño, planeación, procesos y controles dentro de escenarios de pruebas para asegurar confiabilidad, funcionalidad, costo, seguridad, facilidades de mantenimiento y otros aspectos relacionados. CE18. Investiga, reconoce y aplica de manera teórica y práctica las dimensiones de la metodología de UX, por sus siglas en inglés User Experience o bien Experiencia de usuario para el diseño de interacción y arquitectura de información.		
Unidades de aprendizaje relacionadas:	Administración de proyectos de software Diseño de experiencia de usuario Fundamentos de base de datos Lenguajes de programación		
Responsable(s) de elaborar el programa:	Dr. Herman Geovany Ayala Zúñiga	Fecha: 09/07/2025	
Responsable(s) de actualizar el programa:		Fecha:	
2. PROPÓSITO			



Formar profesionales capaces de diseñar y desarrollar soluciones informáticas web de manera ética, segura y eficiente, integrando la evaluación, diseño y despliegue. El estudiante será capaz de aplicar principios técnicos y organizacionales para optimizar recursos computacionales, garantizar la seguridad de los sistemas y proponer soluciones contextualizadas a las necesidades de su entorno.

3. SABERES

Teóricos:	<ul style="list-style-type: none">- Arquitectura cliente-servidor, ciclo de vida HTTP, tipos de contenido web- Estandares abiertos (W3C, WCAG), buenas prácticas de desarrollo y documentación.- Estructura y semántica de HTML5- Modelo visual y responsive con CSS3 (Flexbox/Grid)- Interactividad y asincronía con JavaScript (ES6+), DOM, eventos- Programación en PHP y comunicación con bases de datos (MySQL, PDO)- Seguridad esencial (XSS, CSRF, validación, sanitización, autenticación)- Licenciamiento OSS, privacidad digital y consentimiento informado
Prácticos:	<ul style="list-style-type: none">- Maquetado responsive mobile-first- Formularios accesibles con validación en frontend y backend- Consumo de APIs externas y procesamiento de datos- Desarrollo de CRUD completo seguro en PHP con PDO- Gestión de sesiones y autenticación básica- Integración de formularios con base de datos- Pruebas unitarias básicas, validación con herramientas como Lighthouse- Control de versiones con GitHub, flujo de trabajo ágil- Documentación técnica clara y colaborativa (README, ER, guía de usuario)- Aplicación web contextualizada, inclusiva y documentada- Módulos organizados, funcionalidad probada, criterios técnicos cumplidos
Actitudinales:	<ul style="list-style-type: none">- Conciencia sobre la privacidad, seguridad e impacto tecnológico en comunidades diversas- Participación activa en equipos diversos- Empatía frente a usuarios con distintas necesidades y contextos culturales- Autoevaluación de desempeño técnico y personal- Apertura a la retroalimentación y disposición al aprendizaje permanente- Evaluación de decisiones técnicas desde una perspectiva ética y sostenible- Capacidad para cuestionar y proponer soluciones innovadoras desde el rol social del software

4. CONTENIDOS

Unidad 1: Fundamentos del desarrollo web y entorno de trabajo

- Introducción a la web abierta: cliente-servidor, HTTP, navegadores, ciclo de vida de la petición.
- Herramientas y entorno local: VS Code, XAMPP/Laragon, configuración de workspace.
- Control de versiones con Git y GitHub: flujo básico, ramas, commits, PRs.



- Buenas prácticas en desarrollo: estándares abiertos, ética digital, privacidad de datos.

Unidad 2: Estructura semántica y formularios accesibles con HTML5

- Etiquetas semánticas: <header>, <main>, <article>, <section>, <nav>, <aside>
- Formularios HTML: input types, atributos de accesibilidad, validación nativa, etiquetas <label>, <fieldset>, <legend>
- Integración de multimedia accesible: subtítulos, descripciones, uso correcto de <audio> y <video>.
- Microdatos e introducción al SEO.

Unidad 3: Estilización y diseño responsive con CSS3

- Modelo de caja, cascade, herencia y especificidad.
- Selectores, pseudoelementos, pseudoclasas.
- Flexbox y Grid: estructuras responsive complejas.
- Media queries y diseño mobile-first.
- Variables CSS, animaciones sencillas, dark mode.
- Tokens de diseño y componentes reutilizables.

