

PROYECTO DE SOFTWARE**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***Año:** 3°**Régimen de Cursada:** Semestral**Carácter:** Obligatoria**Correlativas:** Diseño de Bases de Datos- Ingeniería de Software 1- Algoritmos y Estructuras de Datos - Seminario de Lenguajes - Taller de Lecto-comprensión y Traducción en Inglés**Profesor:** *Claudia Banchoff Tzancoff, Diego Vilches Antao, Matías Pagano***Hs. semanales teoría:** 3 hs.**Hs. semanales práctica:** 3 hs.**FUNDAMENTACIÓN**

La asignatura consolida la formación experimental y profesionalizante del estudiante ubicándolo en un entorno de trabajo similar al real y cotidiano. Se trabaja con herramientas y metodologías que se utilizan en entornos profesionales, introduciendo a los y las estudiantes en estos contextos de trabajo.

Por lo general, las aplicaciones desarrolladas surgen de pedidos de colaboración realizados por la comunidad con lo cual los y las estudiantes también adquieren práctica en actividades extensionistas.

OBJETIVOS GENERALES

El objetivo general de esta asignatura es realizar un desarrollo web que signifique para los estudiantes una aplicación concreta de los conocimientos adquiridos, consolidando la formación experimental.

Se promoverán las Instancias Supervisadas de Formación en la Práctica Profesional (ISFPP) haciendo hincapié en trabajos de relevancia y pertinencia social.

Durante el desarrollo de las distintas actividades propuestas se promueve que los estudiantes:

- conozcan, respeten y utilicen los distintos estándares definidos;
- utilicen herramientas de versionado de código;
- aprendan a documentar correctamente un desarrollo;
- adquieran las habilidades del trabajo en equipo;
- puedan presentar y exponer su trabajo.

RESULTADOS DE APRENDIZAJE

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

1.4. Aplicar e integrar conocimientos de otras disciplinas informáticas como apoyo al estudio de la propia área de especialidad (o áreas de especialidad) (Adecuado).

1.5. Demostrar sensibilización ante la necesidad de contar con amplios conocimientos a la hora de crear aplicaciones informáticas en otras áreas temáticas (Adecuado).

2.2. Describir un determinado problema y su solución a varios niveles de abstracción (Adecuado).

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

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 (Adecuado).

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

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

4.1. Demostrar concienciación sobre la necesidad de tener una conducta ética y profesional de primer nivel y conocimientos de los códigos de conducta profesionales (Adecuado).

4.1. Demostrar concienciación sobre la necesidad de tener una conducta ética y profesional de primer nivel y conocimientos de los códigos de conducta profesionales (Adecuado).

4.2. Explicar la forma en la que el contexto comercial, industrial, económico y social afecta la práctica de la informática (Básico).

4.2. Explicar la forma en la que el contexto comercial, industrial, económico y social afecta la práctica de la informática (Adecuado).

4.4. Explicar la importancia de la confidencialidad de la información y cuestiones relativas a la seguridad con respecto al diseño, desarrollo, mantenimiento, supervisión y uso de sistemas informáticos (Básico).

4.4. Explicar la importancia de la confidencialidad de la información y cuestiones relativas a la seguridad con respecto al diseño, desarrollo, mantenimiento, supervisión y uso de sistemas informáticos (Básico).

5.2. Describir y explicar las técnicas de gestión correspondientes al diseño, implementación, análisis, uso y mantenimiento de sistemas informáticos, incluyendo gestión de proyectos, de configuración y de cambios, así como las técnicas de automatización correspondientes (Adecuado).

5.3. Identificar los riesgos, incluyendo riesgos de seguridad, laborales, medioambientales y comerciales y llevar a cabo una evaluación de riesgos, reducción de riesgos y técnicas de gestión de riesgos (Adecuado)

5.3. Identificar los riesgos, incluyendo riesgos de seguridad, laborales, medioambientales y comerciales y llevar a cabo una evaluación de riesgos, reducción de riesgos y técnicas de gestión de riesgos (Adecuado).

5.4. Realizar investigaciones bibliográficas y evaluaciones utilizando bases de datos y otras fuentes de información (Adecuado).

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

6.2. Comunicar mensajes de forma efectiva tanto oralmente como por medio de otros medios de comunicación ante distintas audiencias (Adecuado).

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

6.4. Identificar las distintas maneras de organizar equipos y los distintos roles dentro de dichos equipos (Adecuado).

6.5. Participar de manera efectiva en grupos de trabajo informático (Adecuado).

COMPETENCIAS

- CGS2- Comunicarse con efectividad en forma oral y escrita.
- CGS3- Actuar con ética, responsabilidad profesional y compromiso social y ambiental, considerando el impacto económico, social y ambiental de su actividad en el contexto local, regional y global
- 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.
- CGT10- Capacidad para realizar investigaciones bibliográficas y de diferentes fuentes de información a fin de obtener conocimiento actualizado en temas de la disciplina.

