

**UNIVERSIDAD NACIONAL  
DE  
SANTIAGO DEL ESTERO**

**FACULTAD DE CIENCIAS EXACTAS Y  
TECNOLOGÍAS**

**PLANIFICACIÓN ANUAL 2023**

ASIGNATURA: PROGRAMACIÓN AVANZADA

**LICENCIATURA EN SISTEMAS DE  
INFORMACIÓN**

**Plan de Estudio:**

**2011 / Innovación Curricular 2022**

**Equipo cátedra:**

**Profesor Adjunto:** Méndez Analía

**JTP:** Zarco Raquel

**JTP:** Fernández Reuter Beatriz



## PLANIFICACIÓN DE LA ASIGNATURA

### 1. IDENTIFICACIÓN:

1.1. **Nombre de Asignatura:** Programación Avanzada.

1.2. **Carrera:** Licenciatura en Sistemas de Información.

1.3. **Plan de Estudios:** 2011.

1.4. **Año académico:** 2023.

1.5. **Carácter:** Obligatoria.

### 1.6. Ubicación de la Asignatura en el Plan de Estudios

1.6.1. **Módulo – Año:** 9º Módulo – 5º Año.

1.6.2. **Trayecto al que pertenece la Asignatura/Obligación Curricular**

TRAYECTO	CARGA HORARIA PRESENCIAL
Ciencias Básicas y Específicas	----
Algoritmos y Lenguajes	90 horas
Arquitectura, Sistemas Operativos y Redes	----
Ingeniería de Software, Bases de Datos y Sistemas de Información	----
Aspectos Sociales y Profesionales	----
Otros contenidos	----
<b>CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR</b>	<b>90 horas</b>

Tabla 1: Carga horaria por trayecto

### 1.6.3. Correlativas

#### Plan 2011

#### 1.6.3.1. Anteriores:

Débiles: Sistemas Operativos Distribuidos.

Fuertes: Redes I,

#### 1.6.3.2. Posteriores:

Sistemas de Información III.



## Innovación Curricular 2022

### 1.6.3.3. Anteriores:

Débiles: Lenguajes de Programación y Compiladores, Sistemas Operativos Distribuidos.

Fuertes: Redes I, Programación Lógica y Funcional.

### 1.6.3.4. Posteriores:

Sistemas de Información III.

## 1.7. Carga horaria:

1.7.1. Carga horaria semanal total: 6 horas.

1.7.2. Carga horaria semanal destinada a la formación práctica: 4 horas.

1.7.3. Carga horaria total dedicada a las distintas actividades de formación práctica: 60 horas.

1.8. Ámbitos donde se desarrollan las actividades de formación práctica a las que se hace referencia en el punto anterior: aulas y laboratorio de computadoras.

1.9. Indique la cantidad de comisiones en las que se dicta la asignatura: 1 comisión.

## 2. PRESENTACIÓN

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

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

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 desarrollar destreza en la aplicación de lenguajes de programación en problemas de concurrencia y paralelismo.

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

- Arquitectura de redes, Tipos de redes y normas de comunicación, Sistemas Operativos de Redes y Redes de área local, adquiridos en la asignatura Redes I.



- 
- Paradigma Funcional, Lenguajes funcionales, Paradigma lógico y Lenguajes lógicos, estudiados en Programación Lógica y Funcional.
  - Lenguajes de Programación: Entidades y ligaduras, Generación y optimización de código intermedio y Generación de Código Objeto, estudiados en Lenguajes de Programación y Compiladores.
  - Sistemas Operativos Distribuidos: Estructuras. Comunicación. Sincronización y coordinación distribuida: exclusión mutua; control de concurrencia; gestión de interbloqueos, Sistemas de archivos distribuidos y Transacciones distribuidas, estudiados en Sistemas Operativos Distribuidos.

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

Posee:

- Los conocimientos básicos (lógico-matemáticos y computacionales) para una adecuada fundamentación teórica de su quehacer profesional específico.
- Profundos conocimientos sobre Algoritmos y Lenguajes de Programación.

Está capacitado para:

- Identificar, formular y resolver problemas de Informática.
- Gestar, diseñar y desarrollar proyectos de Informática.
- Gestionar, planificar, ejecutar y controlar proyectos de Informática.
- Utilizar técnicas y herramientas de aplicación en la Informática.
- Generar desarrollos tecnológicos y/o innovaciones tecnológicas.
- Comunicarse de manera efectiva en el ámbito profesional.
- Actuar con ética y responsabilidad social en el ámbito profesional.

