



**Ministerio de Cultura y Educación**  
**Universidad Nacional de San Luis**  
**Facultad de Ciencias Físico Matemáticas y Naturales**  
**Departamento: Informatica**  
**Area: Area V: Automatas y Lenguajes**

**(Programa del año 2024)**

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
DISEÑO Y PARADIGMAS DE LENGUAJES	ING. INFORM.	026/1	2-	2024 2° cuatrimestre
DISEÑO Y PARADIGMAS DE LENGUAJES	ING. EN COMPUT.	08/15	28/12	2024 2° cuatrimestre

### II - Equipo Docente

Docente	Función	Cargo	Dedicación
CAGNINA, LETICIA CECILIA	Prof. Responsable	P.Adj Exc	40 Hs
LOOR, FABRICIO	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	1 Hs	2 Hs	5 Hs

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

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
05/08/2024	15/11/2024	15	75

### IV - Fundamentación

Las prácticas de programación desarrolladas en cursos previos como Programación I y Programación II, brindan al futuro ingeniero una experiencia concreta en la resolución de problemas utilizando ciertos lenguajes de programación. Esta experiencia brinda también al alumno una primera aproximación a distintos aspectos y construcciones de los lenguajes de programación utilizados, como así también a los paradigmas a los cuales estos pertenecen.

Acotar el conocimiento de un ingeniero a un paradigma o lenguaje particular, acota su visión de las herramientas existentes a las particularidades del lenguaje y paradigma utilizados. Esto no responde, en general, a las necesidades actuales, con un mercado laboral que plantea problemas de diversa índole, cuya resolución efectiva puede requerir de construcciones y paradigmas de lenguajes diversos. Todo esto además, en un contexto donde permanentemente se le ofrecen al profesional informático nuevos lenguajes (o adaptaciones de los ya existentes) y paradigmas que intentan dar respuesta a las necesidades que los nuevos tipos de aplicaciones demandan.

Es así que surge la necesidad de brindar al futuro ingeniero una visión más global de los lenguajes, que profundice más allá de la consideración superficial de sus "características" y se exploren los principales conceptos de diseño subyacentes y su efecto sobre la implementación de los lenguajes. Esta visión, de acuerdo a la bibliografía especializada en el tema permite, entre otras cosas, mejorar la habilidad para desarrollar algoritmos eficaces, mejorar el uso del lenguaje de programación disponible, acrecentar el propio vocabulario con construcciones útiles de programación, hacer posible una mejor elección del lenguaje de programación y facilitar el aprendizaje de un nuevo lenguaje. Si a todo esto se le suma la identificación de los

principios subyacentes de los principales paradigmas de lenguajes de programación, y una comparación crítica entre los mismos, se puede decir que se le brindan al estudiante las herramientas necesarias para enfrentar sus necesidades presentes y futuras a la hora de elegir y usar de manera adecuada un lenguaje de programación.

## V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se espera que el alumno sea capaz de:

- Tener una perspectiva general de los paradigmas claves que se usan en el desarrollo de lenguajes de programación modernos, sus bases teóricas, aplicativas y de implementación: lenguajes Imperativos, Funcionales, Lógicos y Orientados a Objetos.
- Desarrollar una visión clara del tipo de situaciones en que los distintos paradigmas son adecuados y hacer uso de lenguajes multi-paradigma que permitan una fácil integración de los mismos y su interacción con lenguajes de programación existentes.
- Evaluar en forma crítica distintos lenguajes de programación existentes y futuros.
- Entender la implementación de distintos lenguajes con suficiente detalle como para reconocer la relación entre un programa fuente y su comportamiento en ejecución.
- Extender sus conocimientos sobre los temas anteriores con bibliografía adecuada y mínima supervisión.

La asignatura contempla el desarrollo de los siguientes ejes transversales:

- Identificación, formulación y resolución de problemas de ingeniería en informática.
- Utilización de técnicas y herramientas de aplicación en ingeniería en informática.
- Fundamentos para el desempeño en equipos de trabajo.
- Fundamentos para una comunicación efectiva.

