



Universidad Nacional de Santiago del Estero

Facultad de Ciencias Exactas y Tecnologías



Carrera de Licenciatura en Sistemas de Información

PROGRAMACION AVANZADA

PLANIFICACIÓN DE LA ASIGNATURA

AÑO 2016

Docentes

Adj.: Ing. Analía Méndez

J.T.P.: Ing. Raquel Zarco



1. IDENTIFICACIÓN.

1.1. Nombre de la Asignatura.

Programación Avanzada.

1.2. Carrera.

Lic. en Sistemas de Información

1.3. Plan de Estudios.

Plan 2011.

1.4. Ubicación de la Asignatura.

1.4.1. Módulo - Año.

Noveno cuatrimestre - Quinto año.

1.4.2. Ciclo al que pertenece la Asignatura.

Segundo Ciclo.

1.4.3. Área a la que pertenece la Asignatura.

ÁREAS	CARGA HORARIA EN HORAS RELOJ
Ciencias Básicas	--
Teoría de la Computación	--
Algoritmos y Lenguajes	90
Arquitectura, Sistemas Operativos y Redes	--
Ingeniería de Software, Bases de Datos y Sistemas de Información	--
Aspectos Profesionales y Sociales	--
Otra	--
CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR	90

1.4.4. Carga horaria semanal.

La asignatura tiene prevista una carga horaria semanal de 6 horas, de las cuales se dedican 4 horas semanales a la formación práctica. Su cursado es de carácter cuatrimestral y su carga horaria total es de 90 horas.

1.4.5. Correlativas anteriores.

Regular: Sistemas Operativos Distribuidos.

Aprobada: Redes II.

1.4.6. Correlativas posteriores.

Sistemas de Información III.

1.5. Objetivos establecidos en el Plan de Estudios para la Asignatura.

El Plan de Estudios no establece objetivos específicos para la asignatura Programación Avanzada, pero cabe destacar los siguientes objetivos indicados para el área Algoritmos y Lenguajes expresados como competencias a lograr por los estudiantes y que se relacionan con la asignatura:

Desarrollar en los estudiantes competencias específicas para...

- Aplicar distintos paradigmas de programación en la resolución de problemas, empleando estructuras de control y estructuras de datos.
- La verificación de la solución de algoritmos desarrollados en los distintos paradigmas de programación.
- El uso de distintos lenguajes de programación como herramientas computacionales.

1.6. Contenidos mínimos establecidos en el Plan de Estudios para la Asignatura.

Concurrencia. Algoritmos concurrentes: problemas de naturaleza concurrente, arquitecturas que soportan la programación concurrente. Lenguajes de aplicación. Procesos concurrentes. Problemas inherentes a la programación concurrente. Verificación de programas concurrentes. Algoritmos distribuidos: taxonomía de los sistemas distribuidos. Modelo computacional distribuido. Declaración de datos distribuidos. Distribución de estado. Lenguajes de aplicación. Patrones comunes de programación distribuida. Protocolos de distribución. Seguridad. Paralelismo. Algoritmos paralelos: Problemas inherentes a la programación paralela. Modelos de paralelismo. Análisis y diseño de algoritmos paralelos. Herramientas y lenguajes de programación paralela.

1.7. Año Académico.

Año 2016.

2. PRESENTACIÓN.

2.1. Ubicación de la Asignatura como tramo de conocimiento de una disciplina.

Esta asignatura corresponde al Área de Algoritmos y Lenguajes. Está orientada al estudio, análisis y modelación de la Concurrencia y los Algoritmos Concurrentes,



Distribuidos y Paralelos, mediante la aplicación de lenguajes de programación. Está centrada en el tratamiento de la concurrencia en los sistemas de computación.

La programación concurrente originada en los años 60 en un inicio fue de importancia para el desarrollo de sistemas operativos, posteriormente con el diseño de máquinas multiprocesadoras ofreció no sólo un reto para los diseñadores de sistemas operativos, sino una oportunidad que los programadores podían aprovechar.

El primer reto para los programadores fue el problema de la sección crítica y para mediados de los 70's se hizo evidente la necesidad de contar con métodos formales para controlar la complejidad de los programas concurrentes.

En la actualidad la proliferación del procesamiento paralelo, del procesamiento distribuido, del procesamiento cliente-servidor, la utilización de Internet y las estaciones y PC multiprocesos han generalizado el hardware concurrente y han hecho más relevante a la programación concurrente.

En este contexto se plantea esta asignatura con el objeto de brindar al alumno la posibilidad de conocer y aplicar los modelos y principios de la concurrencia, analizar algoritmos paralelos y distribuidos, y desarrollar destreza en la aplicación de lenguajes de programación en problemas de concurrencia y sistemas distribuidos.

2.2. Conocimientos y habilidades previas que permiten encarar el aprendizaje de la Asignatura.

En relación a la asignatura Sistemas Operativos Distribuidos:

- Comprensión de los conceptos relativos a Sincronización, Comunicación y Coordinación Distribuida.
- Habilidades y destrezas en la aplicación de Control de Concurrencia en ejercicios prácticos básicos sobre Sistemas Distribuidos.
- Conocimientos básicos respecto a Transacciones Distribuidas y Seguridad en Sistemas Distribuidos.

En relación a la asignatura Redes II:

- Conocimientos básicos respecto a la administración de redes.
- Habilidades y destrezas en la resolución de problemas y ejercicios prácticos utilizando conceptos relativos a Sistemas cliente – servidor y redes de computadoras en general.

