



---

**SEMINARIO DE LENGUAJES  
(OPCION JavaScript)**

**Año 2026**

**Carrera/ Plan:**

Licenciatura en Informática  
Plan 2015 // Plan 2021  
Licenciatura en Sistemas  
Plan 2015 // Plan 2021  
Analista Programador Universitario  
Plan 2015 // Plan 2021  
Analista en Tecnologías de la  
Información y la Comunicación Plan  
2017 // Plan 2021

**Año:** 2°

**Régimen de Cursada:** Semestral

**Carácter:** Obligatoria **Correlativas:**

Taller de Programación

**Profesor:** Paola Amadeo

**Hs. semanales:** 6 hs.

---

**FUNDAMENTACIÓN**

Dentro del marco de la materia genérica Seminario de Lenguajes, esta opción introduce el lenguaje JavaScript. A partir de 1995, la penetración de Internet en todos los ámbitos generó aplicaciones cada vez más complejas y con numerosas capas para cumplir con los requisitos del paradigma centrado en la red. JavaScript es uno de los lenguajes que surgió para dar respuesta a este paradigma siendo hoy uno de los más populares lenguajes de scripts usado para construir aplicaciones front-end, back-end y mobile. Aprender JS en esta instancia de la carrera brinda bases sólidas para materias correlativas así como también para el futuro profesional del estudiante. La metodología aprender haciendo es fundamental en estos cursos, donde el mundo digital requiere competencias de trabajo en equipo, gestión de plataformas para el desarrollo de proyectos, fomenta la motivación y la creatividad.

**OBJETIVOS GENERALES**

Profundizar los conocimientos obtenidos por el alumno en los primeros cursos vinculados con Algoritmos y Programación, permitiéndole desarrollar un estudio teórico-práctico de algún lenguaje de programación (el lenguaje puede variar con los cambios tecnológicos), poniendo énfasis en el análisis formal de las características del lenguaje y su comparación con los que el alumno conociera a ese momento (típicamente Pascal).

**COMPETENCIAS**

- CGS2- Comunicarse con efectividad en forma oral y escrita.



- CGS4- Aprender en forma continua y autónoma, con capacidad de planificar este aprendizaje.
- CGS6- Capacidad para interpretar la evolución de la Informática con una visión de las tendencias tecnológicas futuras.
- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT5- Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Informática.
- LI- CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LS- CE1 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.

## **RESULTADOS DE APRENDIZAJE**

1.3. Describir los avances informáticos actuales e históricos y demostrar cierta visión sobre tendencias y avances futuros (Básico)

3.1. Definir y diseñar hardware/software informático/de red que cumpla con los requisitos establecidos (Básico)

3.3. Elegir y utilizar modelos de proceso adecuados, entornos de programación y técnicas de gestión de datos con respecto a proyectos que impliquen aplicaciones tradicionales, así como aplicaciones emergentes (Básico)

3.4. Describir y explicar el diseño de sistemas e interfaces para interacción persona-ordenador y ordenador-ordenador (Básico)

3.5. Aplicar las correspondientes competencias prácticas y de programación en la creación de programas informáticos y/u otros dispositivos informáticos (Adecuado)

6.1. Organizar su propio trabajo de manera independiente demostrando iniciativa y ejerciendo responsabilidad personal (Básico)

6.3. Planificar su propio proceso de aprendizaje autodidacta y mejorar su rendimiento personal como base de una formación y un desarrollo personal continuos (Básico)

## **CONTENIDOS MINIMOS**

Estudio de un lenguaje de programación en el que se desarrollen aplicaciones concretas. En lo posible la oferta de lenguajes será variable y actualizada con el cambio tecnológico



## **PROGRAMA**

Unidad 1: La WWW. HTML, CSS y JS

Unidad 2: Componentes básicos del lenguaje.

Unidad 3: Estructurando datos. Modularización. Funciones

Unidad 4: Eventos y formularios. Variables. API Fetch

Unidad 5: Librerías Client-side.

Unidad 6: Server-side JS.

Taller de ética profesional.

