



Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

(Programa del año 2026)
(Programa en trámite de aprobación)
(Presentado el 22/09/2026 09:30:02)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
LABORATORIO DE TECNOLOGÍAS	ING. INFORM.	OCD- 3-2/2 025	2026	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
BAIGORRIA FERNANDEZ, LORENA S.	Prof. Responsable	P.Adj Exc	40 Hs
PERALTA, MARIO GABRIEL	Prof. Co-Responsable	P.Adj Exc	40 Hs
BAGLIONI, VALENTINA	Auxiliar de Práctico	A.1ra Simp	10 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	2 Hs	Hs	4 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
03/08/2026	13/11/2026	15	90

IV - Fundamentación

Laboratorio de Tecnologías los/las estudiantes desarrollan aplicaciones para la Web. Se ubica en el 2.º cuatrimestre de 3.º año, momento en el cual los estudiantes ya cuentan con una sólida formación previa en programación orientada a objetos, manejo de bases de datos, y conceptos de ingeniería de software, adquirida en asignaturas anteriores del plan.

El programa aborda los contenidos mínimos correspondientes al desarrollo de aplicaciones Web; programación en el entorno Web con HTML, lenguajes de script, librerías y frameworks de front-end, programación con lenguajes de backend y acceso a bases de datos, lenguajes de script asincrónicos, Web Services, análisis comparativo de plataformas, estándares del desarrollo Web, arquitecturas cliente/servidor en n capas, seguridad y confidencialidad, Web sintáctica y semántica, diseño de interfaces de usuario y nuevas tendencias tecnológicas; instanciándolos sobre un stack tecnológico actual y ampliamente adoptado por la industria: JavaScript como lenguaje de programación tanto en el cliente como en el servidor, Node.js como entorno de ejecución del lado del servidor, y React (con Vite como herramienta de desarrollo y empaquetado) como framework para la construcción de interfaces de usuario en el cliente, estudiándose además Vue.js de manera comparativa como alternativa de mercado. Dado que se trata del primer contacto de los/las estudiantes con el desarrollo Web, el alcance de React se acota deliberadamente a componentes funcionales y a los hooks básicos (useState, useEffect), dejando fuera de programa herramientas de mayor complejidad (gestión de estado global, renderizado del lado del servidor, meta-frameworks), de modo de priorizar la solidez de los fundamentos por sobre la cobertura exhaustiva del ecosistema. Dado el fuerte carácter de

laboratorio de la materia, el eje metodológico central es el desarrollo incremental de proyectos Web de mediana complejidad, capitalizando los conocimientos previos de bases de datos e ingeniería de software de los/las estudiantes y aplicándolos por primera vez al dominio Web.

V - Objetivos / Resultados de Aprendizaje

Objetivo general

Que el estudiante sea capaz de diseñar, desarrollar, integrar y desplegar aplicaciones Web full-stack de mediana complejidad, aplicando los principios arquitectónicos, estándares y prácticas de seguridad propios del desarrollo Web moderno.

Objetivos específicos

- Comprender la arquitectura cliente/servidor en n capas y las características distintivas de las arquitecturas de aplicaciones en la Web.
- Conocer y aplicar el protocolo HTTP y el estilo arquitectónico REST en el diseño de Web Services.
- Programar en el entorno Web utilizando HTML, lenguajes de script y librerías/frameworks de front-end.
- Programar en JavaScript tanto en el navegador como en el entorno de ejecución Node.js, incluyendo lenguajes de script asincrónicos.
- Desarrollar interfaces de usuario dinámicas mediante componentes funcionales de React sobre un proyecto inicializado con Vite, aplicando criterios de diseño de interfaces de usuario centrado en la persona usuaria.
- Diseñar y construir aplicaciones y Web Services con Node.js, incluyendo la persistencia de datos desde el backend.
- Aplicar mecanismos básicos de autenticación de usuarios en aplicaciones Web, como base para la profundización de la seguridad y confidencialidad en asignaturas posteriores del plan.
- Distinguir la Web sintáctica de la Web semántica y aplicar buenas prácticas de estándares del desarrollo Web.
- Integrar front-end y back-end en una aplicación Web completa (full-stack) de mediana complejidad y desplegarla en un entorno accesible.
- Analizar comparativamente distintas plataformas y tecnologías de desarrollo Web disponibles en el mercado, y discutir nuevas tendencias tecnológicas.