- CGT2- Concebir, diseñar y desarrollar proyectos de Informática.
- CGT3- Gestionar, planificar, ejecutar y controlar proyectos de Informática
- CGT6 – Capacidad para identificar y gestionar los riesgos en Informática (ambientales, laborales, de seguridad y económicos).
- CGT8 Capacidad de interpretación y resolución de problemas multidisciplinares, desde los conocimientos de la disciplina informática.
- CGT9 Capacidad de organización de equipos de trabajo en proyectos de Informática, definiendo los roles en los mismos.
- LI- CE1– Planificar, dirigir, realizar y/o evaluar proyectos de especificación, diseño, implementación, verificación, validación, puesta a punto, mantenimiento y actualización para arquitecturas de sistemas de procesamiento de datos, con capacidad de incorporar aspectos emergentes del cambio tecnológico.
- 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 interfaces 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 interfaces humano computador y computador-computador.
- LS- CE9– Analizar y evaluar proyectos de especificación, diseño, implementación, puesta a punto, mantenimiento y actualización de sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico.

CONTENIDOS MÍNIMOS (de acuerdo al Plan de Estudios)

Según el enfoque de los proyectos que se desarrollen, el estudiante recibirá clases teóricas de aspectos avanzados de Ingeniería de Software, Algoritmos/Lenguajes y/o Bases de Datos. Estos conceptos teóricos serán acompañados por una intensa tarea de desarrollo (individual o en equipos) siguiendo todas las etapas conceptuales de un proyecto de software, desde su especificación hasta su verificación y validación.

PROGRAMA ANALÍTICO

Unidad I. Aspectos de software y recursos libres. Conceptos básicos de una arquitectura web clásica. Herramientas de versionado de código y entornos virtuales.

Unidad II. Lenguaje HTML. Estructura de un documento HTML. Componentes. Hojas de Estilo. Validadores. Accesibilidad web: pautas e iniciativas; recomendaciones. DOM (Document Object Model) Especificación y uso.

Unidad III. JavaScript: el lenguaje y sus componentes. Alcances y limitaciones. Eventos. Frameworks para desarrollo frontend: Vue.js.

Unidad IV. Codificación en el servidor. Instalación y configuración de un servidor para aplicaciones web. Uso de Python para el desarrollo de aplicaciones web. Uso de ORM para acceso a bases de datos. Testing.

Unidad V. Arquitectura MVC (Model View Controller). Desarrollo de API (*Application Programming Interface*). Arquitectura RESTful. Frameworks de desarrollo web en Python. Concepto y análisis de las

alternativas actuales: Flask.

Unidad VI. Integración con APIS conocidas como OAuth, OpenStreetMap entre otras. Formatos de intercambio: XML, JSON.

Unidad VII. Aspectos básicos de seguridad web. Vulnerabilidades típicas. La iniciativa OWASP.

BIBLIOGRAFÍA

- Pro Git. Scott Chacon & Ben Straub. Disponible en: <https://git-scm.com/book/en/v2>.
- CSS practica/ CSS Instant result: Cascading Style Sheets For Web. York, Richard.
- Introducción a Javascript. Javier Eguiluz. Disponible en <http://librosweb.es/libro/javascript/>.
- Introducción a CSS. Javier Eguiluz. Disponible en <http://librosweb.es/libro/css/>.
- La seguridad en aplicaciones web – Top Ten https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.
- RESTful Web Services. Leonard Richardson & Sam Rub, disponible en <http://restfulwebapis.org/rws.html>.
- Documentación de Python en línea: <https://www.python.org/>.
- Documentación Flask: <https://flask.palletsprojects.com/en/1.1.x/>.
- Guía oficial Vue.js: <https://es.vuejs.org/v2/guide>.

METODOLOGÍA DE ENSEÑANZA

La asignatura consolida la formación experimental y profesionalizante de los estudiantes ubicándolos en un entorno de trabajo similar al real y cotidiano.

Al comenzar la cursada, se realiza una evaluación diagnóstica sobre los conocimientos iniciales de los estudiantes, a través de una encuesta en línea. En esta encuesta, además se releva otra información de interés como ser su situación laboral, aspectos de conectividad y su situación respecto de las restantes asignaturas: cantidad de asignaturas adeudadas, cantidad de materias que cursan en simultáneo, entre otras.