## **BIBLIOGRAFÍA**

- JavaScript. <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Standard ECMA-262  
<https://www.ecma-international.org/publications/standards/Ecma-262.htm>
- JavaScript: The Definitive Guide, 7th Edition. D. Falagan (2020) . Ed O`Reilly  
<https://www.oreilly.com/library/view/javascript-the-definitive/9781491952016/>
- Referencia de JavaScript.  
<https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia>
- HTML Living standard <https://html.spec.whatwg.org/multipage/>
- HTML 5.2 W3C Recommendation <https://www.w3.org/TR/html52/>
- HTML5: Entiende el cambio, aproveche su potencial. De Luca, Damián. (2011). Ed. Andina  
<http://catalogo.info.unlp.edu.ar/meran/opac-detail.pl?id1=5738#.X70rOrN7IPY>
- Pro HTML5 accessibility: building an inclusive web. O Connor, Joshue (2012). Ed. Apress.  
<http://catalogo.info.unlp.edu.ar/meran/opac-detail.pl?id1=7241#.X70tQLN7IPY>
- W3C. Cascading Style Sheets. <https://www.w3.org/Style/CSS/>
- Visual Studio Code (<https://code.visualstudio.com/download>)–Web Storm  
(<https://www.jetbrains.com/webstorm/>)
- JavaScript Tutorial. <https://www.w3schools.com/js/default.asp>
- NodeJS <https://nodejs.org/es/>
- ReactJS <https://es.react.dev/>



- Entendiendo los frameworks de JavaScript del lado del cliente  
[https://developer.mozilla.org/es/docs/Learn/Herramientas\\_y\\_pruebas/Lado-del-cliente\\_JavaScript\\_frameworks](https://developer.mozilla.org/es/docs/Learn/Herramientas_y_pruebas/Lado-del-cliente_JavaScript_frameworks)
- Using GitHub Copilot with JavaScript.  
<https://learn.microsoft.com/en-us/training/modules/introduction-copilot-javascript/>
- [JavaScript Roadmap Step by step guide to learn JavaScript in 2026](#)
- [Learn to become a TypeScript developer](#)

## **METODOLOGÍA DE ENSEÑANZA**

La metodología es del tipo taller, con clases teóricas donde se desarrollan los aspectos conceptuales del lenguaje, que se dictan utilizando videos y presentaciones del tipo slideshow. Se incluyen ejemplos de código para resolver distintas situaciones problemáticas y animaciones que permitan ilustrar los conceptos más complejos durante la clase. Estas clases no son obligatorias y se planearán actividades para promover el intercambio entre docentes y estudiantes del curso durante el dictado de la misma.

Las clases prácticas se realizan en aulas con PC de la Facultad, basada en proyectos, los estudiantes plantean sus dudas, trabajan con los ayudantes y se corrigen entre ellos, acompañándolos durante todo el proceso. Las prácticas incluyen ejercicios a entregar y tendrán una calificación conceptual. Los ejercicios planteados buscan fomentar el aprendizaje activo a través de la búsqueda de bibliografía actualizada sobre el tema, uso crítico de las inteligencias artificiales, planificación de tareas a realizar y actividades de colaboración grupal. Además, incluyen una instancia de coloquio donde el docente a cargo del grupo realiza distintas preguntas sobre la temática abordada.

Las entregas se realizan a través de la plataforma de aprendizaje virtual <https://catedras.info.unlp.edu.ar> o el repositorio de la cátedra <https://gitlab.catedras.linti.unlp.edu.ar/> En los coloquios, el estudiante expone la tarea realizada en forma individual y el docente evalúa no sólo los conocimientos sino la claridad de la presentación, su organización y la forma de expresión. Esto se refleja en rúbricas que conforman documentación de la evaluación del coloquio.

Esta metodología se complementa con la plataforma virtual Moodle y el repositorio Gitlab instalado en el Linti de la Facultad. Se incluyen autoevaluaciones de carácter optativo para reforzar temas teóricos. Cada ayudante tiene a cargo un grupo de alumnos, y será su función seguirlo en el desenvolvimiento de la cursada, intentando identificar los puntos más problemáticos para poder resolverlos en forma rápida y no provoque el abandono de la cursada.

Se permite el uso de herramientas de inteligencia artificial como apoyo al proceso de aprendizaje para la generación de código, el diseño y el maquetado, entre otras actividades. El equipo docente lleva a cabo instancias de revisión de código y coloquios en las que las y los estudiantes deben fundamentar las decisiones adoptadas y acreditar la comprensión del código desarrollado.

Se utilizará también una herramienta para el seguimiento de la cátedra, de la asistencia de los alumnos a las prácticas y del aprovechamiento de cada práctica.

