Ministerio de Cultura y Educación **(Programa del año 2020)**
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
ANALISIS COMPARATIVO DE LENGUAJES	LIC.CS.COMP.	18/11	2020	2° cuatrimestre
ANALISIS COMPARATIVO DE LENGUAJES	PROF.CS.COMPUT.	02/16	2020	2° cuatrimestre
ANALISIS COMPARATIVO DE LENGUAJES	LIC.CS.COMP.	006/05	2020	2° cuatrimestre
ANALISIS COMPARATIVO DE LENGUAJES	LIC.CS.COMP.	32/12	2020	2° cuatrimestre
ANALISIS COMPARATIVO DE LENGUAJES	PROF.CS.COMPUT.	06/09	2020	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
ROGGERO, PATRICIA BEATRIZ	Prof. Responsable	P.Adj Exc	40 Hs
ERRECALDE, MARCELO LUIS	Prof. Colaborador	P.Asoc 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	4 Hs	5 Hs	1 Hs	10 Hs

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

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
22/09/2020	18/12/2020	13	120

IV - Fundamentación

A partir de las resoluciones del Consejo Superior N° 30/20, N° 32/20, N° 35/20, N° 39/20 y de la Resolución de CD 18/20, en las cuales se aprueba la suspensión de actividades académicas en todo el ámbito de la Universidad Nacional de San Luis, en concordancia a las medidas adoptadas a nivel nacional por la pandemia del COVID-19, este programa se desarrollará en MODALIDAD NO PRESENCIAL, utilizando para tal fin diferentes herramientas virtuales, para el dictado de clases teóricas, clases de práctica, consultas y evaluaciones .

Este curso trata sobre los aspectos principales del diseño e implementación de los lenguajes de programación. También se incluyen la teoría y los modelos formales subyacentes que forman la base para las decisiones tomadas en la creación de este tipo de lenguajes.
En este sentido, el curso complementa los contenidos introducidos en Programación I y Programación II, donde el énfasis fue puesto en los aspectos de programación utilizando lenguajes representativos de algunos de los principales paradigmas de programación. Pero en esta asignatura, la descripción de los lenguajes se realiza por medio de la arquitectura de hardware y software subyacente que se requiere para la ejecución de programas escritos en estos lenguajes.

Los modelos formales que dan origen a los principales paradigmas de programación también son analizados con una visión más general.
En el convencimiento de que los aspectos básicos del diseño de compiladores deberían formar parte del conocimiento general de cualquier buen programador, este curso muestra de manera general la estructura básica de un compilador. Además se abordan aspectos relacionados a la teoría formal de lenguajes que son necesarios para entender el funcionamiento de un compilador, pero servirán además como introducción a tópicos más avanzados a desarrollarse en otros cursos de la carrera como Autómatas y Lenguajes, y Diseño y Construcción de Compiladores.

Los contenidos del curso se corresponden con las unidades de conocimiento recomendadas por la ACM/IEEE Computer Society Joint Curriculum Task Force, para el área de lenguajes de programación.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso los estudiantes serán capaces de:

Evaluar en forma crítica distintos lenguajes de programación existentes y futuros.

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.

Realizar un curso sobre el estudio del diseño y construcción de compiladores.

Reconocer los paradigmas claves usados en el desarrollo de lenguajes de programación modernos, sus bases teóricas, aplicativas y de implementación.

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 tópicos anteriores con bibliografía adecuada y mínima supervisión.

VI - Contenidos

UNIDAD 1. Características del Diseño de Lenguajes de Programación
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, de firmware. Traductores e Intérpretes. Computadoras virtuales y ligaduras. Tiempos de ligadura. Evolución de Paradigmas de Programación.

UNIDAD 2. Programación Lógica
Programación Lógica. El lenguaje Prolog. Elementos básicos: hechos, predicados y consultas. Átomos y variables. Variables anónimas. Matching. Reglas recursivas. Proceso de backtracking. Functores. Listas.

UNIDAD 3. Aspectos Formales de los Lenguajes de Programación
Sintaxis de los lenguajes de programación. Criterios sintácticos generales. Nociones básicas de semántica formal. Elementos sintácticos de un lenguaje. Estructura programa-subprograma completo. Etapas en la traducción. Modelos de traducción formales. Gramáticas Libres del Contexto. BNF. BNFE. Autómatas finitos. Expresiones regulares.