## 2.4. Integración horizontal y vertical con otras asignaturas.

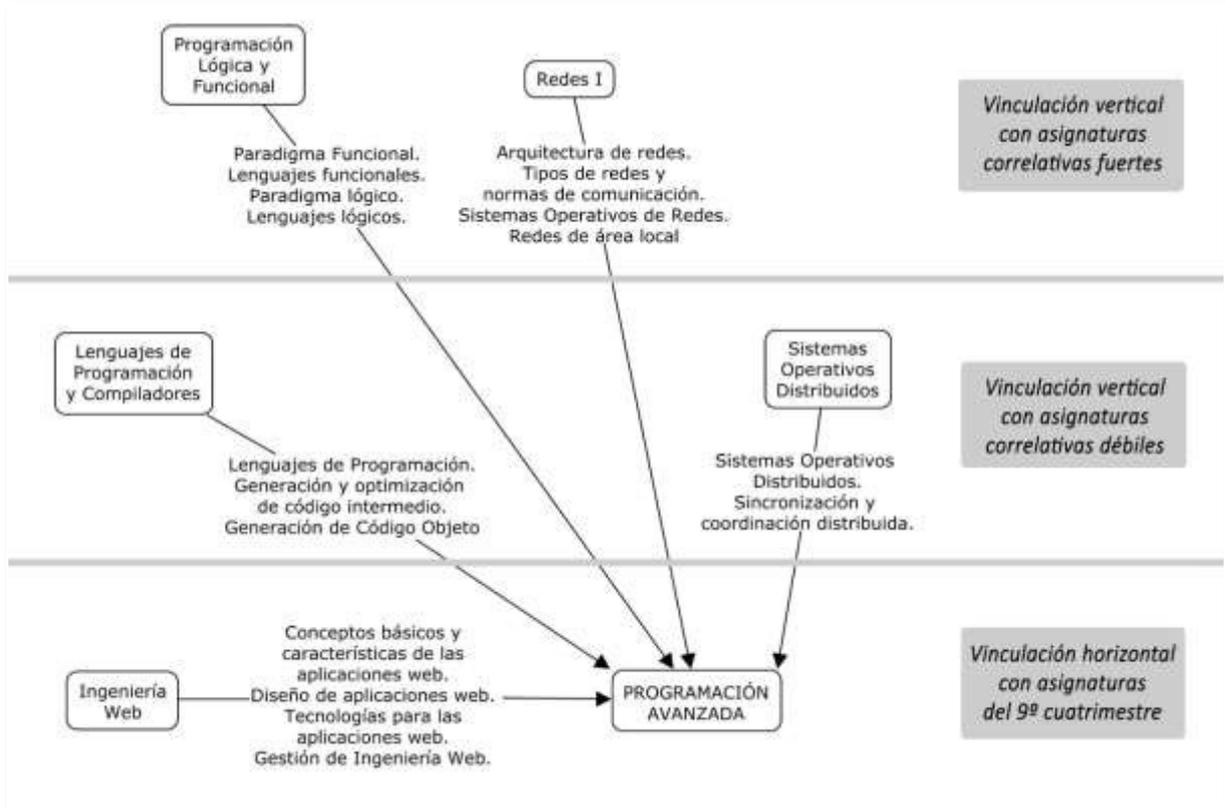


Figura 1. Integración horizontal y vertical de Programación Avanzada.



### 3. OBJETIVOS

#### 3.1. Objetivos Generales

- Desarrollar creativamente soluciones algorítmicas seleccionando criteriosamente la alternativa adecuada, aplicando distintos paradigmas de programación.
- Emplear estructuras de control y estructuras de datos en la resolución de problemas, aplicando distintos paradigmas de programación.
- Verificar algoritmos desarrollados en los distintos paradigmas de programación.
- Aplicar distintos lenguajes de programación como herramientas computacionales para la resolución de problemas.

