



Universidad Nacional del Nordeste.
Facultad de Ciencias Exactas y
Naturales y Agrimensura



RESOLUCIÓN:

CORRIENTES:

04 8 0 23
22 JUN 2023

VISTO el Expediente N° 09-2023-03301 por el cual la Directora de la carrera Licenciatura en Sistemas de Información, Mgter. Gladys N. Dapozo, solicita la aprobación del Programa Analítico y de Examen de la Asignatura "Paradigmas y Lenguajes", y

CONSIDERANDO que corresponde a la propuesta presentada por el Profesor Responsable de la asignatura, Lic. Ricardo Monzón;

QUE se ajusta a las adecuaciones necesarias en función de los estándares de la convocatoria de acreditación;

QUE la presentación cumple con lo requerido en las Resoluciones N° 1074/22 C.D. y N° 1075/22 C.D.

QUE la solicitud cuenta con el aval de la Comisión de Carrera respectiva.

QUE obra el informe del Gabinete Psicopedagógico de FaCENA.

QUE cuenta con el informe de la Secretaría Académica.

LO aconsejado por la Comisión de Enseñanza y Planes de Estudios, criterio compartido por este cuerpo en la sesión del día 22-06-2023;

POR ELLO:

EL CONSEJO DIRECTIVO DE LA FACULTAD DE
CIENCIAS EXACTAS Y NATURALES Y AGRIMENSURA
RESUELVE:

ARTICULO 1º) APROBAR el Programa Analítico y de Examen para la asignatura "Paradigmas y Lenguajes" del Plan de Estudio de la carrera Licenciatura en Sistemas de Información, conforme a los Anexos de la presente Resolución.

ARTICULO 2º) REMITIR copia al Profesor/a Responsable, Secretaría Académica, Dirección de Gestión Académica, Dirección de Gestión Estudios, Dirección de Gestión Biblioteca, Departamento Concurso y Carrera Docente, División Bedelía, Acreditación de Carreras, Secretaría de Departamento.

ARTICULO 3º) REGÍSTRESE, Comuníquese y archívese.
RMR/YV

Lic. YANINA MEDINA
Secretaría Académica
FaCENA - UNNE

Mgter. MARÍA VIVIANA GODOY GUGLIELMONI
DECANA
Facultad de Ciencias Exactas y Naturales y Agrimensura
Universidad Nacional del Nordeste

ES COPIA

CRISTINA GLADIS ESCOBAR
Protocolización y Archivo
FaCENA - UNNE



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:
CORRIENTES:



0480 23
22 JUN 2023

ANEXO I

PROGRAMA ANALÍTICO Y DE EXAMEN

1. IDENTIFICACION

FACULTAD DE CIENCIAS EXACTAS Y NATURALES Y AGRIMENSURA

Departamento:	Informática	
Área:	Programación	
Bloque/s de conocimiento o Trayecto/s de Formación:	Algoritmos y Lenguajes	
Nombre de la asignatura:	Paradigmas y Lenguajes	
Carácter (Obligatoria/Optativa):	Obligatoria	
Carrera:	Licenciatura en Sistemas de información	
Año en que se dicta:	Segundo	
Régimen de cursado (Bim, Trim, Cuat, Anual):	Cuatrimstral	Ubicación (1°, 2°, ...): 1°
Nombre del profesor/a responsable:	Ricardo Monzón	
Máximo título alcanzado:	Magister	

Carga horaria total:	128 hs.	
Carga horaria semanal:	8 hs.	
Teórica:	2 hs	
Teórico/ Práctica:		
Práctica:	4 hs.	
Laboratorio:	2 hs.	
Seminario:		
Otro (Especificar):		

2. DESCRIPCION:

2.1. Fundamentación

En la mayoría de las disciplinas del saber humano existe más de un camino para llegar al mismo destino. En esto la programación no ha sido una excepción. El destino, el fin en última instancia siempre ha sido el mismo. En cambio, los caminos y medios para transportarse fueron cambiando.

En los comienzos de la Programación, el puntapié fundamental fue el de la programación "lineal", que posteriormente se transformó en Programación Estructurada basado en el Paradigma Estructurado, sin abandonar la vertiente del Paradigma Imperativo. A su vez surgió el Paradigma Declarativo, con dos estilos: el Lógico y el Funcional. Una importante evolución posterior, respecto del Paradigma Imperativo es el Orientado a Objetos.