UNIDAD 4. Tipos de Datos
Propiedades de tipos y objetos. Tipos de datos. Especificación e implementación de tipos de datos elementales. Operaciones. Declaraciones. Chequeo de tipos. Tipos de datos numéricos. Enumeraciones. Booleanos. Caracteres. Tipos de datos estructurados. Especificación e implementación. Vectores y arreglos. Registros. Cadenas de caracteres. Punteros. Conjuntos. Control de secuencia explícito e implícito. Secuenciamiento en expresiones aritméticas. Representación de árbol.

UNIDAD 5. Administración de Memoria

Definición y activación de subprogramas. Administración de memoria. 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 6. Control de Datos en Subprogramas
Control de subprogramas. Llamada-retorno simple. La regla de la copia. Subprogramas recursivos. Control de datos. Ambientes de referenciación. Alcance estático y dinámico. Estructura de bloques. Datos compartidos en subprogramas. Parámetros. Pasaje de parámetros. Ambientes comunes explícitos. Alcance dinámico, reglas e implementaciones. Alcance estático, reglas e implementaciones.

UNIDAD 7. Variantes del Control de Subprogramas
Variantes en control de subprogramas. Excepciones y manejadores de excepciones. Corutinas. Subprogramas planificados. Programación paralela: comandos en guardia, tareas.

UNIDAD 8. Programación Orientada a Objetos
Tipos de datos abstractos. Evolución del concepto de tipo de datos. Ocultamiento de la información. Encapsulamiento mediante subprogramas. Tipos de datos abstractos genéricos. Programación orientada a objetos: herencia, polimorfismo y ligadura dinámica. Subclases y subtipos. Herencia simple y múltiple. Aspectos de diseño en Smalltalk, C++ y Java.

VII - Plan de Trabajos Prácticos

Los prácticos de la asignatura son del tipo:
a) Prácticos de aula, con un total de 60 horas.
b) Prácticos de formación experimental, con un total de 15 horas.

Práctico 1: Aspectos del diseño de lenguajes de programación.
Objetivos: visualizar las características de un buen lenguaje, computadora virtual, ejercitar sobre los distintos tipos de ligaduras.
Metodología: análisis de ejercicios propuestos.

Práctico 2: Prolog (aula y formación experimental).
Objetivos: comprender las características de un lenguaje lógico, aprender los aspectos claves del lenguaje Prolog, reglas de matching, recursión y estructuras.
Metodología: resolución de ejercicios en lápiz y papel, luego ejecutar dichos ejercicios en máquina.

Práctico 3: Descripción Formal de Lenguajes.
Objetivos: ejercitar gramáticas, BNF y BNFE.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 4: Tipos de Datos Elementales y Estructurados.
Objetivos: ejercitar sobre operaciones, signatura, declaraciones, tipos de datos elementales y estructurados. Chequeo de tipos.
Metodología: resolución de ejercicios en lápiz y papel.

Práctico 5: 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 y papel.

Práctico 6: 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 7: Variantes en control de subprogramas y POO.
Objetivo: desarrollo de ejercicios con variantes en control de subprogramas: excepciones, corutinas, subprogramas planificados y POO.
Metodología: resolución de ejercicios en lápiz y papel.

Prácticos de Formación Experimental:
Laboratorio 1: Lenguaje Prolog. Realización de un proyecto utilizando el lenguaje.
Objetivos: evaluar consultas, aplicación de las reglas de matching, programación del proyecto.
Metodología: resolución en computadora de escritorio.

Laboratorio 2: implementación del tipo arreglo unidimensional en el lenguaje Java. Realización de un proyecto diferente para cada tipo de representación (secuencial y encadenada).
Objetivos: representación del arreglo con descriptor, acceso de componentes, programación del proyecto, los estudiantes compartan su experiencia para un cabal entendimiento de los aspectos de implementación.
Metodología: resolución en computadora de escritorio.

VIII - Regimen de Aprobación

El estudiante puede regularizar (luego rendir el examen final) o promocionar. Las condiciones en las que se desarrollará el curso, en función a la modalidad de dictado no presencial, son:

A. Régimen para Estudiantes Regulares
1. Entregar el 100% de las evaluaciones periódicas solicitadas y tener aprobadas al menos el 80% de ellas. Estas evaluaciones consistirán de la entrega de ejercicios de prácticos de aula, y/o resolución y entrega de ejercicios extras propuestos por los docentes y/o entrega de trabajos de desarrollo tipo informes, programas, etc. Al menos habrá una entrega por cada Unidad desarrollada.
2. Aprobación de los prácticos de laboratorio propuestos.