#### 3.2. Objetivos Específicos

- Que el alumno desarrolle las siguientes competencias básicas:
  - Seleccionar y procesar la información necesaria para la resolución de problemas y verificar las soluciones encontradas.
- Que el alumno desarrolle las siguientes competencias específicas:
  - Identificar los fundamentos de la concurrencia y el paralelismo, los problemas inherentes a la programación concurrente, los aspectos de rendimiento de sistemas paralelos así como las medidas de prestaciones de algoritmos paralelos.
  - Reconocer las principales características de la especificación de la ejecución concurrente, la verificación de programas concurrentes, la metodología de la programación paralela y los esquemas algorítmicos paralelos.
  - Realizar la búsqueda creativa de soluciones algorítmicas y la selección criteriosa de la alternativa adecuada aplicando los mecanismos propios de los lenguajes de Programación Concurrente, tales como Java, Go, Haskell y/o Python.
  - Corregir y replantear resoluciones a problemas de concurrencia mediante la modificación o reutilización de código.
  - Programar algoritmos en un entorno de programación paralela bajo el modelo de paso de mensajes a través de un interfaz MPI (Message Passing Interface) estándar y bajo el modelo de memoria compartida utilizando la API OpenMP (Application Program Interface).
- Que el alumno desarrolle las siguientes competencias transversales:
  - Expresar de manera clara, comprensible y libre de errores de ortografía y gramática, en forma oral y escrita, los conceptos teóricos relacionándolos adecuadamente con su práctica.
  - Enfocar la comunicación oral y escrita en el intercambio y transmisión de los conocimientos.
  - Integrar participativamente el desarrollo de actividades de revisión teórica y actividades de resolución de problemas aportando sus habilidades y conocimientos al equipo de trabajo.



- Desarrollar los trabajos asignados de manera colaborativa favoreciendo la resolución de los problemas planteados y el desarrollo de la dinámica grupal.
- Identificar, plantear y resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.
- Desarrollar y apropiarse de nuevas habilidades y conocimientos como resultado del estudio y la práctica aplicando Programación Concurrente y Paralela.

#### **4. SELECCIÓN Y ORGANIZACIÓN DE CONTENIDOS**

##### **4.1. Contenidos mínimos establecidos en el Plan de Estudios para la Asignatura.**

Fundamentos de concurrencia y paralelismo. Programación concurrente. Algoritmos concurrentes. Problemas inherentes a la programación concurrente. Especificación de la ejecución concurrente. Verificación de programas concurrentes. Procesamiento paralelo. Modelos de programación paralela. Programación distribuida. Análisis de algoritmos paralelos. Metodología de la programación paralela. Esquemas algorítmicos paralelos. Rendimiento de los sistemas paralelos. Lenguajes y bibliotecas de programación concurrente y distribuida.

##### **4.2. Programa Sintético sobre la base de los contenidos mínimos**

###### **Unidad I: Programación concurrente.**

Fundamentos de concurrencia y paralelismo. Programación concurrente. Algoritmos concurrentes. Problemas inherentes a la programación concurrente. Especificación de la ejecución concurrente. Verificación de programas concurrentes.

###### **Unidad II: Programación paralela.**

Procesamiento paralelo. Modelos de programación paralela. Programación distribuida. Rendimiento de sistemas paralelos. Lenguajes y bibliotecas de programación concurrente y distribuida.

###### **Unidad III: Algoritmos paralelos.**

Análisis de algoritmos paralelos. Metodología de la programación paralela. Esquemas algorítmicos paralelos.