## VI - Contenidos

### Contenidos Mínimos:

Razones del estudio de lenguajes de programación. Características de un buen lenguaje. Paradigmas de lenguajes: imperativo, funcional, orientado a objetos y lógico. Sintaxis y Semántica de los lenguajes. Descripción. Especificación e implementación de Tipos de Datos elementales y estructurados. Evolución del concepto de tipo de dato. Tipos de Datos Abstractos. Ocultamiento de la Información y encapsulamiento. Aspectos de diseño e implementación de los Lenguajes Orientados a Objetos. Administración de la Memoria. Control de Datos. Control de Subprogramas. Variantes en el control de subprogramas. Ejemplos en distintos lenguajes. Introducción a los lenguajes lógicos y funcionales. Lenguajes multi-paradigmas. Conexión a dispositivos de E/S en distintos lenguajes.

### UNIDAD I.

Razones del estudio de lenguajes de programación. Historia de los lenguajes de programación. Características de un buen lenguaje. La estructura y operación de una computadora. Computadora de hardware y firmware. Traductores e Intérpretes. Computadoras virtuales y ligaduras. Tiempos de ligadura.

### UNIDAD II.

Sintaxis y semántica de los lenguajes. Concepto de Objeto de Datos. Declaraciones. Utilidad y propósitos de las declaraciones. Operaciones. Chequeo de tipos: estático y dinámico. Conversión de tipos. Especificación e implementación de tipos de datos elementales y estructurados.

### UNIDAD III.

Definición y activación de subprogramas. Administración de la memoria. Fases de la administración de la memoria. Administración de memoria estática. Administración de memoria basada en pila. Administración de memoria basada en heap

con elementos de tamaño fijo y de tamaño variable. Mecanismos de recuperación. Ejemplos.

#### **UNIDAD IV.**

Control de subprogramas. Llamada-retorno simple. Subprogramas recursivos. Control de datos. Ambientes de referenciación. Alcance estático y dinámico. Datos compartidos en subprogramas. Parámetros. Pasaje de parámetros. Ambientes comunes explícitos. Alcance dinámico, reglas e implementaciones. Alcance estático, reglas e implementaciones.

#### **UNIDAD V.**

Variantes en el control de subprogramas. Excepciones y manejadores de excepciones. Eventos y manejadores de eventos. Corrutinas. Subprogramas planificados. Programación paralela: Comandos en guardia. Tareas.

#### **UNIDAD VI.**

Accediendo al hardware en distintos lenguajes de programación. Introducción a la conexión de dispositivos de E/S. Dispositivos de E/S. Placas Arduino o similares. Características de Hardware y Software de las placas. Conexión placa-pc. Distintas componentes de entrada/salida para incorporar a las placas. Ejemplos de aplicaciones utilizando placas y componentes.

#### **UNIDAD VII.**

Paradigmas de lenguajes: imperativo, funcional, orientado a objetos y lógico. Lenguajes de programación multiparadigma. Definición. Utilidad. Paradigma de Programación Orientada a Objetos: tipos de datos abstractos, herencia, polimorfismo. Diferentes aspectos del diseño de los lenguajes orientados a objetos: Smalltalk, C++ y Java.

## **VII - Plan de Trabajos Prácticos**

Metodología de la enseñanza: luego del dictado de las clases teóricas se espera que los alumnos refuercen los conceptos adquiridos en un trabajo práctico correspondiente a cada unidad. Los prácticos de la asignatura son de dos tipos, de aula y de formación experimental; se describen a continuación.

### a) Prácticos de aula:

Práctico 1: Tipos de Datos Elementales y Estructurados.

Objetivos: ejercitar sobre operaciones, signatura, declaraciones, tipos de datos elementales y chequeo de tipos. Analizar y ejercitar sobre la especificación e implementación de tipos de datos estructurados.

Metodología: resolución de ejercicios en papel. Implementar algunos de los ejercicios en máquina.

Práctico 2: Administración de Memoria.

Objetivos: ejercitar definición y activación de subprogramas, mecanismos de administración de memoria, problemas asociados al uso de punteros y métodos de recuperación.

Metodología: resolución de ejercicios en papel. Implementar algunos de los ejercicios en máquina.

Práctico 3: Control de Secuencia y Datos en Subprogramas.

Objetivos: realizar ejercicios referidos a ambientes de referenciación, implementación de reglas de alcance estático y dinámico.

Metodología: resolución de ejercicios en papel.

Práctico 4: Variantes en el control de subprogramas.

Objetivos: desarrollo de ejercicios con variantes en control de subprogramas: corrutinas, excepciones, eventos y concurrencia.