VI - Contenidos

Contenidos mínimos

Programación en entorno Web: HTML. Lenguajes de Script. Librerías de desarrollo web para front-end. Frameworks. Programación en la Web con lenguajes de backend y acceso a base de datos. Lenguajes de Script asincrónicos. Web Services. Análisis comparativos de plataformas de desarrollo Web disponibles en el mercado. Estudio y discusión de los distintos Estándares utilizados en el ámbito del desarrollo de aplicaciones Web. Introducción a las características distintivas de las Arquitecturas de aplicaciones en la Web. Profundización del estudio del esquema Cliente/Servidor “n capas”. Seguridad y Confidencialidad en aplicaciones Web. Web sintáctica y Web semántica. Características de Diseños de Interfaces de Usuarios. Nuevas Tendencias de Tecnologías utilizadas en el mercado.

Laboratorio: Uso de entornos de desarrollo para proyectos Web de mediana complejidad utilizando los lenguajes vistos en la materia.

Abordaje y evaluación de ejes

A continuación, se detallan los ejes de la formación trabajados en la materia indicando cómo se aborda y cómo se evalúa.

Identificación, formulación y resolución de problemas de informática:

Se aborda a través del análisis de casos de estudio que los/las estudiantes deben traducir en requerimientos concretos de una aplicación Web (qué entidades manejar, qué operaciones ofrecer, qué restricciones cumplir), tanto en los trabajos prácticos individuales como en el proyecto integrador. Se evalúa de manera continua en la corrección de cada TP, verificando que la solución implementada responda efectivamente al problema planteado, y de forma integradora en la defensa oral del proyecto final.

Concepción, diseño y desarrollo de proyectos de informática:

La materia organiza su cursada como un desarrollo incremental, donde cada TP constituye un componente que luego se integra en el proyecto final full-stack. Se evalúa en la aprobación de cada entrega parcial y, de manera integral, en la evaluación del proyecto integrador, donde se pondera la coherencia del diseño de principio a fin.

Utilización de técnicas y herramientas de aplicación en la informática:

Eje central y transversal a toda la cursada: se trabaja mediante el uso cotidiano de Node.js, React, control de versiones, entornos de desarrollo y herramientas de depuración, que los/las estudiantes deben incorporar como parte de su práctica habitual desde el primer trabajo práctico. Su evaluación es continua: se observa en cada entrega el uso correcto y autónomo de estas herramientas.

Generación de desarrollos tecnológicos y/o innovaciones tecnológicas:

Se aplica mediante la construcción, por parte de cada grupo, de una aplicación Web propia a partir de un caso de estudio, en lugar de trabajar sobre ejercicios cerrados o predefinidos. Se evalúa en el proyecto integrador, valorando tanto la funcionalidad lograda como las decisiones técnicas tomadas para resolver el caso propuesto.

Fundamentos para el desempeño en equipos de trabajo:

Se trabaja específicamente en el proyecto integrador, que se desarrolla en grupos reducidos, exigiendo organización interna, distribución de tareas y trabajo colaborativo sobre un mismo repositorio. Se evalúa a través de la defensa oral grupal del proyecto, donde se indaga sobre la participación y comprensión individual de cada integrante respecto del trabajo conjunto.