2.3. Aspectos del Perfil Profesional del Egresado a los que contribuye la asignatura.

Esta asignatura contribuye en diversos aspectos al perfil profesional de un egresado como Licenciado en Sistemas de Información, entre los que se destacan los siguientes:

- Proporcionar al estudiante de los conocimientos básicos (lógico-matemáticos y computacionales) para una adecuada fundamentación teórica de su quehacer profesional específico.
- Otorgar al estudiante profundos conocimientos, entre otros, sobre Teoría de Sistemas, de la Información y de la Comunicación, que le permiten fundamentar el diseño y aplicación de Sistemas de Información y Modelos.
- Capacitar al estudiante para participar con otros especialistas de su disciplina, en la selección y configuración de Sistemas de Computación, según requerimientos de las distintas áreas de aplicación.
- Permitir al estudiante comprender, predecir y justificar el comportamiento de los Sistemas de Información.
- Generar en el estudiante capacidades para diseñar y aplicar Sistemas de Información a diferentes tipos de organizaciones con diferentes estructuras.

3. OBJETIVOS.

3.1. Competencias a adquirir por los estudiantes.

3.1.1. Competencias transversales o genéricas.

3.1.1.1. Competencias instrumentales:

- Capacidad de organización, de abstracción y comprensión.
- Capacidad de gestión de la información.
- Resolución de problemas.

3.1.1.2. Competencias personales:

- Capacidad de trabajo en equipo.
- Razonamiento lógico y crítico.

3.1.1.3. Competencias sistémicas:

- Motivación por la calidad.
- Conocimientos relativos a programas, computadoras, algoritmos y lenguajes de programación.

3.1.2. Competencias específicas.

3.1.2.1. Competencias cognitivas (saber):

- Comprensión de los conceptos básicos de la concurrencia, los sistemas distribuidos y el paralelismo.

- Comprensión de modelos teóricos que posibilitan concurrencia y paralelismo.
- Conocer las técnicas y algoritmos empleados en la programación concurrente.
- Adquirir una visión general de los mecanismos específicos de concurrencia y sincronización en los algoritmos asociados a la programación concurrente así como una visión conjunta de los aspectos más sobresalientes en la programación concurrente y distribuida: procesos, mecanismos de sincronización, mecanismos de comunicación, y seguridad.
- Reconocer y distinguir la ejecución de los procesos concurrentes y distribuidos mediante la exposición de algunos problemas clásicos de programación concurrente: sección crítica, productores y consumidores, lectores y escritores, problema de los filósofos.
- Conocer y entender el funcionamiento de elementos fundamentales a nivel de arquitectura en el diseño de computadores paralelos de memoria compartida y de memoria distribuida.

3.1.2.2. Competencias procedimentales e instrumentales (saber hacer):

- Capacidad para aplicar el concepto de concurrencia.
- Habilidad para tratar problemas básicos de programación concurrente.
- Destreza aplicando lenguajes de programación en problemas de concurrencia y sistemas distribuidos.
- Capacidad para desarrollar, aplicar y verificar la concurrencia entre programas.
- Capacidad para concebir, deducir, recomponer y/o corroborar un código fuente determinado en el que se aplica concurrencia.
- Habilidad para aplicar las técnicas básicas para implementar propiedades de seguridad en programas concurrentes.
- Capacidad para construir aplicaciones concurrentes mediante la utilización de lenguajes tales como Java.
- Capacidad para corregir y replantear resoluciones a problemas de concurrencia mediante la modificación o reutilización de código.
- Habilidad para programar aplicaciones sencillas en un entorno de programación paralela bajo el modelo de paso de mensajes a través de un interfaz MPI (Message Passing Interface) estándar.

3.1.2.3. Competencias actitudinales (ser):

- Habilidad para trabajar de forma autónoma.



- Inquietud para la eficiencia y el rigor.
- Capacidad para comunicar resultados de forma clara y precisa.
- Capacidad para reaccionar frente al error identificando y verificando nuevas estrategias para alcanzar el logro exigido.
- Disposición al análisis crítico y auto-crítico sobre códigos de programa, buscando siempre la superación.

4. SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS.

4.1. Programa Sintético sobre la base de los contenidos mínimos.

Unidad I: CONCURRENCIA. Algoritmos concurrentes: problemas de naturaleza concurrente, arquitecturas que soportan la programación concurrente. Lenguajes de aplicación. Procesos concurrentes. Problemas inherentes a la programación concurrente. Verificación de programas concurrentes.

Unidad II: PROGRAMACIÓN PARALELA. Paralelismo. Algoritmos paralelos: Problemas inherentes a la programación paralela. Modelos de paralelismo. Análisis y diseño de algoritmos paralelos. Herramientas y lenguajes de programación paralela.

Unidad III: PROGRAMACIÓN DISTRIBUIDA. Algoritmos distribuidos: taxonomía de los sistemas distribuidos. Modelo computacional distribuido. Declaración de datos distribuidos. Distribución de estado. Lenguajes de aplicación. Patrones comunes de programación distribuida. Protocolos de distribución. Seguridad.

4.2. Articulación Temática de la Asignatura /Obligación Curricular

En la figura 1 se presentan los principales conceptos a tratar en la asignatura y la relación entre los mismos.