Unidad 4: Interactividad y lógica cliente con JavaScript ES6+

- Sintaxis moderna: let/const, funciones flecha, objetos literales.
- DOM y eventos: selectores, listeners, manipulación dinámica.
- Módulos ES6 y scope.
- Fetch API, consumo de JSON, asincronía con async/await.
- Validación dinámica de formularios, animaciones basadas en eventos.
- Manejo de errores y depuración en navegador.

Unidad 5: Backend introductorio con PHP y persistencia de datos

- Sintaxis de PHP 8: estructuras, funciones, arrays, manejo de errores.
- Formularios POST/GET, sanitización y validación.
- Gestión de sesiones, login básico con password_hash.
- Conexión segura a MySQL/MariaDB con PDO.
- Implementación de CRUD completo (create/read/update/delete).
- Subida de archivos segura y paginación de registros.
- Introducción a MVC mínimo y rutas simples.

Unidad 6: Accesibilidad, seguridad y pruebas

- Principios OWASP básicos: XSS, CSRF, SQLi y mitigaciones.
- Tokens, sanitización, autenticación robusta.
- Accesibilidad AA con ARIA roles, navegación por teclado.
- Calidad de código: PSR-12, linters, estructura modular.



- Pruebas unitarias (PHPUnit, JS manuales), plan de pruebas.
- Uso de herramientas: Lighthouse, axe DevTools, reportes de accesibilidad.

Unidad 7: Proyecto integrador

- Consolidación del proyecto MVP: revisión técnica, checklist final.
- Documentación técnica: README, instalación, estructura, decisiones.
- Ética y responsabilidad en la entrega: propiedad intelectual, aviso de privacidad, licencias.
- Pitch técnico, demo funcional y retroalimentación cruzada.

5. ACTIVIDADES PARA DESARROLLAR LAS COMPETENCIAS

Actividades del docente:

- Actualización continua en tecnologías web (HTML5, CSS3, JS ES6+, PHP, accesibilidad).
- Diseño de guías prácticas con enfoque inclusivo y ético.
- Revisión y prueba de cada práctica antes de clase.
- Aplicación de rúbricas claras y consistentes.
- Retroalimentación constructiva y oportuna.
- Organización de equipos diversos.
- Dinámicas de revisión por pares y coevaluación.
- Fomentar el respeto a las licencias y la propiedad intelectual.
- Discusión ética sobre uso de datos, seguridad y privacidad.
- Uso de checklist de accesibilidad en cada entrega.
- Incorporación de GitHub como espacio de entrega y retroalimentación.
- Aplicación de metodologías como “aprender haciendo”, “problema-proyecto-producto”.
- Organización de demos y pitch técnico para prácticas de comunicación.

Actividades del estudiante:

- Prácticas individuales con retos progresivos.
- Desafíos semanales de código (“fix-it” accesibilidad o seguridad).
- Proyectos intermedios aplicados.
- Code reviews entre pares.
- Discusión de decisiones técnicas en los equipos.
- Presentación del proyecto integrador con enfoque inclusivo.
- Evaluación de buenas prácticas en accesibilidad (WCAG 2.2).
- Implementación de políticas de privacidad y uso responsable de datos.
- Reflexión sobre el impacto social de la app web desarrollada.
- Mantenimiento de portafolio digital de evidencias.
- Control de versiones, documentación técnica y uso de issues.
- Autoevaluaciones y planes de mejora tras cada entrega.
- Redacción de README claro y profesional.
- Diagrama ER, manual de usuario, aviso de privacidad.
- Preparación de pitch y demo técnico.