Metodología: investigación de las diferentes variantes en determinados lenguajes de programación. Resolución y discusión del práctico en clase.

Práctico 5: Dispositivos de Entrada/Salida.

Objetivos: programar ejercicios para visualizar cómo se realiza la entrada/salida de diferentes dispositivos usando una placa

tipo Arduino.

Metodología: resolución de ejercicios e implementación utilizando placas tipo Arduino y/o algún simulador de la mismas.

b) Prácticos de formación experimental:

Laboratorio 1: Aplicaciones de Entrada/Salida.

Objetivos: desarrollo de un proyecto donde se utilicen e implementen facilidades de E/S en placas tipo Arduino.

Metodología: resolución del proyecto utilizando placas tipo Arduino. Elaboración y entrega de un informe y/o exposición oral del laboratorio desarrollado.

Laboratorio 2: Ejecución de programas en un lenguaje multiparadigma y visualización de las características de la POO.

Con respecto a los ejes transversales se describe, a continuación, cómo se aborda y evalúa cada uno de ellos.

- Identificación, formulación y resolución de problemas de ingeniería en informática: tanto en los prácticos de aula como en los laboratorios, se espera que el alumno resuelva los ejercicios utilizando las consignas como guía. Luego, utilizando los conocimientos adquiridos en los distintos lenguajes de programación, formularán una posible resolución. La evaluación será de tipo formativa a través de la puesta en común de las distintas soluciones planteadas por los alumnos a un mismo ejercicio en las clases prácticas y de laboratorio. Se espera que la participación conjunta de los alumnos y docentes fomente el intercambio de ideas además de aportar correcciones en el caso de que sea necesario. Además, se realizará una evaluación sumativa (incluyendo las dos instancias de recuperación previstas) que servirá para acreditar los conocimientos adquiridos por los alumnos durante la cursada.
- Utilización de técnicas y herramientas de aplicación en ingeniería en informática: los alumnos aplicarán sus conocimientos adquiridos en distintos lenguajes de programación, producto del cursado de materias anteriores. La resolución de los prácticos involucra el desarrollo de programas (completos o parciales) que los alumnos pueden corroborar con el docente, y también mediante la ejecución de los mismos en entornos gráficos que soporten dichos lenguajes.
- Fundamentos para el desempeño en equipos de trabajo: en las clases prácticas se fomentará la discusión grupal para permitir que el alumno pueda compartir su opinión y nutrirse de la de sus compañeros. En los laboratorios, se sugerirá trabajar en grupos pequeños para permitir el intercambio de ideas, aunar propuestas y aportar soluciones concretas. Finalizada la etapa de resolución, se solicitará que cada alumno exponga una parte de la misma y que explique su aporte. Además, se solicitará la entrega de un informe escrito con la descripción del ejercicio, resolución del mismo y conclusiones arribadas por el grupo. Este informe se evaluará y se realizará una devolución por parte del equipo docente.
- Fundamentos para una comunicación efectiva: en las clases prácticas se fomentará la discusión de las distintas resoluciones que cada alumnos realizará de los ejercicios propuestos. Para el caso particular de los laboratorios, se incentivará la expresión oral mediante la exposición de las soluciones propuestas, la cual permitirá la sociabilización entre alumnos y docentes, además de la evaluación de las soluciones.

## VIII - Regimen de Aprobación

No se admite rendir la materia en condición de libre. El alumno puede optar por alguna de las dos alternativas del régimen de aprobación: A. Regularizar la materia y luego rendir un examen final; B. Promocionar la materia, cumpliendo las condiciones que se detallan a continuación.

A. Para regularizar la materia, el alumno deberá:

A1. Tener un mínimo de 70% de asistencia a las clases prácticas y teóricas.

A2. Entregar y aprobar el 100% de las evaluaciones periódicas obligatorias solicitadas. Estas evaluaciones consistirán de la entrega de ejercicios de prácticos de aula resueltos, y/o entrega de ejercicios extras propuestos por los docentes, y/o entrega de trabajos de desarrollo tipo informes. Los enunciados de las evaluaciones serán provistos en la plataforma Classroom y/o aulas virtuales de la materia.

A3. Aprobar los prácticos de laboratorio propuestos.

A4. Aprobar el examen parcial de los temas prácticos vistos en la materia o en su defecto, alguna de sus respectivas recuperaciones con al menos el 70% correcto del total y no menos del 50% de cada ejercicio. Tal como lo establece la reglamentación vigente, Ord. 32/14 CS, cada examen parcial tendrá dos (2) recuperaciones.