Fundamentos para la comunicación efectiva:

Se aborda mediante la exigencia de documentación técnica clara en cada entrega y, especialmente, en la defensa oral del proyecto integrador, donde los/las estudiantes deben explicar y justificar las decisiones adoptadas. Se evalúa en esa misma instancia oral, considerando la claridad de la exposición y la capacidad de responder preguntas técnicas.

Fundamentos para la acción ética y responsable:

Se trabaja de manera acotada, en la implementación de autenticación de usuarios mediante tokens (JWT) dentro de la Unidad 7, discutiendo la responsabilidad del desarrollador en la protección de rutas y credenciales; el estudio en profundidad de vulnerabilidades y buenas prácticas de seguridad se aborda en asignaturas posteriores del plan. Se evalúa en el trabajo práctico correspondiente a persistencia y autenticación.

Fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local:

Se aborda de forma introductoria, en el estudio de accesibilidad web y de los principios de diseño de interfaces centrado en la persona usuaria, discutiendo cómo las decisiones de diseño afectan la inclusión de distintos tipos de usuarios. Se evalúa como criterio de corrección dentro del trabajo práctico de maquetado e interfaz, sin constituir una instancia de evaluación independiente.

Fundamentos para el aprendizaje continuo:

Es un eje central en una materia cuyo dominio tecnológico cambia constantemente: se trabaja explícitamente en el análisis comparativo de plataformas y en la discusión de nuevas tendencias del mercado, promoviendo que los/las estudiantes indaguen por su cuenta más allá de las herramientas vistas en clase. Se evalúa en la realización de trabajos y en la instancia de coloquio oral, donde se puede indagar sobre alternativas tecnológicas no cubiertas en profundidad durante las clases.

Unidades**Unidad 1: Introducción a las Arquitecturas de Aplicaciones Web**

Evolución de las aplicaciones Web: de los sitios estáticos a las aplicaciones distribuidas. Esquema cliente/servidor. Introducción al modelo “n capas”: presentación, lógica de negocio y datos. Aplicación del protocolo HTTP: métodos, códigos de estado, cabeceras, cookies y sesiones. Estándares del desarrollo Web (W3C, WHATWG, IETF, ECMA). Web sintáctica y Web semántica. Panorama de plataformas y tecnologías de desarrollo Web disponibles en el mercado.

Unidad 2: Programación en el Entorno Web: HTML y CSS

Características de diseño de interfaces de usuario (UI/UX): usabilidad, accesibilidad y diseño centrado en la persona usuaria. HTML5: estructura semántica de documentos, formularios y elementos multimedia. Hojas de Estilo en Cascada (CSS): selectores, modelo de caja y diseño responsivo

Unidad 3: Fundamentos de JavaScript y Lenguajes de Script Asincrónicos

Sintaxis, tipos de datos, funciones, closures y clases (ES6+). Manipulación del DOM y manejo de eventos. Lenguajes de script asincrónicos: callbacks, Promises y async/await. Módulos de JavaScript (ES Modules) y gestión de paquetes con npm.

Unidad 4: Librerías y Frameworks de Desarrollo Front-End: React

Panorama de librerías y frameworks de front-end (comparación conceptual: jQuery como librería; React, Angular y Vue.js como frameworks). Inicialización y estructura de un proyecto front-end con Vite. React: componentes funcionales, props y JSX. Hooks básicos (useState, useEffect) para manejo de estado local y efectos. Enrutamiento en el cliente (React Router) y consumo de Web Services REST desde el cliente (fetch/axios).

Unidad 5: Estilo Arquitectónico REST y Web Services

Principios REST: recursos, verbos HTTP, representaciones. Diseño de APIs REST: convenciones, versionado, formato JSON. Introducción a otros estilos de Web Services. Documentación de APIs.

Unidad 6: Node.js y Desarrollo de Servidores de Aplicaciones

Node.js: modelo de ejecución, event loop y módulos. Express: enrutamiento, middlewares y manejo centralizado de errores. Diseño de servidores de aplicaciones Web y Web Services con Node.js/Express. Uso de asistentes de código basados en IA (LLMs/extensiones del editor) como herramienta de apoyo en el desarrollo backend.