UNIVERSIDAD AUTÓNOMA DE SINALOA
FACULTAD DE INGENIERIA MOCHIS
LICENCIATURA EN INGENIERIA DE SOFTWARE



PROGRAMA DE ESTUDIO

6. EVALUACIÓN DE LAS COMPETENCIAS	
6.1. Criterios de desempeño	6.2 Portafolio de evidencias
<p>Desarrolla soluciones web funcionales, accesibles y seguras.</p> <ul style="list-style-type: none">- Cumple estándares de HTML/CSS/JS.- Aplica principios de seguridad básica. <p>Organiza su aprendizaje mediante evidencia y autogestión.</p> <ul style="list-style-type: none">- Presenta portafolio actualizado.- Realiza autoevaluaciones con propuestas de mejora. <p>Colabora de forma activa, respetuosa y comprometida.</p> <ul style="list-style-type: none">- Participa en revisiones de pares.- Respeta códigos de conducta. <p>Documenta y presenta soluciones con claridad y profesionalismo.</p> <ul style="list-style-type: none">- Redacta README claro.- Realiza pitch técnico coherente. <p>Reflexiona sobre el impacto de la tecnología en contextos diversos.</p> <ul style="list-style-type: none">- Analiza implicaciones éticas de su producto.- Propone mejoras desde la inclusión y sostenibilidad.	<ul style="list-style-type: none">- Documento con marco conceptual y principios básicos de desarrollo web.- Infografía sobre buenas prácticas en accesibilidad web (WCAG 2.2).- Reflexión breve sobre el papel del desarrollo web en contextos sociales diversos- Entregas de ejercicios de HTML, CSS, JS y PHP con comentarios explicativos.- Capturas y anotaciones de correcciones realizadas.- Resultados de evaluaciones automáticas (validators, linters, Lighthouse, etc.).- Registro de participación en actividades grupales.- Evidencia de revisiones por pares.- Repositorio del proyecto en GitHub con README técnico y manual de usuario.- Diagrama de arquitectura (ER, navegación, flujo de datos).- Demo explicando funciones, y diseño.
6.3. Calificación y acreditación:	
Parcial: <ul style="list-style-type: none">• Parcial 1 (unidad 1 y 2): 10%• Parcial 2 (unidad 3 y 4): 10%• Parcial 3 (unidad 5 y 6): 10%	Final: <ul style="list-style-type: none">• Evaluaciones parciales (3): 30% (10% cada una)• Prácticas: 25%• Proyecto integrador: 35%• Evaluación final integral: 10%
7. RECURSOS DIDÁCTICOS	
<ul style="list-style-type: none">• Guías de laboratorio• Documentación oficial (MDN Web Docs, php.net, WGAC 2.2)• Videos explicativos y simuladores en línea	



UNIVERSIDAD AUTÓNOMA DE SINALOA
FACULTAD DE INGENIERIA MOCHIS
LICENCIATURA EN INGENIERIA DE SOFTWARE



PROGRAMA DE ESTUDIO

- Plataformas colaborativas (ClickUp, GitHub, Moodle)

8. FUENTES DE INFORMACIÓN

Bibliografía básica

Autor(es)	Título	Editorial	Año	URL o biblioteca digital donde está disponible
Jordi Sánchez Cano	Introducción a la programación web avanzada	UOC		https://infolibros.org/pdfview/introduccion-a-la-programacion-web-avanzada-jordi-sanchez-cano-613/
Marijn Haverbeke	Eloquent JavaScript	No Starch Press	2024	
Jon Duckett	HTML and CSS: Design and Build Websites	Wiley	2011	

Bibliografía complementaria

Autor(es)	Título	Editorial	Año	URL o biblioteca digital donde está disponible
Eric A. Meyer, Estelle Weyl	CSS: The Definitive Guide	O'Reilly Media	2023	
Jon Duckett	PHP & MySQL: Server-Side Web Development	Wiley	2022	

9. PERFIL DEL DOCENTE

Profesional con experiencia en desarrollo de aplicaciones web, familiarizado con prácticas éticas. Capacidad para integrar herramientas modernas en el desarrollo de competencias técnicas y actitudinales. Experiencia en educación superior, diseño de temarios y acompañamiento formativo.



UNIVERSIDAD AUTÓNOMA DE SINALOA
FACULTAD DE INGENIERIA MOCHIS
LICENCIATURA EN INGENIERIA DE SOFTWARE



PROGRAMA DE ESTUDIO