Asimismo, con el avance del tiempo y la tecnología, nos encontramos con conceptos como programación controlada por eventos, concurrencia, paralelismo y sincronización, todos ellos



Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:
CORRIENTES:



0480 23
22 JUN 2023

presentes en la programación multihilos, y sumado a todos ellos la WEB. Estos últimos conceptos forman parte de la llamada Programación Multiparadigma.

Por otra parte, la necesidad de grandes procesos de cálculo y el avance en la potencia de cálculo de los microprocesadores llevaron a profundizar conocimientos en la rama de la Concurrencia y Paralelismo, siendo esta hoy la elite de la programación en cálculo científico.

Una primera clasificación engloba casi todos los paradigmas de programación conocidos hasta ahora, que son los Paradigmas: Imperativo, Declarativo y el Concurrente (paralelo).

Específicamente, y luego de abordar todos los Paradigmas y Lenguajes de Programación existentes, la asignatura estará enfocada en el Paradigma Concurrente y el Paradigma Declarativo en su componente FUNCIONAL.

La Programación declarativa es un paradigma de programación basado en la Lógica/Funcional en el que se estudian de forma simple aspectos avanzados de los lenguajes de programación modernos. Este estilo de programación encuentra su hábitat en campos de las bases de datos, ingeniería del software, procesadores de lenguajes, lenguaje natural, investigación operativa, seguridad de redes, etc.

El Paradigma Concurrente hace referencia a la rama de la informática que trata de las técnicas de programación que se usan para expresar el potencial paralelismo entre tareas y resolver los problemas de comunicación y sincronización entre procesos. Es ampliamente utilizada en Sistemas Operativos, Aplicaciones en Tiempo Real, Sistemas Distribuidos, etc.

Para finalizar, se desarrollan temas de programación avanzada como ser: Entidades y Ligaduras, Sistemas de Tipos, Niveles de Polimorfismo, Encapsulamiento y Abstracción, Criterios de Diseño y de Implementación de Lenguajes de Programación y Nociones básicas de semántica formal.

2.2. Objetivos generales de enseñanza de la asignatura

El objetivo general de esta asignatura, es que el alumno conozca, identifique, analice y aplique la técnica de programación (lenguaje) más adecuada relacionada al problema a resolver, asociado al paradigma que representa dicha técnica. Todo esto sin dejar de lado la semántica, sintaxis y reglas propias de cada uno de ellos.

Asimismo, poder asociar las metodologías impuestas por cada paradigma con los lenguajes de programación existentes hoy en día.

En este marco, se ponen en conocimiento técnicas y herramientas avanzadas de programación para poder comprender el funcionamiento de cada paradigma y lenguaje abordado.

2.3. Metodología

El propósito de la asignatura es establecer una permanente dedicación para brindar a los alumnos herramientas que le den argumentos de base para reforzar sus conceptos de lógica y abstracción e inculcar los de concurrencia y paralelismo.

Para lograr estos objetivos, en las clases teóricas se plantean abundantes problemas de la vida real, y en las clases prácticas, se adoptarán permanentemente nuevos problemas a resolver que sean fácilmente comprensibles y de aplicación práctica visible en la vida cotidiana.

Los contenidos previstos se impartirán a través del dictado de clases teóricas, prácticas y de laboratorio, y con apoyo de material en el Aula Virtual de la UNNE. Las primeras brindarán los conceptos necesarios que serán integrados con las actividades prácticas. Estas últimas tienen como propósito fundamental que los alumnos consoliden el pensamiento lógico y la capacidad de abstracción que se requiere para resolver problemas utilizando computadoras.

Los objetivos perseguidos son que el alumno pueda integrar teoría y práctica de acuerdo a como un Analista Programador Universitario aplica sus conocimientos en una actividad profesional y no en forma aislada de la realidad.

