Ministerio de Cultura y Educación **(Programa del año 2024)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area V: Automatas y Lenguajes

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
FUNDAMENTOS DEL DISEÑO DE LENGUAJES DE PROGRAMACIÓN	LIC.CS.COMP.	RD-3-1/2023	2024	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
ROGGERO, PATRICIA BEATRIZ	Prof. Responsable	P.Asoc Exc	40 Hs
ARAGON, VICTORIA SOLEDAD	Prof. Colaborador	P.Adj Exc	40 Hs
FUNEZ, DARIO GUSTAVO	Responsable de Práctico	JTP Exc	40 Hs
GATICA, CLAUDIA RUTH	Auxiliar de Práctico	A.1ra Semi	20 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
Hs	3 Hs	3 Hs	2 Hs	8 Hs

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

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
05/08/2024	15/11/2024	15	120

IV - Fundamentación

Los conceptos involucrados en el estudio del diseño de lenguajes de programación y los paradigmas a los que responden, forman parte del conocimiento general de un Licenciado en Ciencias de la
Computación, con el perfil requerido por el correspondiente plan de estudios. Esta asignatura complementa los contenidos introducidos en las materias Programación I, Programación II y Lógica para Computación, donde el énfasis está puesto en la resolución de problemas utilizando diferentes lenguajes de programación, además de estudiar las características de los distintos paradigmas de programación, lo cual brinda al/la estudiante una primera aproximación a distintos aspectos y construcciones de los lenguajes de programación utilizados.
Circunscribir el conocimiento de un Licenciado en Ciencias de la Computación 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 cambiante, que plantea de forma continua problemas de diversa naturaleza, cuya resolución efectiva puede requerir de construcciones y lenguajes diversos.
Es por esto que resulta necesario brindar al futuro/a Licenciado/a una visión global de los lenguajes, donde se exploren los principales conceptos de diseño subyacentes y su efecto sobre la implementación de los lenguajes. Esta visión 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 brinda a los/las estudiantes las herramientas necesarias para enfrentar sus necesidades presentes y futuras a la hora de elegir y usar de manera adecuada un lenguaje de programación.
Además, en el convencimiento de que los aspectos básicos del diseño de compiladores deberían formar parte del conocimiento general de cualquier buen programador/a, este curso introduce la estructura básica de un compilador. También se abordan aspectos relacionados a la teoría formal de lenguajes, que son necesarios para entender el funcionamiento de un compilador típico.
Desde este lugar la propuesta es la de, en un trabajo conjunto de docentes y estudiantes poder dar respuesta a las siguientes preguntas: ¿por qué estudiar lo que subyace al concepto de lenguajes de programación?, ¿es necesario ahondar en los aspectos que refieren a la implementación de lenguajes?, ¿por qué?, ¿es posible pensarnos como futuros/as desarrolladores/as de lenguajes?, ¿cómo se puede visualizar que todo esto ayuda al desarrollo de algoritmos eficaces, a mejorar el uso de lenguajes disponibles, a incrementar el vocabulario en lo que refiere a construcciones de programación útiles? En pos a dar respuesta a estos interrogantes, teniendo como meta trabajar para lograr una buena enseñanza, todo enmarcado en un clima de camaradería entre docentes y estudiantes, propiciando el debate, el trabajo colaborativo, la tarea en grupos, la autonomía, la participación, el contexto particular del aula y la educación para un mundo mejor, y para que los/las estudiantes empiecen a verse/sentirse como futuros profesionales, las clases serán llevadas a cabo de manera coherente para trabajar en estos aspectos descriptos.

V - Objetivos / Resultados de Aprendizaje

Se presentan a continuación una serie de objetivos tanto de carácter general o transversal como de carácter específico, que se pretenden que alcanzar a lo largo del dictado del curso.

Objetivos Generales:
- Analizar críticamente los contenidos abordados, propiciando actividades de análisis de los marcos teóricos con la realidad del campo laboral.
- Vincular la teoría y la práctica con la intención de encontrar alternativas que promuevan la buena enseñanza.
- Integrar el trabajo grupal, colaborativo y la autonomía, en toda la secuencia didáctica propuesta.
- Brindar las ideas centrales del proceso completo de diseño de lenguajes de programación.