Unidad 7: Programación con Lenguajes de Backend, Acceso a Bases de Datos y Autenticación

Conexión de Node.js/Express a bases de datos relacionales y/o no relacionales. Operaciones a través de Web Services REST. Autenticación de usuarios mediante tokens (JWT): emisión, verificación y protección de rutas con middlewares de autenticación.

Unidad 8: Integración Full-Stack, Despliegue, Nuevas Tendencias y Análisis Comparativo

Integración de front-end (React) y back-end (Node.js/Express) en una aplicación completa. Despliegue de aplicaciones Web. Análisis comparativo de plataformas de desarrollo Web. Nuevas tendencias tecnológicas del mercado.

VII - Plan de Trabajos Prácticos

Los trabajos prácticos se desarrollan íntegramente en las clases de laboratorio, sobre un stack tecnológico HTML5/CSS3/JavaScript/Node.js/React, articulándose de manera incremental hacia un proyecto integrador final full-stack de mediana complejidad. A continuación, se detalla el cronograma de trabajos prácticos, indicando para cada uno las semanas asignadas, su descripción y la unidad temática con la que se relaciona.

TP N.º 1: (Unidad 1 y 2)

Maquetado de una interfaz Web estática con HTML5 y CSS3, aplicando principios de diseño de interfaz de usuario y diseño responsivo.

TP N.º 2: (Unidad 3)

Programación en JavaScript: manipulación del DOM, manejo de eventos y consumo de un Web Service público mediante Promises/fetch. Consumo de servicios de IA (LLMs) mediante Web Services REST, como caso de aplicación de Promises y async/await

TP N.º 3: (Unidad 4)

Desarrollo de una interfaz de usuario dinámica (SPA) con React sobre un proyecto inicializado con Vite: componentes funcionales, hooks básicos y enrutamiento en el cliente.

TP N.º 4: (Unidad 5 y 6)

Diseño y construcción de un Web Service REST con Node.js y Express, incluyendo middlewares y manejo de errores.

TP N.º 5: (Unidad 7)

Persistencia de datos desde el backend: integración del Web Service desarrollado con una base de datos, incluyendo autenticación de usuarios mediante tokens (JWT) y protección de rutas.

Proyecto Integrador:

Desarrollo grupal de una aplicación Web full-stack de mediana complejidad: SPA en React consumiendo un Web Service propio, autenticación de usuarios, persistencia en base de datos y despliegue en un entorno accesible. Este trabajo se realiza desde la unidad 1, en entregas incrementales aplicando los contenidos de la unidad 8. Finalmente se realiza la defensa oral.

VIII - Regimen de Aprobación

Condiciones para obtener la regularidad

- Aprobar el seguimiento continuo de los trabajos prácticos bajo la modalidad ABP (Aprendizaje Basado en Proyectos) con nota mayor o igual a 6 (seis).
- Aprobar las entregas parciales del proyecto integrador con nota mayor o igual a 6 (seis).
- Contar con al menos el 70% de asistencia a clases.
- Aprobar el proyecto integrador y la defensa oral del mismo con nota mayor o igual a 6 (seis).

Condiciones para obtener la promoción de la materia

Regularizar la materia con las siguientes condiciones adicionales:

- Aprobar el seguimiento continuo de los trabajos prácticos bajo la modalidad ABP (Aprendizaje Basado en Proyectos) con nota mayor o igual a 7 (siete).
- Contar con al menos el 80% de asistencia a clases.
- Aprobar las entregas parciales del proyecto integrador con nota mayor o igual a 7 (siete).
- Aprobar el proyecto integrador y la defensa oral del mismo con nota mayor o igual a 7 (siete).
- Aprobar un coloquio oral con nota mayor o igual a 7 (siete).