### 4.3. Articulación Temática de la Asignatura.

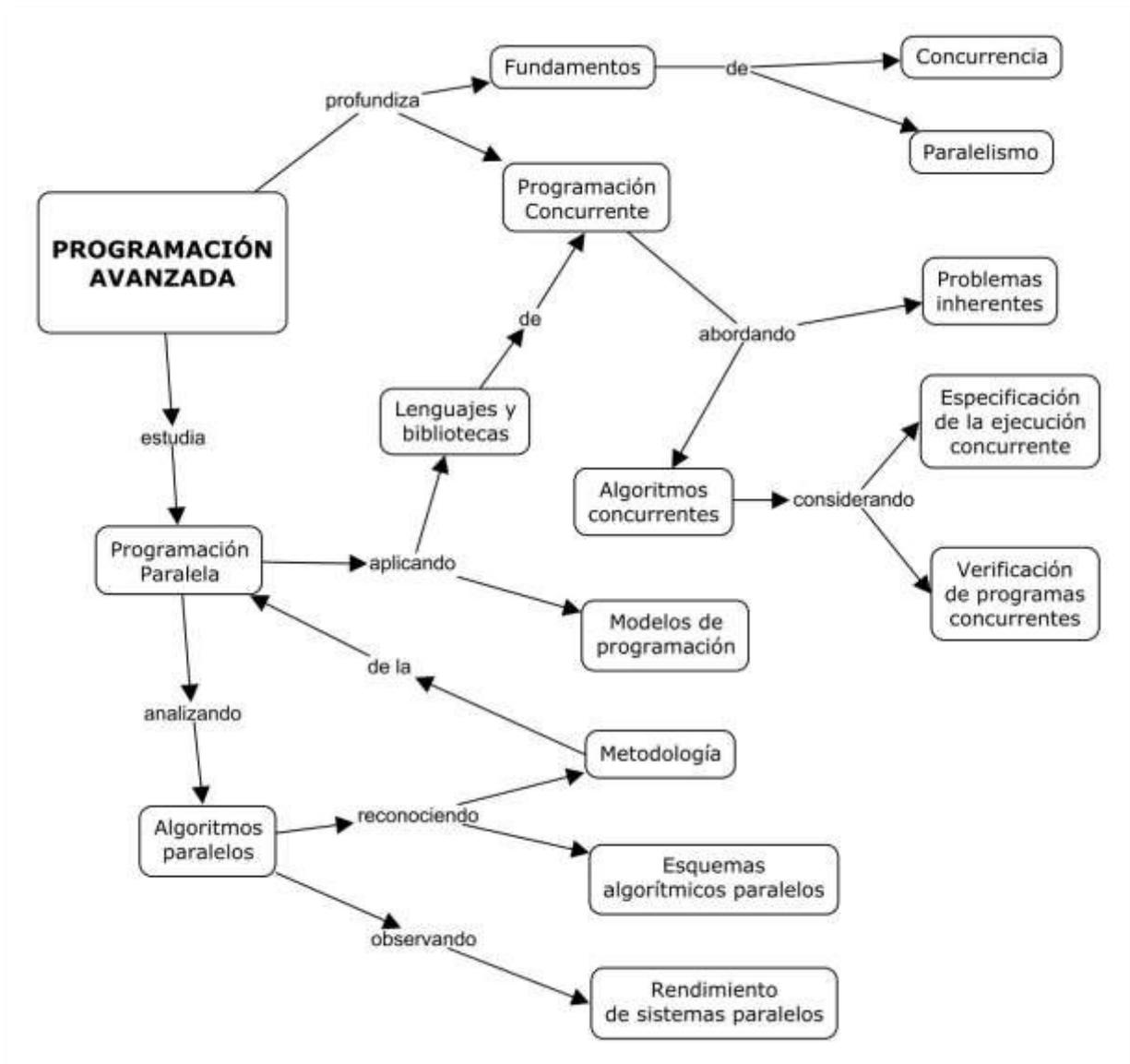


Figura 2. Articulación temática de la asignatura.



---

#### 4.4. Programa Analítico

##### **Unidad I: Programación concurrente.**

Fundamentos de concurrencia y paralelismo. Programación concurrente. Principios generales de la concurrencia. Formas de concurrencia. Algoritmos concurrentes. Problemas inherentes a la programación concurrente.

Especificación de la ejecución concurrente. Modelado de procesos concurrentes en FSP (Finite State Processes).

Verificación de programas concurrentes. Herramienta de verificación de sistemas concurrentes LTSA (Labelled Transition System Analyser). Análisis de algoritmos concurrentes.

Lenguajes de programación concurrente. Programación concurrente en Java.

##### **Unidad II: Programación paralela.**

Procesamiento paralelo. Clasificación de sistemas paralelos. Fuentes de paralelismo.

Modelos de programación paralela. Programación mediante paso de mensajes. Programación en memoria compartida.

Programación distribuida. Comunicación entre procesos distribuidos. Objetos distribuidos.

Rendimiento de sistemas paralelos. Aceleración y eficiencia. Escalabilidad. Complejidad algorítmica paralela.

Lenguajes y bibliotecas de programación concurrente y distribuida. Implementación de algoritmos con OpenMP.