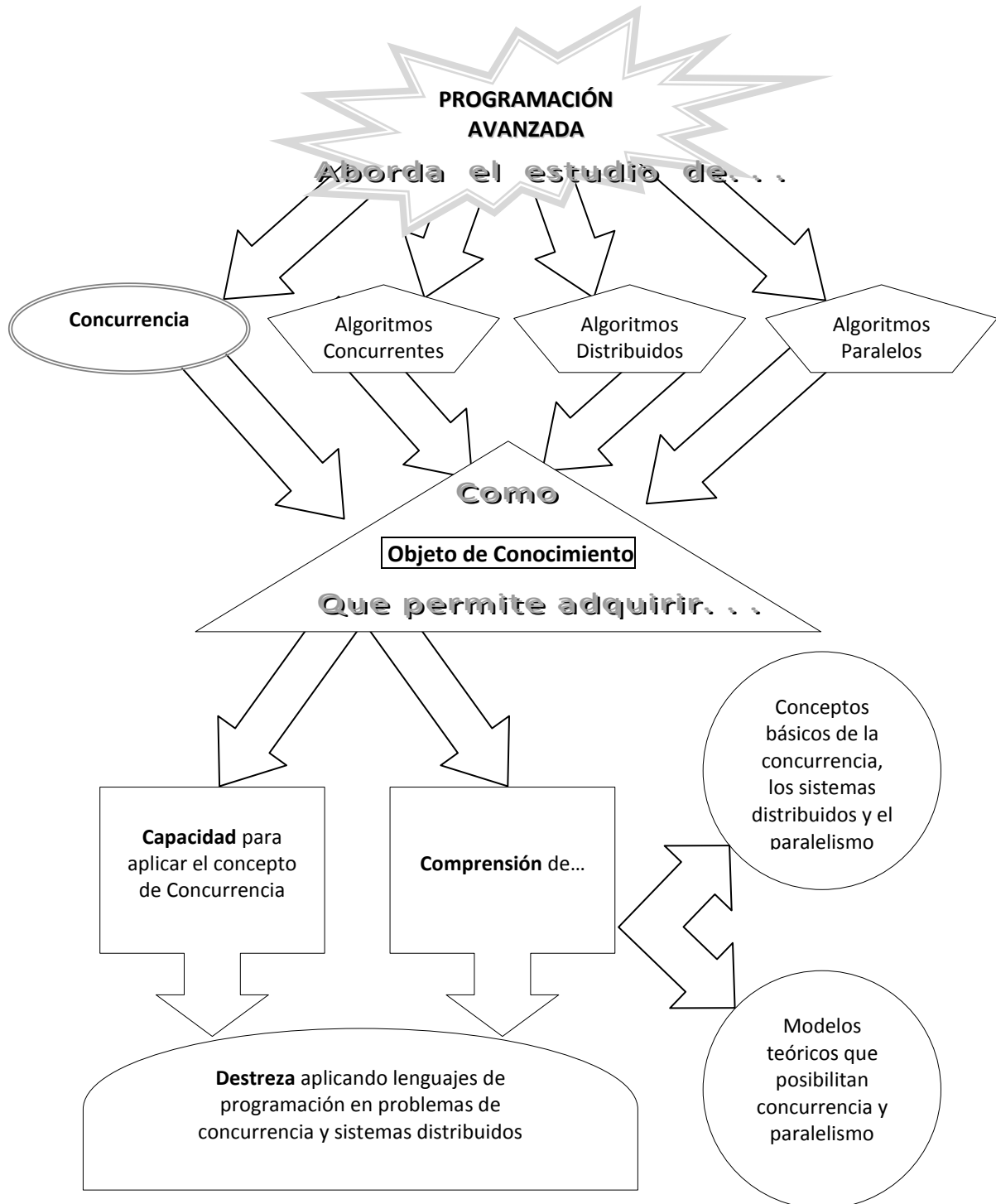


Figura 1. Principales conceptos y sus relaciones.

4.3. Integración horizontal y vertical con otras asignaturas.

En la figura 2 se muestra la ubicación de la asignatura Programación Avanzada en el 5º Año, 9º módulo, de la carrera de Licenciatura en Sistemas de Información. Se incluyen todas las asignaturas relacionadas cursadas previamente por el alumno, indicándose con línea punteada las relaciones de correlatividad débil entre las materias y con línea llena las relaciones de correlatividad fuerte.