RMR/BJO. -

ES COPIA



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:

CORRIENTES:



0480 23
22 JUN 2023

Los conocimientos Teóricos y Prácticos se apoyan sobre ejemplos concretos, buscando en forma permanente la participación activa del alumno. A partir de conocimientos adquiridos en forma vertical y horizontal se incentiva al alumno a encontrar las relaciones interdisciplinarias y su comprensión.

Los ejercicios prácticos se desarrollarán luego en el Laboratorio, bajo la supervisión del equipo docente.

Los criterios de evaluación son: Capacidad para resolver los problemas planteados utilizando las técnicas de la programación funcional y concurrente, capacidad de abstracción y de modelización, utilización de los términos técnicos específicos, capacidad de síntesis y de relación de los distintos conceptos.

Al inicio del cursado se pone a disposición de los alumnos la planificación completa de todas las actividades previstas: clases teóricas y prácticas, fechas de exámenes parciales y finales, respetando el calendario académico establecido. Esta información, disponible en su totalidad en el Aula Virtual, les permite organizar su tiempo de estudio y realizar su propia planificación, en función de sus otras obligaciones académicas.

Para atender a los alumnos, el plantel docente está formado por: un (1) Profesor Adjunto responsable de la asignatura y de la Teoría, dos (2) Jefes de Trabajos Prácticos y dos (2) Auxiliares Docentes de Primera. Adicionalmente se refuerza el plantel con dos (2) adscriptos por el ciclo lectivo correspondiente.

3. Nivel de aporte de la asignatura al desarrollo de las Competencias de Egreso de la carrera

Categoría (CE, CGT, CGS)	Competencia	0	1	2	3
CE1	Especificar, proyectar y desarrollar sistemas de información, sistemas de comunicación de datos y software cuya utilización pueda afectar la seguridad, salud, bienes o derechos.			X	
CE2	Proyectar y dirigir lo referido a seguridad informática.	X			
CE3	Establecer métricas y normas de calidad de software.		X		
CE4	Certificar el funcionamiento, condición de uso o estado de lo mencionado anteriormente.	X			
CE5	Dirigir y controlar la implementación, operación y mantenimiento de lo anteriormente mencionado.	X			
CGT1	Identificación, formulación y resolución de problemas de informática.				X
CGT2	Concepción, diseño y desarrollo de proyectos de informática.	X			
CGT3	Gestión, planificación, ejecución y control de proyectos de informática.	X			
CGT4	Utilización de técnicas y herramientas de aplicación en la informática.				X
CGT5	Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.	X			
CGS1	Fundamentos para el desempeño en equipos de trabajo.			X	
CGS2	Fundamentos para la comunicación efectiva.			X	
CGS3	Fundamentos para la acción ética y responsable.		X		
CGS4	Fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local.	X			
CGS5	Fundamentos para el aprendizaje continuo.			X	
CGS6	Fundamentos para la acción emprendedora.		X		

RMR/BJO. -

ES COPIA



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:
CORRIENTES:

1983 2023

0480 23
22 JUN 2023

4. Propuesta para el desarrollo de las competencias

Resultado de Aprendizaje	Unidades / Temas	Guía de Trabajos Prácticos	Actividad Formativa	Estrategia de Enseñanza
RA01: Aplica Paradigmas de Programación para identificar soluciones a problemas de la vida real de manera óptima y eficiente.	I.	TP 1.	Clases Teóricas. Clases Prácticas en aula y en laboratorio. Acompañamiento en la identificación y resolución de problemas.	Clases teóricas y prácticas Presenciales. Análisis de situaciones problemáticas. Aprendizaje basado en problemas. Trabajo colaborativo. Disponible en espacio virtual como complemento a las clases presenciales: Material de lectura adicional, plantillas de presentación de los temas, video, cuestionarios en línea, etc. Plataforma UNNE Moodle.
RA02: Utiliza Programación Funcional para desarrollar soluciones (código) en un entorno de Programación Declarativa.	II, III, IV, V	TP 2, TP 3, TP 4 y TP 5.	Clases Teóricas. Clases Prácticas en aula y en laboratorio. Acompañamiento en la identificación y resolución de problemas.	Clases teóricas y prácticas Presenciales. Análisis de situaciones problemáticas. Aprendizaje basado en problemas. Disponible en espacio virtual como complemento a las clases presenciales: Material de lectura adicional, plantillas de presentación de los temas, video, cuestionarios en línea, etc. Plataforma UNNE Moodle.
RA03: Utiliza Programación Concurrente y Paralela para optimizar código en entornos con potencial paralelismo.	VI, VII y VII.	TP 6, TP 7 y TP 8.	Clases Teórico-Prácticas. Supervisión del desarrollo de los trabajos prácticos.	Clases teórico- prácticas Presenciales. Análisis de situaciones problemáticas. Aprendizaje basado en problemas. Disponible en espacio virtual como complemento a las clases presenciales: Material de lectura adicional, plantillas de presentación de los temas, video, cuestionarios en línea, etc. Plataforma UNNE Moodle.