Objetivos Específicos:
- Evaluar en forma crítica distintos lenguajes de programación existentes y futuros, desde la perspectiva del diseño de los mismos.
- Responder a cuestiones tales como las motivaciones de la existencia de tantos lenguajes de programación, cómo y porqué fueron desarrollados, en qué se asemejan y difieren.
- Entender la implementación de distintos lenguajes, como para reconocer la relación entre un programa fuente y su comportamiento en ejecución. En particular comprender el funcionamiento de un compilador.
- Extender sus conocimientos sobre los tópicos anteriores con bibliografía adecuada y mínima supervisión.
- Adquirir la capacidad de evaluar lenguajes de programación desde distintos puntos de vista, ya sea como su diseñador/a, implementador/a o como usuario/a del lenguaje.

En el transcurso del dictado de esta asignatura se trabajarán de manera transversal los siguientes ejes:
* Identificación, formulación y resolución de problemas de informática.
* Utilización de técnicas y herramientas de aplicación en la informática.
* Fundamentos para el desempeño en equipos de trabajo.
* Fundamentos para la comunicación efectiva.
* Fundamentos para la acción ética y responsable.
* Fundamentos para el aprendizaje continuo.

VI - Contenidos

Contenidos mínimos:
Aspectos centrales del diseño de los lenguajes de programación. Criterios para la evaluación de lenguajes de programación. Especificación formal de lenguajes. Vinculación de lenguajes formales y lenguajes de programación. Aspectos de la

implementación de lenguajes de programación. Evolución del concepto de tipos de datos. Representación de datos en memoria, estrategias de implementación. Administración de memoria. Manejo de memoria en ejecución. Subprogramas, implementación, estructura de llamada-retorno. Datos compartidos entre subprogramas. Variantes en el control de subprogramas. Excepciones.

UNIDAD 1. Características del Diseño de Lenguajes de Programación
Historia y evolución de los lenguajes de programación. Aspectos centrales del diseño de los lenguajes de programación. Razones del estudio de lenguajes de programación. Criterios para la evaluación de lenguajes de programación. Características de un buen lenguaje. Aspectos de la implementación de lenguajes de programación: Traductores e Intérpretes. Computadoras virtuales y ligaduras. Tiempos de ligadura.

UNIDAD 2. Aspectos Formales de los Lenguajes de Programación
Especificación formal de lenguajes. Lenguajes Regulares y Lenguajes Libres del Contexto. Expresiones Regulares. Autómatas Finitos. Gramáticas Libres del Contexto. BNF, BNFE. Vinculación de lenguajes formales y lenguajes de programación.Sintaxis de los lenguajes de programación. Criterios sintácticos generales. Elementos sintácticos de un lenguaje. Etapas de la traducción. Análisis lexicográfico. Análisis sintáctico top-down por el método descendente recursivo. Errores lexicográficos y sintácticos.

UNIDAD 3. Tipos de Datos
Evolución del concepto de tipo de datos. Propiedades de tipos y objetos. Tipos de datos. Especificación e implementación (representación en memoria) de tipos de datos elementales. Operaciones. Declaraciones. Chequeo de tipos. Tipos de datos numéricos. Enumeraciones. Booleanos. Caracteres. Tipos de datos estructurados. Especificación e implementación (representación en memoria). Vectores y arreglos multidimensionales. Registros. Cadenas de caracteres. Punteros. Estrategias de implementación. Análisis semántico. Tabla de símbolos. Esquema de recuperación de errores semánticos.

UNIDAD 4. Administración de Memoria
Definición y activación de subprogramas. Administración de memoria. Manejo de memoria en ejecución. Fases de la administración de memoria. Administración de memoria estática. Administración de memoria basada en Pila. Heap con elementos de tamaño fijo y de tamaño variable. Mecanismos de Recuperación.

UNIDAD 5. Control de Datos en Subprogramas
Subprogramas, implementación, estructura de llamada-retorno. Subprogramas recursivos. Control de datos. Ambientes de referenciación. Alcance estático y dinámico. Estructura de bloques. Datos compartidos entre subprogramas. Parámetros. Pasaje de parámetros. Ambientes comunes explícitos. Alcance dinámico, reglas e implementaciones. Alcance estático, reglas e implementaciones. Representación intermedia de los programas. Código de máquina abstracta. Arquitectura de la máquina virtual. Análisis y traducción de expresiones aritméticas y lógicas, asignaciones, estructuras de control de repetición y selección. Ubicación del generador de código dentro del proceso de compilación. Esquema de traducción dirigido por la sintaxis para la generación de código intermedio para expresiones, asignaciones y estructuras de control. Generación de código intermedio embebido en un parser descendente recursivo.

UNIDAD 6. Variantes del Control de Subprogramas
Variantes en control de subprogramas. Excepciones y manejadores de excepciones. Co-rutinas. Subprogramas planificados. Programación paralela: comandos en guardia, tareas, sincronización.