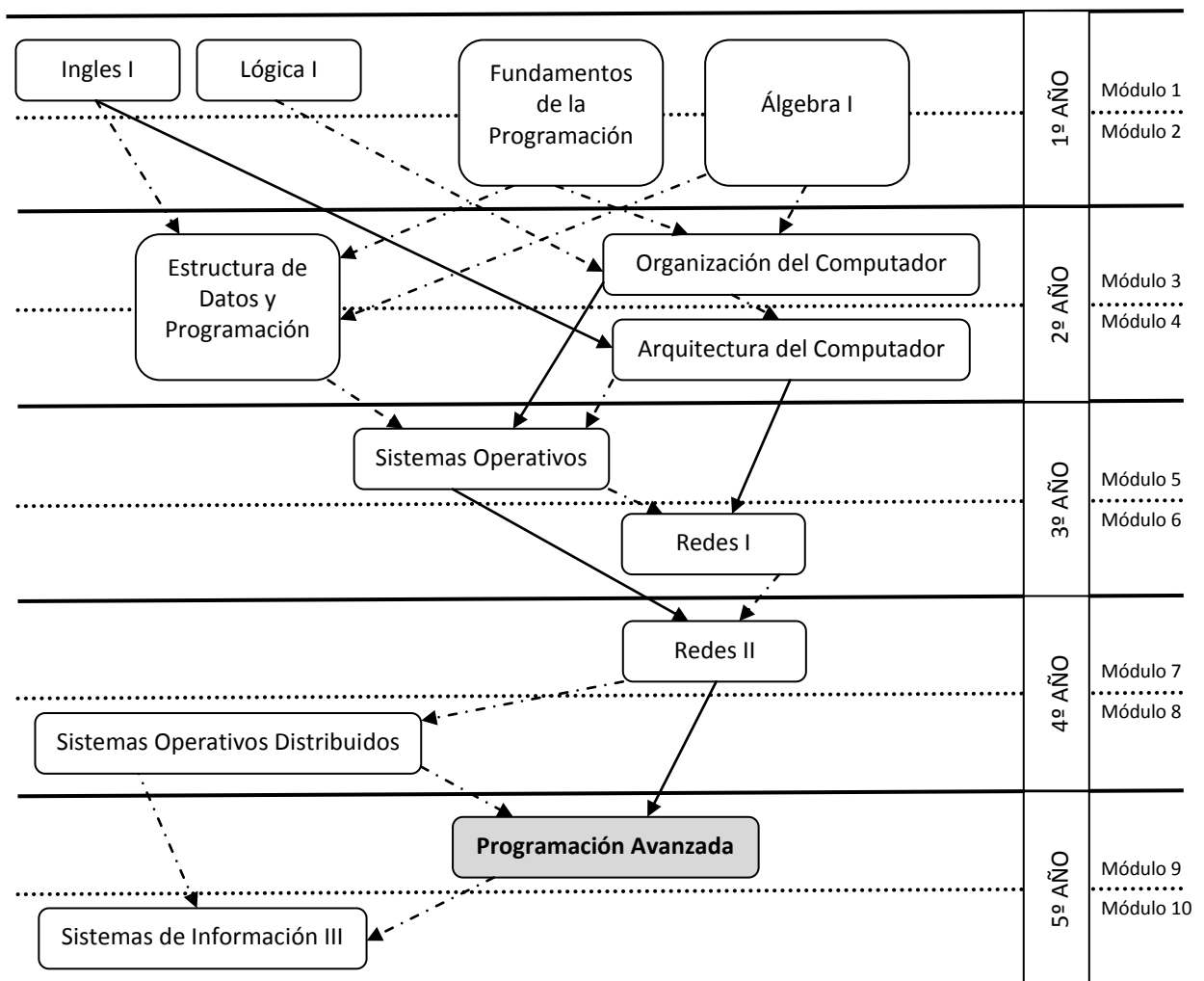


Figura 2: Articulación horizontal y vertical de la asignatura en el Plan de Estudios de la Licenciatura en Sistemas de Información

Se presenta a continuación en la figura 3 un esquema gráfico representativo de la integración horizontal y vertical con las asignaturas directamente relacionadas con Programación Avanzada.

Se expresan las vinculaciones temáticas con las asignaturas Redes II y Sistemas Operativos Distribuidos, como correlativas previas, y la asignatura Sistemas de Información III, como correlativa posterior.