ES COPIA



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:
CORRIENTES:

1983 2023
40 AÑOS DE DEMOCRACIA EN LA REPUBLICA ARGENTINA

04 80 23
22 JUN 2023

Resultado de Aprendizaje	Unidades / Temas	Guía de Trabajos Prácticos	Actividad Formativa	Estrategia de Enseñanza
RA04: Aplica Conceptos de Programación Avanzada para aprovechar la reutilización de código.	IX	TP 9.	Clases Teórico-Prácticas. Supervisión del desarrollo de los trabajos prácticos.	Clases teórico- prácticas Presenciales. Análisis de situaciones problemáticas. Aprendizaje basado en problemas. Disponible en espacio virtual como complemento a las clases presenciales: Material de lectura adicional, plantillas de presentación de los temas, video, cuestionarios en línea, etc. Plataforma UNNE Moodle.

Unidades/Temas: Unidad temática según Programa de Contenidos

Actividad formativa: Clase Teórica, Clase Teórico-Práctica, Seminarios, Talleres, Clases Prácticas, Laboratorio, Prácticas Externas, Tutorías, Estudio y trabajo en grupo, Estudio y trabajo individual /autónomo, otros.

Estrategia de enseñanza: debate, Lección Magistral, Clase expositiva dialogada, Estudio de casos, Resolución de problemas, Aprendizaje basado en problemas, Aprendizaje orientado a proyectos, Aprendizaje cooperativo, otros

ES COPIA



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:
CORRIENTES:



5. Sistema de Evaluación

Tipo de evaluación	Criterio de Evaluación	RA N°	Técnica de evaluación
Diagnóstica	Adecuada utilización de conceptos teóricos respondiendo cuestionarios con objetivos de 60% de aciertos.	RA01, RA02, RA03, RA04.	Prueba de respuesta de selección múltiple en Aula virtual.
Formativa	Correcta aplicación de los conceptos teóricos y prácticos. Correcta y eficiente aplicación de las tecnologías a utilizar. Utilización apropiada de las herramientas de programación.	RA01, RA02, RA03, RA04	Cuestionario electrónico disponible en el Aula Virtual. Verificación del uso de herramientas.
Sumativa	Correcta y eficiente aplicación de las normas, procedimientos, técnicas y recomendaciones para afrontar situaciones de la vida real.	RA01, RA02, RA03, RA04.	Examen parcial escrito. Corrección de Trabajos Prácticos. Supervisión de trabajos de laboratorio. Cuestionarios de selección múltiple.

6. REGIMEN DE ACREDITACION

6.1. Condiciones para regularizar la materia:

- 75% de asistencia a las clases prácticas.
- 75% de asistencia a las clases de laboratorio.
- Aprobación de dos exámenes parciales (con recuperatorios y un extraordinario) con calificación igual o superior a 6(seis).

6.2. Condiciones para aprobar la materia sin examen final (promoción).

- 75% de asistencia a las clases prácticas.
- 75 % de asistencia a las clases de laboratorio.
- Obtener una calificación de 7 (siete) o más en cada uno de los parciales Teórico y Prácticos.

La nota final será el promedio de las notas de los parciales aprobados.

6.3. Condiciones para aprobar la materia con examen final.

- Los alumnos regulares rinden un examen oral o escrito sobre los contenidos teóricos del programa vigente.

Los alumnos libres deben aprobar un examen práctico para luego rendir el examen teórico, según reglamentación vigente en la FaCENA.