##### **Unidad III: Algoritmos paralelos.**

Análisis de algoritmos paralelos. Tiempo de ejecución paralelo. Medidas de prestaciones de algoritmos paralelos. Escalabilidad.

Metodología de la programación paralela. Descomposición en tareas. Asignación de tareas. Estrategias para mejorar prestaciones y reducir costos. Diseño modular de programas paralelos.

Esquemas algorítmicos paralelos. Paralelismo de datos. Particionado de datos. Esquemas en árbol y grafo. Computación en pipeline. Computación síncrona. Esquema maestro/esclavo.



#### 4.5. Cronograma para el desarrollo de las Unidades Temáticas.

UNIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
Unidad 1: Programación concurrente	8	Semanas 1 a 4
Unidad 2: Programación paralela	12	Semanas 5 a 10
Unidad 3: Algoritmos paralelos	10	Semanas 11 a 15
<b>TOTAL</b>	<b>30</b>	

Tabla 2: Cronograma para el desarrollo teórico de las unidades temáticas

## 5. FORMACIÓN PRÁCTICA

### 5.1. Descripción de las actividades de formación práctica

Las actividades prácticas de la asignatura se desarrollan en Talleres organizados de la siguiente manera:

- Taller 1: Programación concurrente
- Taller 2: Aplicación de lenguajes de programación concurrentes.
- Taller 3: Bibliotecas de programación concurrente y distribuida.

ASPECTOS COMUNES A TODOS LOS TALLERES	
<b>OBJETIVOS</b>	<ul style="list-style-type: none"><li>• Integrar participativamente el desarrollo de actividades de resolución de problemas aportando sus habilidades y conocimientos al equipo de trabajo, desarrollándose de manera colaborativa para favorecer la resolución de los problemas planteados y el desarrollo de la dinámica grupal.</li><li>• Trabajar de forma autónoma desarrollando inquietud para la eficiencia y el rigor, cumpliendo con las modalidades y tiempos de presentación de las actividades prácticas, reaccionando positivamente frente al error e identificando y verificando nuevas estrategias para alcanzar el logro exigido a fin de completar el desarrollo de la tarea recibida.</li><li>• Participar en las actividades prácticas y en las tareas de aula con respeto y con responsabilidad ética, tanto en grupo como individualmente.</li></ul>
<b>METODOLOGÍA</b>	El desarrollo de cada Taller será de carácter teórico y práctico, con actividades prácticas guiadas, resolución de ejercicios de programación en aula, y el desarrollo de proyectos domiciliarios. Las clases se desarrollarán en el Laboratorio de Informática.
<b>EVALUACIÓN</b>	Para aprobar cada Taller el alumno deberá completar las actividades prácticas propuestas, individuales y grupales, y luego presentar y defender satisfactoriamente proyecto de programación. La escala de valoración a emplear será cualitativa dicotómica (aprobado – desaprobado).
<b>CRITERIOS DE EVALUACIÓN Y APROBACIÓN</b>	Presentación en tiempo y forma de las tareas individuales y grupales, de acuerdo a lo solicitado en cada actividad planteada, desarrollándose con respeto y responsabilidad ética y participando colaborativamente en la resolución de los problemas planteados y el desarrollo de la dinámica grupal.

Tabla 3: Aspectos comunes de los Talleres



---

A continuación se describen los aspectos particulares de cada uno de los Talleres.

## TALLER 1: PROGRAMACIÓN CONCURRENTE

### Contenidos

Programas concurrentes. Especificación de la ejecución concurrente. Sistemas de transición de estados etiquetados. Modelado de procesos concurrentes usando el lenguaje FSP (Finite State Processes). Semántica de procesos. Prefijos de acciones. Recursión. Elección. Procesos y acciones indexados. Acciones con guardas. Etiquetas de procesos. Ocultamiento. Verificación de programas concurrentes. Herramienta de verificación de sistemas concurrentes LTSA (Labelled Transition System Analyser).

### Objetivos

- Modelar procesos concurrentes especificando la concurrencia en lenguaje FSP (Finite State Processes).
- Realizar la verificación de programas concurrentes aplicando LTSA (Labelled Transition System Analyser).