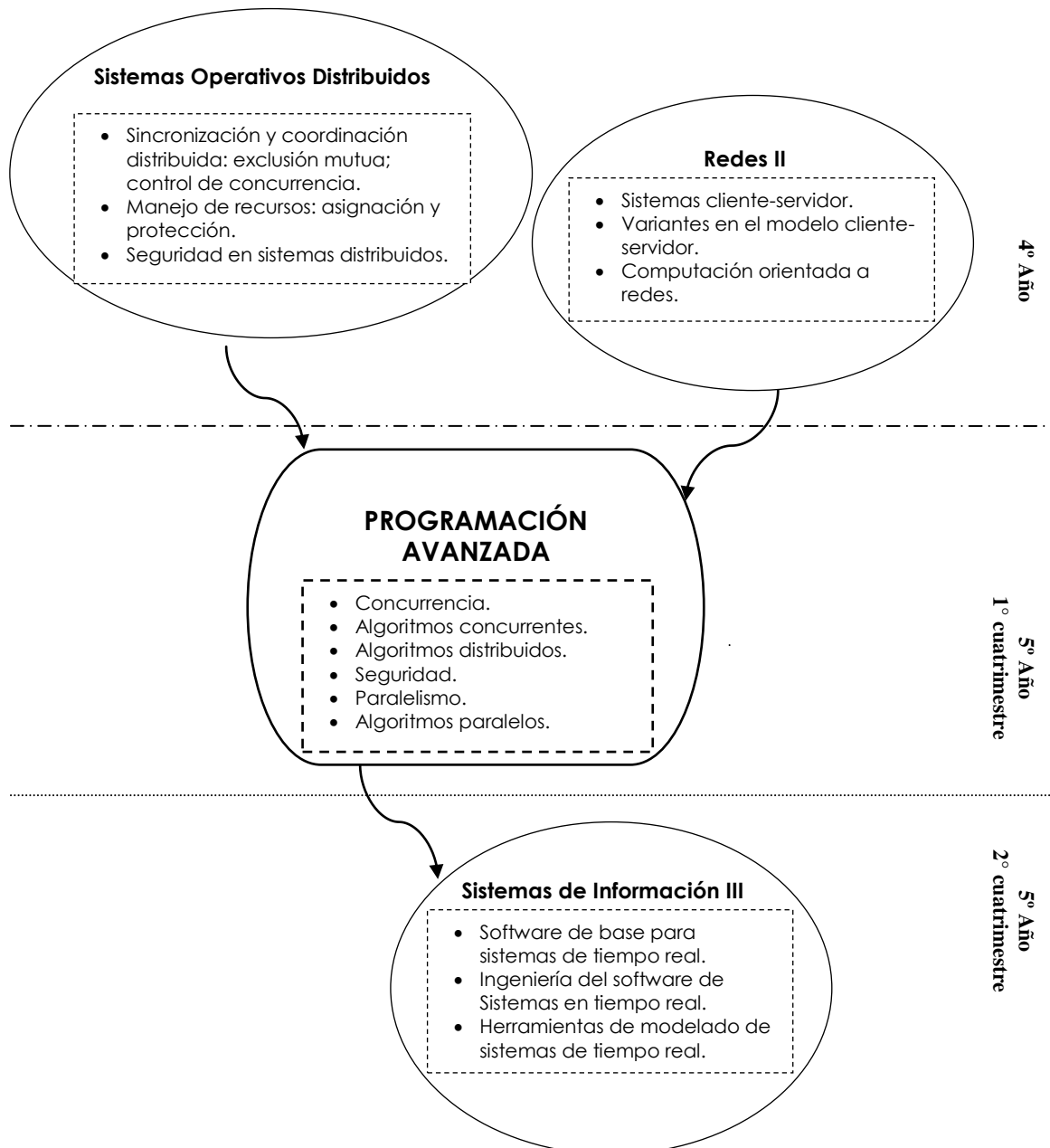


Figura 3: Articulación horizontal y vertical de la asignatura con sus correlativas previas y posteriores.



4.4. Programa Analítico

Unidad I: CONCURRENCIA.

Procesos concurrentes: Concurrencia. Concepto. Concurrencia inherente o potencial. Problemas inherentes a la programación concurrente.

Ejecución concurrente: Lenguajes de alto nivel. Abstracción de la concurrencia. Lenguajes concurrentes.

Características de los sistemas concurrentes. Orden de ejecución de instrucciones. Indeterminismo.

Algoritmos concurrentes: Problemas de naturaleza concurrente. Ejemplos. Arquitecturas que soportan la programación concurrente.

Verificación de programas concurrentes: Propiedades de seguridad. Análisis de algoritmos concurrentes.

Lenguajes de aplicación: Lenguajes de programación concurrentes. Programación concurrente en Java.

Unidad II: PROGRAMACIÓN PARALELA.

Paralelismo. Modelos de paralelismo. Programación mediante paso de mensajes. Programación en memoria compartida.

Algoritmos paralelos: Problemas inherentes a la programación paralela. Paralelismo de datos. Particionado de datos. Esquemas en árbol y grafo.

Análisis y diseño de algoritmos paralelos: Tiempo de ejecución paralelo. Modelado del tiempo de ejecución. Medidas de rendimiento de algoritmos paralelos. Escalabilidad. Herramientas para estudiar el paralelismo.

Metodología de la programación paralela: Descomposición de tareas. Asignación de tareas. Diseño modular de programas paralelos.

Herramientas y lenguajes de programación paralela.

Unidad III: PROGRAMACIÓN DISTRIBUIDA.

Algoritmos distribuidos: Taxonomía de sistemas distribuidos. Modelo computacional distribuido.

Declaración de datos distribuidos. Distribución de estado. Distribución abierta y nombres globales. Declaración de datos compartidos. Distribución global. Distribución local. Comunicación con canales.



Patrones comunes de programación distribuida: Objetos estacionarios y móviles. Objetos asincrónicos y flujo de datos.

Protocolos de distribución: Entidades de lenguaje. Protocolo de estado móvil. Protocolo de ligadura distribuida. Administración de memoria. Seguridad.

Lenguajes de aplicación. Programación distribuida en Java. Programación Sockets Java.

4.5. Programa y cronograma de Actividades de Formación Experimental.

4.5.1. Programa de Trabajos Prácticos.

	TEMA	COMPETENCIAS ESPECÍFICAS
PRÁCTICO 1	Introducción a la programación concurrente	<ul style="list-style-type: none"> • Capacidad para aplicar el concepto de concurrencia. • Habilidad para tratar problemas básicos de programación concurrente.
PRÁCTICO 2	Hilos en Java	<ul style="list-style-type: none"> • Capacidad para desarrollar, aplicar y verificar la concurrencia entre programas. • Capacidad para concebir, deducir, recomponer y/o corroborar un código fuente determinado en el que se aplica concurrencia. • Capacidad para construir aplicaciones concurrentes mediante la utilización de lenguajes tales como Java.
PRÁCTICO 3	Paso de mensajes con MPI	<ul style="list-style-type: none"> • Destreza aplicando lenguajes de programación en problemas de concurrencia y sistemas distribuidos. • Habilidad para aplicar las técnicas básicas para implementar propiedades de seguridad en programas concurrentes. • Capacidad para corregir y replantear resoluciones a problemas de concurrencia mediante la modificación o reutilización de código. • Habilidad para programar aplicaciones sencillas en un entorno de programación paralela bajo el modelo de paso de mensajes a través de un interfaz MPI (Message Passing Interface) estándar.