7. PROGRAMA ANALITICO

7.1. Contenidos mínimos (del Plan de Estudios)

Sintaxis y semántica. Nociones básicas de semántica formal. Semántica operacional. Lenguajes de programación: entidades y ligaduras. Sistemas de tipos. Conceptos de intérpretes y compiladores. Paradigmas de programación: imperativo, declarativo (funcional y lógico), orientado a objetos. Concurrencia y paralelismo.



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:
CORRIENTES:



0480 23
22 JUN 2023

7.2. Contenidos por unidad/tema

UNIDAD I: INTRODUCCIÓN.

TEMA 1. -PARADIGMAS Y LENGUAJES DE PROGRAMACION.

Paradigmas. Paradigmas de Programación. Paradigma Imperativo, Paradigma Procedimental, Paradigma Declarativo, Paradigma Demostrativo, Paradigma Orientado a Objetos, Paradigma Concurrente y Paralelo. Comparaciones.

Introducción a los Lenguajes. Clasificación. Generaciones. Historia y Características. Intérpretes y Compiladores. Estilos de Programación. Programación imperativa y programación declarativa. Ventajas e inconvenientes de la Programación Declarativa. Comparaciones. Programación dirigida por eventos. Programación Orientada a Aspectos. Programación Orientada a Objetos. Programación con restricciones. Programación por Capas. Programación Multiparadigma. Programación Concurrente y Paralela.

UNIDAD II: PROGRAMACION FUNCIONAL.

TEMA 2. - INTRODUCCION A LA PROGRAMACION FUNCIONAL. FUNCIONES Y PREDICADOS

Conceptos fundamentales de la programación funcional. Tipos de datos básicos. Nombres. Funciones. Ámbito. Tipos de Datos. Átomos, Listas y Tuplas, Funciones. Primitivas para construcción de listas. Repertorio básico de primitivas. Cálculo aritmético. Cálculo simbólico. Definición y tratamiento de funciones. Predicados. Variables locales y globales. Procedimientos como argumentos. Funciones que usan a otras funciones. Funciones sin nombres (lambda). Funciones de Control. Funciones anónimas (every, some).

TEMA 3. - MANEJO DE LISTAS Y ARREGLOS

Construcción de listas. Acceso a componentes de listas. Información de listas. Eliminación de elementos de listas. Listas/Árboles. Manipulación de estructuras. Manipulación de arreglos. Las A-Listas. Las P-Listas.

TEMA 4. - ESTRUCTURAS DE CONTROL DE FLUJO Y RECURSIVIDAD.

Secuencia de acciones. Estructuras condicionales. Estructuras iterativas. Comunicación con el usuario: entrada/salida. Recursividad. Definición de procedimientos que se usan a sí mismos. Recursividad de cola, eficiencia. Procedimientos iterativos. Aplicación explícita y reiterada de una función a todos los datos contenidos en listas. Destructividad/No destructividad. Lista de funciones.

TEMA 5 - CALCULO LAMBDA.

Introducción. Historia. Sintaxis. Convenciones sintácticas. Ámbito de variables. Variables libres y Ligadas. Relación de Equivalencia. Semántica Operacional. Reducciones. Conversiones. Equivalencia de expresiones. Sustitución. Captura de variables. Sustitución segura. Redex. Forma normal. Propiedades. Ejemplos.

RMR/BJO. -

ES COPIA



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:
CORRIENTES:



0480 23
22 JUN 2023

UNIDAD III: PROGRAMACION CONCURRENTE Y PARALELA

TEMA 6. – INTRODUCCION A LA PROGRAMACION CONCURRENTE. CONCURRENCIA Y PARALELISMO

Introducción. Programación Concurrente y Paralela. Concepto de Concurrencia. Exclusión Mutua y Sincronización. Corrección en Sistemas Concurrentes. Consideraciones sobre Hardware. Sistemas Distribuidos. Sistemas de Tiempo Real. Procesamiento Paralelo. Especificación Formal de Sistemas Concurrentes. Sentencias Concurrentes. Condiciones de Bernstein. Grafos de Precedencia. Sentencias de CoBegin y CoEnd. Sistemas con Memoria Común. Concurrencia. Ventajas. Hilos. Ventajas.

TEMA 7. – MECANISMOS.