La teoría y la práctica, dictadas semanalmente, se encuentran estrechamente vinculadas. La teoría trabaja lineamientos conceptuales que se aplican en los desarrollos prácticos.

Se organizan actividades planificadas para los estudiantes, en las que se proponen desafíos que deben convertirse en ideas proyecto y posteriormente en potenciales desarrollos. Se trata que el estudiante logre abstraer una serie de pasos que respondan a una metodología clásica de investigación y lo ayuden a aprender en forma continua y autónoma, con capacidad de planificar este aprendizaje:

- búsqueda de bibliografía actualizada sobre el tema;
- uso crítico de herramientas de inteligencia artificial;
- discusión de alternativas tecnológicas para resolver el tema propuesto;
- abstracción del desafío y/o 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 y escrita, de la solución al desafío.

Se pone énfasis en el proceso de identificación de problemas del mundo real, especificación de los mismos como problemas resolubles desde la Informática, formulación de proyectos para su solución y en el análisis y diseño de las soluciones en el marco del proyecto formulado. Se propone la realización de un trabajo integrador de relevancia y pertinencia social, que surge generalmente de demandas provenientes de la comunidad, por lo general, a través de la Secretaría de Extensión de la Facultad. Este trabajo, realizado en etapas con un seguimiento exhaustivo por parte del equipo docente, se desarrolla en grupos

de tres o cuatro estudiantes abordando distintos roles en el marco de un proyecto de software. En el seguimiento del proyecto se pone énfasis en la evaluación del trabajo colaborativo y en el cumplimiento del rol de cada uno.

Los proyectos a realizar son multidisciplinarios, en donde se interactúa con profesionales y/o especialistas de otras disciplinas. Se trata de acompañar al estudiante en la interpretación del rol del Informático como articulador de soluciones en áreas de conocimiento muy diferentes que requieren participación de expertos extra-disciplinares.

Se pone énfasis en el proceso de identificación de riesgos asociados con proyectos que resuelven problemas del mundo real. En especial se trata de transmitir que el análisis de riesgos tiene diferentes ejes y puede resultar decisivo para valorar un proyecto o solución informática. La cátedra acompaña el aprendizaje del estudiante sobre esta competencia con materiales bibliográficos de casos de interés que resulten informativos y motivadores.

En las instancias prácticas los estudiantes resuelven las actividades propuestas utilizando herramientas de soporte y desarrollo típicas en la comunidad de software libre, entre las cuales se menciona: un servidor web, un motor de base de datos, un sistema de versionado de código, ambientes integrados de desarrollo (IDE - *Integrated Development Environment*), herramientas de *debugging* y validación.

Cada instancia planteada es supervisada por el equipo docente.

Además, se proponen actividades donde los estudiantes presentan sus ideas respecto de aspectos de responsabilidad profesional y ética personal y de la organización involucrada en el caso. Asimismo se discuten casos en los que el producto informático que se genera tiene un impacto económico/social o ambiental. En la evaluación de estos análisis de casos las respuestas del estudiante se reflejan en una planilla detallada compartida por los responsables de teoría y práctica para considerarlas en las evaluaciones de los estudiantes o darles una devolución específica que puede ser una recomendación o material de estudio complementario.

Se trabaja en la capacidad para interpretar la evolución de la Informática con una visión de las tendencias tecnológicas futuras, analizando tendencias sobre distintas tecnologías utilizadas y/o a utilizar y se desafía a los estudiantes a presentar posible evolución de la solución para ese tipo de problema y en qué podría mejorarse la solución/soluciones actuales.

Los estudiantes deben acompañar su trabajo con un informe final. Para el desarrollo de este informe existe una instancia específica donde se indican las pautas para su correcta redacción de un trabajo académico. Esto incluye aspectos de redacción, reglas de estilo y formas de incluir citas bibliográficas. También se dictan los lineamientos para la presentación oral de su trabajo.

Todos los estudiantes realizarán un 50% de las horas destinadas a trabajos prácticos como Instancias Supervisadas de Formación en la Práctica Profesional (ISFPP). La asignación de los grupos a las ISFPP será responsabilidad de la cátedra, con el acuerdo de los estudiantes y del organismo donde las realicen.

Se trabaja con los siguiente recursos:

- guías, diapositivas, videos, libros, tutoriales y especificaciones de estándares a utilizar;
- demostraciones de usos de herramientas con ejemplos en vivo;
- herramientas de versionado de código, servidor web, IDE de desarrollo, base de datos, API y frameworks de desarrollo;
- el EVEA Moodle como entorno de soporte: <https://catedras.linti.unlp.edu.ar/>

La interacción con los estudiantes se realiza a través de los mensajes directos o foros provistos por el

EVEA y a través de un sistema alternativo tal como Discord.