	OBJETIVOS	MODALIDAD	CRITERIOS DE EVALUACIÓN
PRÁCTICO 1	<ul style="list-style-type: none"> • Comprender y aplicar el concepto de concurrencia. • Analizar y resolver problemas básicos de programación concurrente. 	Ejercicios del tipo Múltiple Choice con justificación. Resolución de problemas.	<ul style="list-style-type: none"> • Presentación • Claridad en exposición • Completitud
PRÁCTICO 2	<ul style="list-style-type: none"> • Desarrollar programas concurrentes, aplicar y verificar la concurrencia entre dichos programas. • Identificar y corroborar la aplicación de concurrencia en un código fuente determinado. 	Resolución de problemas	<ul style="list-style-type: none"> • Manejo conceptual y metodológico • Manejo bibliográfico

	OBJETIVOS	MODALIDAD	CRITERIOS DE EVALUACIÓN
	<ul style="list-style-type: none"> • Construir aplicaciones concurrentes mediante la utilización de lenguajes tales como Java. 		
PRÁCTICO 3	<ul style="list-style-type: none"> • Aplicar lenguajes de programación en problemas de concurrencia y sistemas distribuidos. • Corregir y replantear resoluciones a problemas de concurrencia mediante la modificación o reutilización de código. • Programar aplicaciones sencillas en un entorno de programación paralela bajo el modelo de paso de mensajes a través de una interfaz MPI (Message Passing Interface) estándar. 	Resolución de problemas	

4.5.2. Cronograma de Prácticos.

PRÁCTICO	DESARROLLO	PRESENTACIÓN
1	Marzo – Abril	4º Semana de Abril
2	Mayo	3º Semana de Mayo
3	Junio	3º Semana de Junio

El desarrollo de los trabajos prácticos se llevará adelante en laboratorio adecuadamente equipado por software y hardware de computadoras, realizándose la presentación de los problemas y el seguimiento y apoyo de las actividades de los alumnos.

4.5.3. Programa de Talleres.

TALLER	TEMA	OBJETIVO	CRITERIOS DE EVALUACIÓN
1	Sincronización de hilos aplicando semáforos	Transformar aplicaciones secuenciales en aplicaciones concurrentes	o Nivel de aplicación de los conceptos estudiados.
2	Implementación de aplicaciones utilizando Java concurrente	Implementar aplicaciones utilizando diferentes patrones de concurrencia	o Correcta resolución del problema abordado y adecuada exposición de las conclusiones obtenidas.
3	Implementación de algoritmos distribuidos con MPI	Programar algoritmos bajo el modelo de paso de mensajes a través de una interfaz MPI estándar	o Capacidad para el análisis crítico.

4.5.4. Cronograma de Actividades de Formación Experimental.

TALLER	FECHA	CARGA HORARIA EN CLASE
1	5° Semana de Abril 1° Semana de Mayo	10
2	4° y 5° Semanas de Mayo	10
3	4° y 5° Semanas de Junio	10

Las actividades de formación práctica se llevarán a cabo en laboratorio equipado con computadoras y software necesario para el desarrollo de las actividades.

5. BIBLIOGRAFIA.

5.1. Bibliografía Específica.

Título	Autor/es	Editorial	Año y Lugar de edición	Disponible en	Cant. Ejemplares disponibles
INTRODUCCION A LA PROGRAMACION PARALELA	Almeida - Giménez - Mantas - Vidal	Paraninfo	2008	Biblioteca del Departamento de Informática	1
CONCEPTS, TECHNIQUES, AND MODELS OF COMPUTER PROGRAMMING	Peter Van Roy - Seif Haridi	The MIT Press	2004		1
PROGRAMACIÓN CONCURRENTE EN JAVA.	Doug Lea	Addison Wesley	2001		1
PROGRAMACIÓN EN JAVA 2 - ALGORITMOS, ESTRUCTURAS DE DATOS Y PROGRAMACIÓN ORIENTADA A OBJETOS.	Joyanes Aguilar - Zahonero Martinez.	Mc Graw Hill.	2002		1
SISTEMAS DE TIEMPO REAL Y LENGUAJES DE PROGRAMACIÓN	Alan Burns y Andy Wellings	Pearson Education	2003		1

2.3. Bibliografía General o de Consulta.

2.3.1. Referencias web.

Título	Dirección
Designing and Building Parallel Programs, by Ian Foster	http://www.mcs.anl.gov/~itf/dbpp/

Título	Dirección
Concurrency: State Models & Java Programs	http://www.doc.ic.ac.uk/~jnm/book/
Inside the Java Virtual Machine Thread Synchronization by Bill Venners	http://www.artima.com/insidejvm/ed2/threadsynch.html
Introducción a MPI	http://informatica.uv.es/iiguia/ALP/materiales2005/2_2_introMPI.htm
The Message Passing Interface (MPI) standard	http://www.mcs.anl.gov/research/projects/mpi/
Open MPI: Open Source High Performance Computing	https://www.open-mpi.org/

2.3.2. Archivos digitales.

Título	Autor	Editorial	Cant. De Páginas	Tamaño
Concepts, Techniques, and Models of Computer Programming	Peter Van Roy - Seif Haridi	The MIT Press Cambridge, Massachusetts, London, England	931	9,89 MB

NOTA: este archivo estará disponible para los estudiantes en el equipamiento del Laboratorio de Informática.