## **EVALUACIÓN**

La **aprobación de la cursada** estará dada por un trabajo final integrador. Tendrá la forma de “desafíos” que deben convertirse en “ideas proyecto” y posteriormente en potenciales desarrollos del estudiante. Se trata de que alumno logre abstraer una serie de pasos que



respondan a una metodología clásica de investigación y lo ayuden a formarse en esta competencia:

- Búsqueda de bibliografía actualizada sobre el tema.
- Abstracción del desafío/problema como una “idea proyecto a resolver”.
- Expresión sintética de la especificación del proyecto, con recursos humanos requeridos y plan de tareas.
- Implementación y defensa oral/escrita de la solución al desafío.

La cátedra acompaña el proceso del alumno, para consolidar sus habilidades para esta competencia.

La evaluación de estas competencias se refleja en una planilla detallada, donde se indica la capacidad del alumno para desarrollar su aprendizaje y la formulación de la solución al desafío en forma autónoma.

En la cátedra se pone énfasis en la capacidad del alumno para conocer técnicas y herramientas de aplicación en Informática (en lo posible siguiendo las tendencias marcadas por el cambio tecnológico) y en la aplicación efectiva de las mismas. La cátedra acompaña el proceso con materiales para que el alumno estudie casos y valore la selección y empleo eficiente de herramientas y técnicas determinadas para cada problema. La evaluación de esta competencia forma parte de las evaluaciones de trabajos prácticos.

La **aprobación de la materia** estará dada por la aprobación de la cursada y un trabajo final. La asistencia, la participación en las clases teóricas y la participación en actividades de Extensión de la Facultad, debidamente acreditadas, aportará a la calificación final.

## **CRONOGRAMA DE CLASES Y EVALUACIONES**

**Fecha de comienzo: semana 23/03/2026**

**Fecha de comienzo de prácticas: semana 30/03/2026.** Martes de 17:00 a 20:00 hs y  
Miércoles 16:00 a 18:00 hs

25/3	Clase 1 - Fundamentos del Desarrollo Web: Explorar el funcionamiento de la web a través del modelo cliente-servidor e indagar en la tríada esencial del desarrollo web: HTML, CSS y JavaScript.
1 / 4	Clase 2 - HTML y CSS. Grid, Flexbox. Responsive design. Accesibilidad Web WCAG.
8/4	Clase 3 - Profundización en JavaScript y Formularios HTML: Lógica, Estructuras y Datos. Sistema de tipos. Comparación con Pascal.
15/4	Clase 4 - JavaScript y HTML: interacción de usuarios y gestión de eventos. Modelos de ejecución: event loop, call stack y microtasks vs macrotasks.
22/4	Clase 5 - Introducción a Objetos en JavaScript. Prototipos vs Clases. Funciones (declarativas, expresiones, arrow functions), scope (var, let, const), hoisting, closures. "this"
29/4	Clase 6 – Arreglos en JS
6/5	Clase 7 - Comunicación Asíncrona en la Web: HTTP, API REST, Fetch y Promesas. API de IA.
13/5	Clase 8 - Inmersión en Node.js: Fundamentos del Desarrollo Server-side
20/5	Clase 9 - Profundizando en <a href="https://nodejs.org/">Node.js</a> . Express. Routing. Middleware. Manejo de errores.
27/5	Clase 10 - Node.js Continuación. Deno y Bun.
3/6	Presentación del Trabajo Final (Semana 1)
10/6	Desarrollo Trabajo Final (Semana 2)
17/6	Desarrollo Trabajo Final (Semana 3)
24/6	Desarrollo Trabajo Final (Semana 4)
1/7	Desarrollo Trabajo Final (Semana 5)
8/7	Correcciones del Trabajo Final (Semana 6)
15/7	Luego de la entrega final se realizará una puesta en común de todos los estudiantes de su trabajo final, haciendo hincapié en los problemas que tuvieron en el desarrollo del mismo y la forma de solucionarlos, así como también haciendo un balance de su desempeño

20/07/2026 al 2/08/2026 RECESO

8/8 FIN DEL CUATRIMESTRE

**Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos):**

[seminariojs@info.unlp.edu.ar](mailto:seminariojs@info.unlp.edu.ar). [Catedras.info](https://catedras.info) [Plataforma virtual](#)

**Firmas del profesor responsable:**