VII - Plan de Trabajos Prácticos

Todos los prácticos comienzan con ejercicios sencillos para que los/las estudiantes puedan resolver de manera autónoma y posteriormente en clase consultar la resolución. Además, cuentan con ejercicios para que los/las estudiantes resuelvan en clase, con apoyo docente, en algún caso para trabajar de manera individual y en otros buscar las soluciones de manera grupal y con discusión en clase de los resultados obtenidos.

Práctico 1: Características del Diseño de Lenguajes de Programación.
Objetivos: visualizar las características de un buen lenguaje, compiladores e intérpretes, ejercitar sobre los distintos tipos de

ligaduras.
Metodología: análisis de ejercicios resueltos y resolución de ejercicios en lápiz y papel.

Práctico 2: Aspectos Formales de los Lenguajes de Programación
Objetivos: ejercitar expresiones regulares, gramáticas libres del contexto, BNF, BNFE, análisis sintáctico.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 3: Tipos de Datos Elementales y Estructurados.
Objetivos: ejercitar sobre operaciones, signatura, declaraciones, representación en memoria de tipos de datos elementales y estructurados. Chequeo de tipos.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 4: 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 lápiz, papel y computadora.

Práctico 5: Control de Datos en Subprogramas.
Objetivo: realizar ejercicios referidos a ambientes de referenciación, implementación de reglas de alcance estático y dinámico.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 6: Variantes en control de subprogramas
Objetivo: desarrollo de ejercicios con variantes en control de subprogramas: excepciones, co-rutinas, subprogramas planificados, comandos en guardia.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico de Investigación Grupal:
Dicho trabajo consistirá en la realización de un informe (en base a una guía de pautas propuestas), sobre el tema Análisis Lexicográfico, primera etapa de cualquier compilador típico.
Objetivos: investigar sobre el tema propuesto y escribir un informe que describa los aspectos centrales de esta etapa de análisis, juntamente con el uso de herramientas para la generación de analizadores lexicográficos.
Metodología: recopilación de material, análisis de esta etapa inicial de cualquier compilador típico. Uso de un analizador lexicográfico provisto por la cátedra con la finalidad de comprender su funcionamiento.

Práctico de Formación Experimental Grupal:
Desarrollo de un proyecto que involucra parte del diseño e implementación de un compilador para un lenguaje de programación determinado, para ello se trabajará con una metodología modular, esto es diseñar parte de los distintos módulos del compilador de manera incremental, primero el desarrollo del parser descendente recursivo, una vez verificada su corrección, le incorporarán el análisis semántico y luego la generación de código intermedio.
Objetivo: el objetivo general del Laboratorio es diseñar e implementar parte de un compilador para un lenguaje de Programación inspirado en C++.
Metodología: los/las estudiantes deberán programar parte del diseño de cada módulo del compilador, dado que se entregará a cada grupo código correspondiente a cada uno de los módulos para que realicen la interpretación del mismo, para luego hacer las incorporaciones que se requieran. Cada una de las 3 etapas tendrá una entrega y aprobación parcial, en las fechas predeterminadas en el cronograma de la materia.
Además los/las estudiantes deberán presentar en forma oral y escrita un informe que releve las características y decisiones de diseño tomadas para la implementación del compilador junto con el rol que desempeñó cada integrante del grupo y la dinámica grupal con la cual trabajaron, entre otros. Al finalizar cada grupo compartirá para toda la clase el trabajo realizado.

A continuación, se presenta la descripción del abordaje de cada uno de los ejes introducidos, previamente en los objetivos, como así también de las metodologías de evaluación de los mismos:
- Identificación, formulación y resolución de problemas de informática.
Se aborda en el práctico 2 y 4 y, particularmente en los prácticos de investigación y de formación experimental a partir de

trabajar en la comprensión y modificación de un compilador particular. El/la estudiante debe trabajar sobre el proyecto de diseño e implementación de un compilador para un lenguaje de programación determinado, para ello se trabajará con una metodología modular, esto es diseñar los distintos módulos del compilador de manera incremental, primero el desarrollo del parser, una vez verificada su corrección, le incorporarán el análisis semántico y luego la generación de código intermedio. Para la realización de estas actividades se proveen instructivos, explicación audiovisual de los mismos y consultas grupales y personalizadas Se realiza la evaluación a través de la presentación de: 1) códigos fuentes, su ejecución con lotes de prueba predefinidos, 2) un informe escrito y una presentación oral del mismo.