6. ESTRATEGIAS METODOLOGICAS.

6.1. Aspectos pedagógicos y didácticos.

El desarrollo de la asignatura estará caracterizado por una combinación de exposición oral explicativa por parte de los docentes y de un estudio dirigido de actividades prácticas, y actividades de Taller.

Para cada uno de los bloques fundamentales de la materia: concurrencia, programación paralela, y programación distribuida, la metodología de enseñanza constará de los siguientes pasos:

- Exposición de los fundamentos básicos de cada bloque con ejemplos ilustrativos.
- Puesta en práctica de los fundamentos básicos mediante pequeños ejemplos dirigidos utilizando recursos hardware y software, con el fin de consolidar la adquisición de dichos conocimientos básicos.
- Exposición de los contenidos más avanzados de cada bloque ilustrados con un caso de estudio seleccionado.
- Puesta en práctica de los contenidos avanzados a través de casos de estudio.

Como consecuencia de la metodología de enseñanza, el objetivo es conseguir un aprendizaje incremental por parte del alumno de los contenidos de los bloques de la materia. En un primer paso, el alumno adquiere los fundamentos básicos; a continuación, los consolida a través de la realización de supuestos prácticos representativos; luego, el alumno adquiere contenidos más avanzados apoyados también por casos prácticos; finalmente, el alumno dispone de los medios necesarios para afrontar individualmente un supuesto práctico avanzado sobre la materia.

6.2. Actividades de los alumnos y de los docentes.

6.2.A. Actividades de los docentes:

Profesores Adjuntos:

- Desarrollar las clases teórico - prácticas.
- Atender consultas de los alumnos.
- Coordinar el equipo cátedra.
- Seleccionar bibliografía y preparar material didáctico.
- Evaluar permanentemente.
- Supervisar el desarrollo de las clases prácticas.
- Supervisar la preparación de los trabajos prácticos.

Auxiliar Docente:

- Coordinar el desarrollo de las clases prácticas.
- Desarrollar las clases prácticas.
- Preparar los trabajos prácticos.
- Atender consultas de alumnos.
- Colaborar en la preparación de material didáctico.
- Colaborar y participar en el proceso de evaluación.

6.2.B. Actividades de los alumnos

Las actividades de los alumnos son:

- Asistir y participar de las clases de la asignatura.
- Desarrollar las prácticas propuestas por la cátedra y presentarlas en tiempo y forma.
- Realizar revisión y consulta bibliográfica para profundizar los temas estudiados en clase.



- Conformar grupos de estudio con actitud colaborativa y con una participación activa y dedicada.
- Integrar, participar y colaborar en las actividades grupales que se propongan.
- Exponer y defender teóricamente los trabajos propuestos con dicha modalidad.
- Participar y completar las actividades de Prácticos y Talleres propuestas.

6.3. Mecanismos para la integración de docentes.

Se plantea principalmente la integración con la práctica de la asignatura Sistemas Operativos Distribuidos. Ésta integra el módulo 8° del 4° año de la carrera de Licenciatura en Sistemas de Información y ahonda exhaustivamente conceptos Sincronización y coordinación distribuida: exclusión mutua; control de concurrencia, Manejo de recursos: asignación y protección, así como Seguridad en sistemas distribuidos.

6.4. Cuadro sintético.

Teóricas	Formación Practica					
	Formación Experimental	Resolución de problemas del mundo real	Actividades de proyectos y diseño de Sistemas de Información	Instancias supervisadas de formación en la práctica profesional	Otras	Total
30	60	--	--	--	--	60

6.5. Recursos didácticos.

Se prevé la utilización de recursos didácticos: pizarra, libros, apuntes, Internet, plataforma Moodle del Centro Universitario Virtual (CUV), laboratorio de informática, diapositivas, retroproyector o videoprojector y softwares, tales como:

- IDE NetBeans (Entorno de Desarrollo Integrado).
- Java EE (Java Platform Enterprise Edition).
- Lenguaje C/C++.
- Biblioteca MPI (Message Passing Interface, Interfaz de Paso de Mensajes), OpenMPI.
- Sistema operativo Linux.
- Sistema operativo Windows.

7. EVALUACIÓN.

7.1. Evaluación diagnóstica.

Esta evaluación se llevará a cabo en la etapa inicial de la asignatura y tendrá por objetivo la constatación del nivel de conocimientos previos del alumnado. Así mismo, proporcionará información útil para el desenvolvimiento inicial de las actividades a fin de preparar al alumnado adecuadamente para el logro de los objetivos relativos al desarrollo de esta asignatura.

7.2. Evaluación formativa.

Esta modalidad de evaluación permitirá identificar la evolución en el aprendizaje de los alumnos y el grado de impacto de la propuesta educativa que lleva a cabo la cátedra. Se evaluará:

- Participación del alumno en clase y en las actividades propuestas.
- Disposición y desempeño del alumno en la resolución de las actividades prácticas.
- Presentación en tiempo y forma de las prácticas propuestas.
- Capacidad de resolución y de análisis de los problemas de carácter teórico y práctico que se le presenten al alumno.