Nota:
1. Setenta por ciento (70%) es el porcentaje mínimo, para aprobar las evaluaciones.
2. Si cualquier punto no fuera cumplimentado implicará que el estudiante pase a condición de libre.

B. Régimen para Estudiantes Promocionales
1. Idem a lo requerido para estudiantes regulares.
2. Además el estudiante tendrá que responder 3 cuestionarios de tipo teórico, que podrán ser orales y/o escritos, a desarrollar y/o a libro abierto, los cuáles incluirán los temas desarrollados en la asignatura.
La nota final se computará promediando, las notas obtenidas en las evaluaciones requeridas para la regularidad y la nota obtenida en cada cuestionario del ítem 2, que debe ser igual a 7 o superior, en cada una, según lo establece el Régimen Académico.

C. El curso no admite rendir el examen final en condición de Libre.

D. El examen final puede ser oral y/o escrito.

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. Edición 2019 y Edición 2016.
[3] "Prolog, programming for artificial intelligence". Bratko, Ivan. Addison-Wesley. Tercera Edición, 2001.
[4] "El lenguaje de Programación C". Kernighan, Brian y Ritchie, Dennies. Prentice Hall, 1991.
[5] Apunte de la Cátedra Aspectos formales de la traducción de lenguajes
[6] Apunte de la Cátedra de Evolución del Concepto de Tipos de Datos - Tipo de Dato Abstracto - Programación Orientada a

Objetos.
[7] Apunte de la Cátedra del Lenguaje Prolog.

X - Bibliografia Complementaria

[1] "Lenguajes de Programación - Diseño e Implementación". Pratt, Terrence y Zelkowitz, Marvin. Tercera edición. Prentice Hall, 1999.
[2] "Smalltalk-80. The Language and its implementation". Goldberg, Adele y Robson, David. Addison-Wesley, 1985.
[3] The Java class libraries. Chan, Patrick - Lee, Rosanna y Kramer, Douglas. Addsion Wesley. Segunda Edición, 1998.
[4] "Piensa en Java". Eckel Bruce. Pearson Alhambra. Cuarta Edición, 2007.
[5] "Introduction to Java". Carlos Kavka. ICTP, 2004.
[6] "Introduction to Java". Hume, J.N. Patterson y Stephenson, Christine. Canada Holt Software Associates. Primera Edición, 2000.

XI - Resumen de Objetivos

El curso tiene como objetivo introducir al alumno a la problemática del diseño e implementación de lenguajes, incluyendo fundamentos teóricos y modelos formales. 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. Sistemas de traducción. Sintaxis y semántica. Gramáticas, expresiones regulares, autómatas. Características esenciales de los lenguajes de programación y su implementación: tipos de datos y su representación, control de secuencia y datos. Administración de Memoria. Abstracción de Datos. Programación lógica, lenguaje Prolog. Control de datos a nivel de subprogramas. Variantes en el control de subprogramas.

XIII - Imprevistos

El DECNU(520/2020) de distanciamiento social, obligatorio y preventivo, establecido por el Gobierno Nacional y la necesidad de reajustar el Calendario Académico de la Universidad Nacional de San Luis, en lo referente al Segundo Cuatrimestre 2020, el Consejo Superior en su sesión del día 01/09/2020 estableció en el Artículo 1 de la Resolución Nº 68/2020, que el Segundo Cuatrimestre sea de 13 semanas. A los efectos de que se impartan todos los contenidos y se respete el crédito horario establecido en el Plan de estudios de la carrera para esta asignatura, se establece que se de como máximo 10hs por semana (algunas de las semanas) distribuidas en teorías, prácticos de aula, laboratorios, trabajos, consultas, hasta completar las 120hs.
La metodología de la asignatura tiene las siguientes características: el dictado de las clases teóricas es mediante videoconferencias en plataformas tipo zoom (o googlemeet, entre otras) apoyadas con TIC.
Los prácticos de aula y laboratorio se realizan individualmente, salvo para algún tema específico en el que el trabajo de laboratorio se podrá realizar de manera grupal, de a lo sumo 2 estudiantes, con al menos 2 consultas por semana. Para las clases de consulta también se hará uso de diferentes plataformas y el material completo estará disponible en el Aula Virtual de la UNSL y/o Classroom.

XIV - Otros