B. Para promocionar la materia, el alumno deberá cumplir con las pautas listadas a continuación y la nota final de la materia se computará promediando las calificaciones obtenidas en cada uno de los puntos mencionados:

B1. Tener un mínimo de 80% de asistencia a las clases prácticas y teóricas.

B2. Cumplir con los requerimientos A2 y A3 enunciados anteriormente.

B3. Aprobar el examen parcial de los temas prácticos vistos en la materia con al menos el 80% correcto del total y no menos del 60% de cada ejercicio.

B4. Aprobar con al menos el 80% correcto del total un examen integrador oral y/o escrito al final del cuatrimestre. La modalidad se comunicará oportunamente.

## IX - Bibliografía Básica

[1] "Programming Languages - Design and Implementation". Pratt, Terrence y Zelkowitz, Marvin. 4° edición. Prentice Hall, 2001.

[2] "Concepts of Programming Languages". Sebesta, Robert. 12° edición. Pearson India. 2023.

[3] "Programming Languages: Principles and Paradigms". Gabbrielli, Maurizio, Martini, Simone y Giallorenzo, Saverio. Springer. 2023.

[4] "Practical C Programming". Oualline, Steve. 3° edición. O'Reilly Media. 1997.

[5] "Python programming for beginners: An Introduction to the Python Computer Language and Computer Programming". Cannon, Jason. Createspace Independent Publishing Platform. 2014.

[6] "Java. Cómo programar. Deitel, Harvey y Deitel, Paul. Pearson. 2008.

[7] "Evolución del Concepto de Tipo de Datos-Tipo de Datos Abstractos-Programación Orientada a Objetos". Apunte de la cátedra. 2022.

## X - Bibliografía Complementaria

[1] "Smalltalk-80. The Language and its implementation". Goldberg, Adele y Robson, David. Addison-Wesley, 1985.

[2] "The Java class libraries". Chan, Patrick - Lee, Rosanna y Kramer, Douglas. Addison Wesley. Segunda Edición, 1998.

[3] "Piensa en Java". Eckel Bruce. Pearson Alhambra. Cuarta Edición, 2007.

[4] "Sintaxis, Traducción, Fases de la Compilación". Apunte de la cátedra. 2020.

[5] "El lenguaje de Programación C". Kernighan, Brian y Ritchie, Dennies. Prentice Hall, 1991.

[6] "Object Oriented Programming (Programming Models)". Edet, Theophilus. CompreQuest Books. 2023.

## XI - Resumen de Objetivos

El curso tiene como objetivo introducir al alumno a la problemática del diseño e implementación de lenguajes de programación, incluyendo fundamentos teóricos y prácticos. El estudio se realiza teniendo en cuenta todos los paradigmas actuales de programación, realizando un estudio comparativo de las técnicas de implementación de cada uno de ellos.

## XII - Resumen del Programa

Historia de los lenguajes de programación. Evolución de los paradigmas de programación. Computadoras virtuales. Características esenciales de los lenguajes de programación y su implementación: tipos de datos y su representación, control de secuencia y datos en subprogramas. Administración de Memoria. Abstracción de Datos. Variantes en el control de subprogramas. Programación Multi-paradigma.

### **XIII - Imprevistos**

El dictado de las clases teóricas será presencial en las instalaciones de la UNSL aunque circunstancialmente y, frente a situaciones excepcionales, se podrían impartir algunos contenidos de forma no presencial mediante videoconferencia en alguna plataforma tipo Google Meet y con el apoyo de TICs (Google Classroom, página web de la materia, Moodle, etc.). El presente programa podría presentar ajustes pero toda modificación será acordada con el estudiantado e informada a Secretaría Académica.

Información de contacto: Dra. Leticia Cagnina - lcagnina@email.unsl.edu.ar.

### **XIV - Otros**

A los efectos de que se impartan todos los contenidos mínimos de la materia y se respete el crédito horario establecido en el plan de estudios de la carrera para esta asignatura, se establece que se dicten como máximo 5hs por semana distribuidas en teorías, prácticos de aula, laboratorios y consultas, hasta completar las 75hs del crédito total.

Los laboratorios se realizarán en computadoras individuales mediante el uso de lenguajes de programación específicos y/o el uso de simuladores.