Las actividades que se aplicarán para la Evaluación Formativa implicarán el seguimiento del desarrollo de los trabajos prácticos y Talleres, y la presentación expositiva de los Talleres realizados y sus conclusiones. En este último aspecto cabe destacar la importancia de la aplicación de las técnicas, métodos, bibliotecas de softwares y estrategias de resolución planteadas en el desarrollo de las clases teóricas.

7.3. Evaluación parcial.

7.3.1. Programa y Cronograma de Evaluaciones Parciales.

Parciales	Tema	Meses				Fecha
		Abril	Mayo	Junio	Julio	
Evaluación Parcial 1	Concurrencia y paralelismo	---	3º Semana	---	---	16/05
Recuperatorio Eval.1		---	---	5º Semana	---	30/05
Evaluación Parcial 2	Concurrencia en sistemas distribuidos	---	---	3º Semana	---	13/06
Recuperatorio Eval. 2		---	---	---	5º Semana	30/06

7.3.2. Criterios de evaluación.

Estructura	Objetivo	Criterios de evaluación	Ponderación en la calificación
Evaluación Parcial 1			
Planteamientos teóricos y prácticos que exijan aplicar conceptos básicos referidos a concurrencia y paralelismo	Determinar si los alumnos han adquirido los conocimientos suficientes relativos a la concurrencia y el paralelismo en sistemas	Capacidad para reconocer aspectos de relevancia en la temática planteada. Habilidad para relacionar, contrastar y vincular los contenidos estudiados.	40%
Ejercicios de aplicación de técnicas y algoritmos empleados en la programación concurrente. Ejercicios de análisis de los modelos teóricos en los que se aplica concurrencia y paralelismo.	Determinar si los alumnos han adquirido las habilidades necesarias para reconocer los modelos teóricos que posibilitan concurrencia y paralelismo. Determinar si los alumnos conocen las técnicas y algoritmos empleados en la programación concurrente.	Habilidad para formalizar problemas. Capacidad para aplicar las teorías estudiadas. Capacidad para interpretar resultados y sacar conclusiones.	60%
Evaluación Parcial 2			
Ejercicio/s que permita/n aplicar problemas clásicos de programación concurrente y distribuida.	Determinar si los alumnos han adquirido las habilidades necesarias para reconocer y distinguir la ejecución de los procesos concurrentes y distribuidos.	Capacidad para resolver problemas propios de la concurrencia mediante el uso y aplicación de los problemas clásicos de productores y consumidores, lectores y escritores, etc.	60%
Ejercicio de análisis de un código en el que se aplica concurrencia.	Determinar si los alumnos han adquirido la habilidad de analizar, deducir, recomponer y/o corroborar un código fuente determinado en el que se aplica concurrencia	Habilidad para formalizar problemas. Capacidad para aplicar las teorías estudiadas. Capacidad para interpretar resultados y sacar conclusiones.	40%

7.3.3. Escala de Valoración, instancias de recuperación y condiciones de promoción.

La escala de valoración a emplear para las evaluaciones parciales y los recuperatorios será cuantitativa politómica. Escala: 1 al 100.

El puntaje mínimo para aprobar los parciales es de cincuenta (50) puntos, sobre una calificación máxima de 100.

Se otorga una sola recuperación en caso de desaprobación.

Actividad	Escala de Valoración	Recuperación
Trabajos Prácticos	Cualitativa dicotómica (aprobado – desaprobadado)	Recuperatorios de práctica
Talleres		
Evaluaciones Parciales	Cuantitativa politómica. Escala: 1 al 100. El puntaje mínimo para aprobar es de cincuenta (50) puntos, sobre una calificación máxima de 100.	

7.4. Evaluación integradora.

La evaluación integradora se llevará a cabo a través de un Trabajo Integrador Final en el que los alumnos deberán generar un informe respecto a las conclusiones a las que arribaron mediante las prácticas realizadas aplicando programación paralela y programación distribuida, incluyendo una comparación entre ambas.

7.5. Autoevaluación

La autoevaluación de la asignatura desde la perspectiva de los alumnos, se llevará a cabo a través de encuesta de respuestas cerradas implementada al finalizar el cursado.

La autoevaluación de la asignatura desde la perspectiva de los docentes se llevará a cabo a partir de los resultados obtenidos en las evaluaciones y condiciones finales de cursado.

7.6. Evaluación sumativa.

7.6.1. Condiciones para lograr la promoción sin Examen Final de la Asignatura.

No corresponde.

7.6.2. Condiciones para lograr la regularidad de la Asignatura.

- a) Asistir como mínimo al 75% del total de sesiones.
- b) Aprobar los Prácticos propuestos.
- c) Aprobar las evaluaciones parciales o los recuperatorios.
- d) Aprobar la Actividad de Formación Experimental.

7.7. Examen final.

La evaluación final será escrita u oral sobre los temas incluidos en la programación de la asignatura.

7.8. Examen libre.

Primera Etapa: Presentación, prueba y defensa de un planteamiento práctico. El trabajo deberá ser presentado en soporte digital e impreso, cumplimentando las pautas



establecidas en el enunciado, la presentación deberá realizarse al término del plazo de días hábiles indicado a partir de la entrega del enunciado.

Segunda Etapa: Aprobar una evaluación oral/escrita de contenidos teóricos del programa analítico.

Tercera Etapa: Aprobar una evaluación en computadora aplicando los contenidos teóricos y prácticos de la asignatura.

-----0ooo><ooo0-----

Ing.Méndez
Docente a cargo