EVALUACIÓN

A modo de ejercitación y evaluación se plantean, a lo largo de la cursada, entregas de ejercicios que los estudiantes deben desarrollar y entregar en las prácticas y teorías.

Se entrega un trabajo en distintas etapas. En cada etapa se presenta una producción que los estudiantes defienden en forma de coloquio oral. Además se acompañan con actividades donde se indaga sobre los conceptos teóricos vistos en cada etapa del desarrollo. Esto es un requisito para la aprobación de la cursada. Estas entregas son de seguimiento y de evaluación con calificación.

Al finalizar la cursada existe una instancia de evaluación final donde los estudiantes exponen, en forma completa, el trabajo realizado. Esto se realiza en coloquios en los cuales deben exponer la tarea realizada en forma individual y donde el docente evalúa no sólo los conocimientos sino la claridad de la presentación, su organización y la forma de expresión.

Toda evaluación realizada a los estudiantes queda plasmada en planillas muy bien detallada, donde se indican los resultados de las diferentes evaluaciones realizada a los mismos: capacidad del estudiante para desarrollar su aprendizaje, claridad de las presentaciones realizadas, forma de organización y expresión en las diferentes instancias de evaluación oral, formulación de la solución de los diferentes desafíos en forma autónoma, entre otros.

Las ISFPP serán evaluadas para la aprobación de los trabajos prácticos de la asignatura y se tendrán en cuenta los informes de los estudiantes y la opinión de la institución en la que se realicen.

Respecto al uso de la inteligencia artificial, si bien se permite su utilización para la generación de código, se alienta un uso crítico de la misma. Es decir, se realizarán revisiones de código en la que el estudiante deberá justificar las decisiones tomadas en los trabajos realizados.

La materia se aprueba con el 70% de las evaluaciones prácticas y teóricas aprobadas y con el trabajo integrador aprobado.

CRONOGRAMA DE CLASES Y EVALUACIONES TENTATIVO

Clase	Fecha	Contenidos/Actividades
1	Semana del 17 de agosto	Presentación de la materia. Actividad diagnóstica grupal sobre conceptos básicos.
2	Semana del 24 de agosto	Git. Explicación de práctica inicial. Actividad grupal sobre estándares y validadores
3	Semana del 31 de agosto	Aspectos sobre software y recursos libre. Arquitectura web clásica. Conceptos básicos sobre el lenguaje HTML y hojas de estilos. Aspectos relacionados a la accesibilidad web. Programación básica en el cliente. Actividad práctica individual sobre git.
4	Semana del 7 de septiembre	Actividad individual sobre aspectos básicos teóricos vistos. Programación básica en el cliente (cont.). DOM. Introducción a la programación en servidor: conceptos básicos de Python.
5	Semana del 14 de septiembre	Presentación del trabajo integrador.

		Actividad individual en la práctica.
6	Semana del 21 de septiembre	Entornos Virtuales. Codificación en el servidor: aspectos de configuración de un servidor web. Microframeworks. Acceso a BBDD.
7	Semana del 28 de septiembre	El modelo MVC. API REST. Primera entrega del trabajo integrador.
8	Semana del 5 de octubre	Manejo de sesiones. Vulnerabilidades comunes en desarrollo web: SQLi y XSS. AJAX. CORS Actividad individual sobre MVC y programación cliente.servidor. Sistemas de templates. ORM
10	Semana del 12 de octubre	Otros modelos de desarrollo: Vue.js Segunda entrega del trabajo integrador.
11	Semana del 19 de octubre	Vue.js (continuación)
12	Semana del 26 de octubre	Vue.js (continuación)
13	Semana del 2 de noviembre	API: OSM - OAuth2
14	Semana del 9 de noviembre	Actividad individual sobre API y Vue.js. OAuth (Cont.)
15	Semana del 16 de noviembre	Tercera entrega del trabajo integrador.
16	Semana del 23 de noviembre	Guías para realizar el informe y la presentación final del trabajo integrador. Taller sobre ética en Informática. Taller en clase.
17	Semana del 30 de noviembre	Desarrollo del trabajo integrador
18	Semana del 7 de diciembre	Cuarta entrega del trabajo integrador. Actividad individual integradora sobre conceptos teóricos.
19	Semana del 14 de diciembre	Instancias de recuperación.

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):mail: proyecto@info.unlp.edu.arEVEA: <https://catedras.linti.unlp.edu.ar>

Firma de los profesor/es:

Prof. Claudia Banchoff

Prof. Diego Vilches

Prof. Matías Pagano