- Utilización de técnicas y herramientas de aplicación en la informática.
Se aborda la utilización de técnicas y herramientas de aplicación en los trabajos de investigación y de formación experimental. El estudiante debe programar parte del diseño de cada módulo del compilador, empleando un lenguaje de programación imperativo C, un editor de texto, software para presentaciones gráficas. De forma opcional, software para control de versiones. Las herramientas provistas las selecciona el estudiante, sólo se provee, de la cátedra, parte del código fuente del compilador. Respecto a la evaluación, se verifica que se utilicen de manera correcta el lenguaje seleccionado y las herramientas informáticas utilizadas, para ello se trabaja con entregas parciales, las cuales tendrán sus respectivas devoluciones, hasta obtener la versión definitiva.

- Fundamentos para el desempeño en equipos de trabajo.
La propuesta es que los trabajos de investigación y de formación experimental, se realice conformando equipos, por otro lado, el trabajo de investigación se realizará en grupos. Los proyectos grupales permiten que los/las estudiantes interactúen en el grupo y también, cuando se hacen las presentaciones orales, la interacción con los demás grupos. A lo largo de las entregas parciales del trabajo de formación experimental se verifica que cada integrante del grupo pueda explicar la tarea realizada. En el informe los estudiantes deben describir ventajas y desventajas de trabajar en equipo. Tanto el trabajo de formación experimental como el trabajo de investigación, tiene que ser expuesto y debatido en clase, con la participación de todos los integrantes del equipo. Finalizadas las presentaciones se realiza una devolución por parte del cuerpo docente.

- Fundamentos para la comunicación efectiva.
Tanto para el trabajo de formación experimental, como para el trabajo de investigación, se solicita la presentación de un informe escrito en formato académico, en base a una guía propuesta. Además, se solicita la presentación oral de ambos trabajos en clase. En las correcciones informadas se hace hincapié no sólo en lo disciplinar sino también en lo referido a redacción. Con la presentación del trabajo de formación experimental y el de investigación, se verifica que el/la estudiante vaya adquiriendo la capacidad de expresarse utilizando la terminología adecuada, acorde a los contenidos vistos en la asignatura, como así también que se respete el formato especificado en la guía de informe. Se evalúa la comunicación oral y la capacidad de transmitir las ideas tanto de los expositores, como de la clase en el momento de realizar preguntas. También se evalúa el tipo de presentación utilizada.

- Fundamentos para la acción ética y responsable.
Se comparte el cronograma con la descripción de las actividades/evaluaciones. En todas las actividades solicitadas se describen plazos y forma de entrega. En la redacción de informes se enfatiza sobre la correcta referencia al material bibliográfico. Se verifica que las actividades sean entregadas respetando las fechas y formatos pactados, que sean producciones propias y que en las entregas grupales hayan participado todos los integrantes.

- Fundamentos para el aprendizaje continuo.
Tanto las actividades teóricas como prácticas comienzan con un repaso de contenidos previos, con la participación de los/las estudiantes, a partir de dar respuestas a preguntas y/o consultas y/o resolución de ejercicios en pizarrón por parte de los/las estudiantes, etc. La evaluación formativa, el trabajo de investigación y el práctico de formación experimental, permiten desarrollar en el/la estudiante la capacidad de aprender de forma continua. Se verifica que todos/as los/las estudiantes participen de las distintas actividades propuestas, además de promover la participación. Se realiza una corrección informada de las actividades para que cada estudiante vaya observando el nivel de apropiación de los contenidos.

VIII - Regimen de Aprobación

La propuesta de trabajo, en lo que respecta al régimen de aprobación, está basada en la evaluación continua o formativa, con el objetivo de relacionar la información sobre la evolución del proceso de aprendizaje de los/las estudiantes con las características de la acción didáctica, a medida que se desarrollan y progresan las actividades de enseñanza y aprendizaje.

Al comenzar el dictado de la materia se proveerá a los/las estudiantes el cronograma de las distintas instancias que conforman el proceso de evaluación continua.
El/la estudiante puede regularizar (para luego rendir el examen final) o promocionar, las condiciones que se establecen son las siguientes:

A. Régimen para Estudiantes Regulares
1. Entregar, en tiempo y forma, y aprobar el 100% de las actividades prácticas requeridas por la cátedra, las cuales consistirán en: la resolución de ejercicios del tipo de los prácticos de aula y/o la entrega de ejercicios particulares de los prácticos de aula, el práctico de formación experimental y el trabajo de investigación.
2. Aprobar 1 evaluación parcial o alguna de sus 2 (dos) recuperaciones, según lo establecido en la normativa vigente. Dicha evaluación parcial se aprueba con una nota mínima de 7 (siete).
3. Contar con un porcentaje mínimo igual o superior al 70% de asistencia a clases teóricas y prácticas.
Con la finalidad de realizar un trabajo colaborativo de comprensión, se efectuarán devoluciones de las instancias evaluativas como parte del proceso de aprendizaje. Además, algunas actividades pueden tener una instancia de defensa oral posterior a la entrega donde el/la estudiante exponga, explique y responda dudas o inquietudes de sus compañeros y/o docentes acerca del trabajo realizado en sus entregas.