Exclusión Mutua. Aproximaciones. Implementación. Soluciones. Algoritmos. Mecanismos para sincronización. Semáforos. Binario. Productor-Consumidor. Monitores. Sincronización. Deadlocks. Ejemplos.

TEMA 8. – COMUNICACION.

Sockets. Comunicación. Cliente. Servidor. Métodos. Paralelismo. Conceptos. Taxonomía de Flynn. Arquitectura. Multiprocesadores. Multicomputadoras. Diseño de Algoritmos. Memoria Compartida. Paso de Mensajes. MPI. Ejemplos.

UNIDAD IV: CONCEPTOS AVANZADOS.

TEMA 9. – CONCEPTOS AVANZADOS DE PROGRAMACION.

Entidades y Ligaduras. Sistemas de Tipos, Niveles de Polimorfismo. Encapsulamiento y Abstracción. Criterios de Diseño y de Implementación de Lenguajes de Programación. Nociones básicas de semántica formal.

8. BIBLIOGRAFIA:

8.1. Bibliografía Específica

1. Introducción a la Programación Paralela, Almeida Francisco, Paraninfo, 2008
2. Lógica para informáticos, Ferre Cirera Rafael, Marcombo, 2010.
3. Introduction to parallel Computing, Grama Ananth, Pearson Addison, 2003.
4. Fundamentos de Algoritmos y Programación, Lage Fernando, Nueva Librería, 2008.
5. Análisis y Diseño de Algoritmos: Implementaciones en C y Pascal, López Gustavo, Alfaomega, 2009.
6. An introduction to parallel programming, Pacheco Peter, Elsevier, 2011.
7. Programación Python 3, Summerfield Mark, Anaya Multimedia, 2009.
8. Programming Language design concepts, Watt David Anthony, Jhon Wiley & Sons, 2004.
9. Fundamentos de Programación: Algoritmos, estructuras de datos y objetos, Luis Joyanes Aguilar, Mc Graw Hill, 2003.

8.2. Bibliografía en e-libros.

Los alumnos acceden al repositorio eLibro desde la plataforma de UNNE Virtual. Contacto: e-libros@bib.unne.edu.ar

1. Ayala San Martín, G. (2020). Algoritmos y programación: mejores prácticas... Fundación Universidad de las Américas Puebla (UDLAP). <https://elibro.net/es/lc/unne/titulos/180290>

ES COPIA



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura



RESOLUCIÓN:

048023

CORRIENTES:

22 JUN 2023

2. Mancilla Herrera, A. (2015). Diseño y construcción de algoritmos. Universidad del Norte. <https://elibro.net/es/lc/unne/titulos/69931>
3. FERNÁNDEZ HUERTA, I. ; DíEZ, P. E. Fundamentos básicos de programación: aplicación práctica con Scratch y Python. 1. ed. Madrid: Delta Publicaciones, 2018. 264 p. Disponible en: <https://elibro.net/es/ereader/unne/227315?page=29>
4. Villegas Jaramillo, E. J. & Guerrero Mendieta, L. E. (2016). *Análisis y diseño de algoritmos: un enfoque práctico..* Editorial Universidad Nacional de Colombia. <https://elibro.net/es/lc/unne/titulos/129991>
5. Oviedo Regino, E. M. (2012). *Lógica de programación (2a. ed.)*. 2. Ecoe Ediciones. <https://elibro.net/es/lc/unne/titulos/215912>
6. Martínez López, F. (2015). *Teoría, diseño e implementación de compiladores de Lenguajes*. RA-MA Editorial. <https://elibro.net/es/lc/unne/titulos/106460>
7. Jiménez Millán, J. A. (2014). *Compiladores y procesadores de lenguajes*. Servicio de Publicaciones de la Universidad de Cádiz. <https://elibro.net/es/lc/unne/titulos/33847>