En caso de no promocionar, el estudiante deberá rendir un examen final oral.

Alumnos libres: no se admiten exámenes libres debido al carácter eminentemente práctico de la materia.

Al tratarse de una modalidad de evaluación continua (ABP), se otorgan dos instancias de recuperación para las entregas no aprobadas del seguimiento continuo y del proyecto integrador, conforme lo establece la reglamentación actual

IX - Bibliografía Básica

- [1] Mozilla Developer Network (MDN Web Docs). (2025). Web Development Documentation. <https://developer.mozilla.org>
- [2] Haverbeke, M. (2024). Eloquent JavaScript (4.ª ed.). No Starch Press. <https://eloquentjavascript.net>
- [3] Casciaro, M., & Mammino, L. (2020). Node.js Design Patterns (3.ª ed.). Packt Publishing.
- [4] Goldberg, Y. et al. Node.js Best Practices. <https://github.com/goldbergonyi/nodebestpractices>
- [5] React Core Team. (2025). Documentación oficial de React. <https://react.dev>
- [6] Vite Team. (2025). Documentación oficial de Vite. <https://vite.dev>
- [7] Expressjs.com Team. (2025). Documentación oficial de Express. <https://expressjs.com>
- [8] Richardson, L., & Amundsen, M. (2013). RESTful Web APIs. O'Reilly Media.
- [9] Microsoft REST API Guidelines. <https://github.com/microsoft/api-guidelines>
- [10] W3C. (2025). HTML: The Living Standard y pautas de accesibilidad (WCAG). <https://www.w3.org>

X - Bibliografía Complementaria

- [1] OWASP Foundation. (2025). OWASP Top 10 Web Application Security Risks. <https://owasp.org/www-project-top-ten/>
- [2] Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures (tesis doctoral). University of California, Irvine. <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [3] Node.js Foundation. (2025). Node.js Documentation. <https://nodejs.org/docs>
- [4] Vue.js Core Team. (2025). Guía oficial de Vue.js (referencia comparativa). <https://vuejs.org/guide/>
- [5] WHATWG. (2025). HTML Living Standard. <https://html.spec.whatwg.org>
- [6] Simpson, K. (2020). You Don't Know JS Yet (2.ª ed., acceso público). <https://github.com/getify/You-Dont-Know-JS>

[7] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. Scientific American, 284(5), 34-43.
<https://www.scientificamerican.com/article/the-semantic-web/>

XI - Resumen de Objetivos

Formar a los/las estudiantes en el diseño, desarrollo, integración, seguridad y despliegue de aplicaciones Web full-stack de mediana complejidad, aplicando buenas prácticas propias del dominio Web sobre un stack tecnológico moderno (HTML, CSS, JavaScript, Node.js, React).

XII - Resumen del Programa

El programa comprende las siguientes unidades: Arquitecturas de aplicaciones Web y esquema cliente/servidor n capas, Programación en el entorno Web con HTML y CSS. Fundamentos de JavaScript y lenguajes de script asíncronos. Librerías y frameworks de front-end (React con Vite). Estilo arquitectónico REST y Web Services. Node.js y desarrollo de servidores de aplicaciones. Programación con lenguajes de backend, acceso a bases de datos y autenticación de usuarios. Integración full-stack, despliegue, nuevas tendencias y análisis comparativo de plataformas.

XIII - Imprevistos

Ante circunstancias extraordinarias que impidan el normal desarrollo de alguna clase teórica o de laboratorio (feriados, suspensión de actividades académicas u otras causas de fuerza mayor), la cátedra reprogramará la actividad correspondiente e informará oportunamente a los estudiantes a través de los canales de comunicación habituales.

XIV - Otros

Datos de contacto de la cátedra: Oficina 1, 1er piso Bloque 2
Correo electrónico: dweb.unsl@gmail.com
Classroom Laboratorio de Tecnologías

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA

	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	