B. Régimen para Estudiantes Promocionales
1. Ídem a lo requerido para estudiantes regulares, salvo el ítem 3., dado que se requiere un 80 % de asistencia a las clases teóricas y prácticas.
2. Los/las estudiantes deberán responder un test teórico, en la instancia de evaluación de los prácticos 1 y 2, y en la del parcial, en la que se requiere un mínimo del 70% correcto de cada ejercicio, con el objetivo de que el/la estudiante pueda corroborar el nivel de apropiación de los contenidos teóricos, favoreciendo de esta manera la formación continua.
3. Habiendo cumplimentado los ítems 1. y 2., el/la estudiante tendrá que desarrollar y aprobar un coloquio de carácter integrador oral o escrito que incluya el análisis y desarrollo de todos los conceptos teóricos dictados en el curso.
En la nota final de aprobación se contemplarán las distintas instancias propuestas (actividades prácticas y la evaluación parcial). En todas las instancias, la nota obtenida por el/la estudiante debe ser igual a 7 o superior, incluido el coloquio de carácter integrador.
C. El curso no admite rendir el examen final en condición de Libre.

IX - Bibliografía Básica

[1] "Programming Languages - Design and Implementation". Pratt, Terrence y Zelkowitz, Marvin. Cuarta edición. Prentice Hall, 2001.
[2] "Concepts of Programming Languages". Sebesta, Robert. Addison-Wesley. Duodécima edición y undécima edición. 2022.
[3] "El lenguaje de Programación C". Kernighan, Brian y Ritchie, Dennies. Prentice Hall, 1991.
[4] "Piensa en Java". Eckel, Bruce. Pearson Alhambra. Cuarta edición, 2007.
[5] Notas de clase y Apuntes desarrollados por la cátedra.

X - Bibliografia Complementaria

[1] "Lenguajes de Programación - Diseño e Implementación". Pratt, Terrence y Zelkowitz, Marvin. Tercera edición. Prentice Hall, 1999.
[2] "Programming Language Pragmatics" por Michael L. Scott. Tercera edición. Morgan Kaufmann Publishers, 2009.
[3] "Smalltalk-80. The Language and its implementation". Goldberg, Adele y Robson, David. Addison-Wesley, 1985.
[4] The Java class libraries. Chan, Patrick - Lee, Rosanna y Kramer, Douglas. Addsion Wesley. Segunda Edición, 1998.

XI - Resumen de Objetivos

El curso tiene como objetivos introducir al/la estudiante a la problemática de las características principales del diseño e implementación de lenguajes de programación, incluyendo fundamentos teóricos y modelos formales. Comprender el funcionamiento de los modelos de implementación de los lenguajes de programación, en particular de compiladores. Adquirir la habilidad de evaluar lenguajes de programación. El estudio se realiza teniendo en cuenta todos los paradigmas actuales de

programación, propiciando el análisis crítico de los contenidos, el trabajo grupal, colaborativo y la autonomía.

XII - Resumen del Programa

Historia y evolución de los lenguajes de programación. Características de un buen lenguaje de programación. Métodos de implementación. Sintaxis de los lenguajes de programación. Gramáticas, expresiones regulares, autómatas. Análisis Lexicográficos. Características esenciales de los lenguajes de programación y su implementación. Análisis Sintáctico. Evolución del concepto de tipos de datos. Tipos de datos: especificación y representación. Análisis Semántico. Administración de Memoria. Implementación de subprogramas. Control de datos a nivel de subprogramas. Generación de Código. Variantes en el control de subprogramas.

XIII - Imprevistos

XIV - Otros

Las vías de comunicación con los estudiantes son las siguientes:
- http://www.dirinfo.unsl.edu.ar/academico/carreras/licenciatura-en-ciencias-de-la-computacion
- Classroom
- Correos electrónicos de los docentes:

Patricia Roggero: proggero@email.unsl.edu.ar
Victoria Aragón: vsaragon@email.unsl.edu.ar
Darío Funez: dgfunez@email.unsl.edu.ar
Claudia Gatica: crgatica@email.unsl.edu.ar