8.3. Bibliografía complementaria.

1. Apuntes Propios de la Cátedra, preparados especialmente de acuerdo al Programa vigente.
2. Lenguajes de Programación: Paradigma y Práctica; ISBN: 9789701019450; Julius Vandekopple, Doris Appleby, McGraw-Hill / Interamericana De México, 1998.
3. Lenguajes de Programación (2ª. Edición): Principios y Práctica, Kenneth C. Louden. Ediciones Paraninfo, S.A. 2004 ISBN: 9789706862846
4. Razonando con Haskell – Un curso sobre programación funcional, Bals C. Ruiz, Ediciones Paraninfo S.A 2004, Isbn: 9788497322775
5. Programming Language Design Concepts, David Watt, Ed. Wiley 2004. ISBN: 978047085320
6. Introducción a la Programación Funcional. ISBN 8483221764 Autor Bird Richard. Editorial Prentice-Hall. Edición 2000.
7. Introducción a la Programación Paralela. ISBN: 9788497326742. Editorial: Paraninfo. Almeida Rodríguez, Giménez Canova, Mantas Ruiz, Vidal Macia. 2008.
8. Programación Concurrente y Tiempo Real. ISBN: 978-1518608261. Vallejo Fernández, González Morcillo, Albusac Jiménez. 2016. Ed. Vallejo Fernández.

9. PROGRAMA DE EXAMEN:

No corresponde.

ES COPIA



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:
CORRIENTES:



0480 23
22 JUN 2023

10. NOMINA DE TRABAJOS PRÁCTICOS:

10.1. Resolución de situaciones problemáticas. Las clases se desarrollan en aulas. Los alumnos pueden trabajar en sus propias notebooks o con papel y lápiz.

Nro. TP	Modalidad	Tema/Descripción
1	Individual	<i>Tipos de Datos. Átomos. Listas. Funciones. Predicados. Estructuras. Destructividad-No Destructividad.</i> - Resolver ejercicios para conocer el Paradigma Funcional y sus elementos.
2	Individual	<i>Cálculo aritmético. Cálculo simbólico. Variables. Procedimientos. Funciones de Control.</i> - Resolver problemas aplicando los conceptos de Lenguaje de Programación Funcional LISP.
3	Individual	<i>Construcción y Manipulación de Listas. Inserción, eliminación, combinación.</i> - Resolver ejercicios en maquina utilizando X-LispStat para comprender las técnicas de manipulación de listas.
4	Individual	<i>Estructuras condicionales, iterativas, entrada-salida.</i> - Plantear resoluciones de problemas utilizando técnicas y herramientas del LISP.
5	Individual	<i>Recursividad, estructuras recursivas. Asignación de variables en paralelo.</i> - Aplicar técnica de recursividad en la resolución de ejercicios iterativos.
6	Individual	<i>Cálculo Lambda. Funciones. Ámbito de Variables. Equivalencia.</i> - Desarrollar funciones utilizando los conceptos de funciones lambda.
7	De a pares	<i>Concurrencia. Especificación formal. Condiciones de Bernstein.</i> - Esquematizar soluciones a problemas utilizando lógica de Bernstein.
8	De a pares	<i>Concurrencia y Paralelismo. Matriz de Concurrencia. COBEGIN y COEND.</i> - Plantear soluciones de pseudo-código para entender el potencial paralelismo de los problemas.
9	Grupal	<i>Exclusión mutua, semáforos, sincronización, deadlocks. Monitores, comunicación.</i> - Discutir ejemplos para entender las técnicas y arquitecturas de paralelismo.
10	Grupal	<i>Tipos de Datos y Entidades.</i> - Analizar mediante ejemplos los conceptos de programación avanzada.

10.2. Laboratorio / Trabajo de campo

Se implementan las mismas Guías de Trabajos Prácticos. Las clases se desarrollan en los laboratorios de informática. Los alumnos tienen a disposición computadoras para llevar a cabo la tarea.

1

RMR/BJO. -

ES COPIA



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:
CORRIENTES:



ANEXO II
CARGA HORARIA

1. IDENTIFICACION

FACULTAD DE CIENCIAS EXACTAS Y NATURALES Y AGRIMENSURA

Departamento:	Informática	
Área:	Programación	
Bloque/s de conocimiento o Trayecto/s de Formación:	Algoritmos y Lenguajes	
Nombre de la asignatura:	Paradigmas y Lenguajes	
Carácter (Obligatoria/Optativa):	Obligatoria	
Carrera:	Licenciatura en Sistemas de información	
Año en que se dicta:	Segundo	
Régimen de cursado (Bim, Trim, Cuat, Anual):	Cuatrimstral	Ubicación (1°, 2°, ...): 1°
Nombre del profesor/a responsable:	Ricardo Monzón	
Máximo título alcanzado:	Magister	
Carga horaria total:	128 hs.	
Carga horaria semanal:	8 hs.	
Teórica:	2 hs	
Teórico/ Práctica:		
Práctica:	4 hs.	
Laboratorio:	2 hs.	
Seminario:		
Otro (Especificar):		

2. RÉGIMEN DE ACREDITACIÓN

2.1. Condiciones para regularizar la materia:

- 75% de asistencia a las clases prácticas.
- 75% de asistencia a las clases de Laboratorio.
- Aprobación de dos exámenes parciales (con recuperatorios y un extraordinario) con calificación igual o superior a 6(seis).

2.2. Condiciones para aprobar la materia sin examen final (promoción).

- 75% de asistencia a las clases prácticas.
- 75 % de asistencia a las clases de Laboratorio.
- Obtener una calificación de 7 (siete) o más en cada uno de los parciales Teóricos y Prácticos.

La nota final será el promedio de las notas de los parciales aprobados.

2.3. Condiciones para aprobar la materia con examen final.

- Los alumnos **regulares** rinden un examen oral o escrito sobre los contenidos teóricos del programa vigente.

Los alumnos **libres** deben aprobar un examen práctico para luego rendir el examen teórico, según reglamentación vigente en la FaCENA.



Universidad Nacional del Nordeste
Facultad de Ciencias Exactas y
Naturales y Agrimensura

RESOLUCIÓN:
CORRIENTES:



0480 23
22 JUN 2023

3. NOMINA DE TRABAJOS PRACTICOS:

3.1. Resolución de situaciones problemáticas. Las clases se desarrollan en aulas. Los alumnos pueden trabajar en sus propias notebooks o con papel y lápiz.

Nro. TP	Modalidad	Tema/Descripción
1	Individual	<i>Tipos de Datos. Átomos. Listas. Funciones. Predicados. Estructuras. Destructividad-No Destructividad.</i> - Resolver ejercicios para conocer el Paradigma Funcional y sus elementos.
2	Individual	<i>Cálculo aritmético. Cálculo simbólico. Variables. Procedimientos. Funciones de Control.</i> - Resolver problemas aplicando los conceptos de Lenguaje de Programación Funcional LISP.
3	Individual	<i>Construcción y Manipulación de Listas. Inserción, eliminación, combinación.</i> - Resolver ejercicios en maquina utilizando X-LispStat para comprender las técnicas de manipulación de listas.
4	Individual	<i>Estructuras condicionales, iterativas, entrada-salida.</i> - Plantear resoluciones de problemas utilizando técnicas y herramientas del LISP.
5	Individual	<i>Recursividad, estructuras recursivas. Asignación de variables en paralelo.</i> - Aplicar técnica de recursividad en la resolución de ejercicios iterativos.
6	Individual	<i>Cálculo Lambda. Funciones. Ámbito de Variables. Equivalencia.</i> - Desarrollar funciones utilizando los conceptos de funciones lambda.
7	De a pares	<i>Concurrencia. Especificación formal. Condiciones de Bernstein.</i> - Esquematizar soluciones a problemas utilizando lógica de Bernstein.
8	De a pares	<i>Concurrencia y Paralelismo. Matriz de Concurrencia. COBEGIN y COEND.</i> - Plantear soluciones de pseudo-código para entender el potencial paralelismo de los problemas.
9	Grupal	<i>Exclusión mutua, semáforos, sincronización, deadlocks. Monitores, comunicación.</i> - Discutir ejemplos para entender las técnicas y arquitecturas de paralelismo.
10	Grupal	<i>Tipos de Datos y Entidades.</i> - Analizar mediante ejemplos los conceptos de programación avanzada.

3.2. Laboratorio / Trabajo de campo

Se implementan las mismas Guías de Trabajos Prácticos. Las clases se desarrollan en los laboratorios de informática. Los alumnos tienen a disposición computadoras para llevar a cabo la tarea.

RMR/BJO. -

ES COPIA