### Criterios de evaluación y aprobación

- Correcta aplicación del modelado de los procesos concurrentes.
- Adecuada ejecución práctica de la verificación de procesos concurrentes.
- Elaboración de conclusiones y observaciones sobre la práctica realizada, debidamente fundamentadas.

### Instrumentos para la actividad

Para desarrollar esta actividad de formación experimental es necesario contar con los siguientes recursos:

- Espacio de la asignatura en la plataforma CUV (Centro Universitario Virtual).
- Laboratorio de computadoras con la instalación de LTSA (Labelled Transition System Analyser).

## TALLER 2: LENGUAJES DE PROGRAMACIÓN CONCURRENTE.

### Contenidos

Implementación de aplicaciones utilizando Java concurrente. Hilos en Java. Clase Thread. Interface Runnable. Sincronización de hilos en Java. Métodos sincronizados. Aplicación de los lenguajes Java, Go, Haskell y/o Python, entre otros, en casos de estudio de naturaleza concurrente. Algoritmos concurrentes.

### Objetivos.

- Construir aplicaciones concurrentes e implementar algoritmos concurrentes mediante la utilización de lenguajes tales como Java, Go, Haskell o Python, entre otros.



- Corregir y replantear resoluciones a problemas de concurrencia mediante la modificación o reutilización de código.

#### **Criterios de evaluación y aprobación**

- Adecuada aplicación de la concurrencia en el desarrollo de programas utilizando lenguajes de programación.
- Correcta implementación y verificación de algoritmos concurrentes.

#### **Instrumentos para la actividad**

Para desarrollar esta actividad de formación experimental es necesario contar con los siguientes recursos:

- Espacio de la asignatura en la plataforma CUV (Centro Universitario Virtual).
- Laboratorio de computadoras con la instalación de los lenguajes de programación aplicados.

### **TALLER 3: BIBLIOTECAS DE PROGRAMACIÓN PARALELA Y DISTRIBUIDA**

#### **Contenidos**

Programación mediante paso de mensajes. Modelo de ejecución de un programa paralelo de paso de mensajes. Estructura de un programa de paso de mensajes. El estándar MPI de paso de mensajes. Modelo de ejecución de un programa paralelo en memoria compartida. El estándar OpenMP para programación en memoria compartida. Implementación de algoritmos utilizando MPI y OpenMP.

#### **Objetivos.**

- Programar algoritmos en un entorno de programación paralela bajo el modelo de paso de mensajes a través de un interfaz MPI (Message Passing Interface) estándar.
- Programar algoritmos básicos con OpenMP a través de ejecuciones paralelas en máquinas multinúcleo.

#### **Criterios de evaluación y aprobación**

- Adecuada instalación y configuración de los entornos para la ejecución de programas paralelos con OpenMP y MPI.
- Correcta observación y comparación de los resultados obtenidos a partir de la ejecución de programas paralelos con MPI y OpenMP.

#### **Instrumentos para la actividad**

Para desarrollar esta actividad de formación experimental es necesario contar con los siguientes recursos:

- Espacio de la asignatura en la plataforma CUV (Centro Universitario Virtual).
- Laboratorio de computadoras con la instalación del Lenguaje C/C++.
- Biblioteca MPI (Message Passing Interface, Interfaz de Paso de Mensajes), OpenMPI.



## 5.2. Formación en Ejes Transversales

En la Tabla 3 se relacionan cada uno de los Ejes Transversales de Formación con las actividades y los resultados de aprendizaje esperados.

Tabla 4: Formación en Ejes Transversales

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GRADO DE PROFUNDIDAD EN EL TRATAMIENTO
Identificación, formulación y resolución de problemas de informática	• Talleres	<ul style="list-style-type: none"><li>• Identificar, plantear y resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.</li><li>• Seleccionar y procesar la información necesaria para la resolución de problemas y verificar las soluciones encontradas.</li></ul>	Alto
Concepción, diseño y desarrollo de proyectos de informática		<ul style="list-style-type: none"><li>• Desarrollar proyectos de programación aplicando lenguajes concurrentes para la resolución de algoritmos.</li></ul>	Alto
Gestión, planificación, ejecución y control de proyectos de informática		<ul style="list-style-type: none"><li>• Diseñar, codificar, ejecutar, depurar e interpretar programas utilizando lenguajes de programación concurrentes, tales como Java, Haskell, Python, Go entre otros.</li></ul>	Alto
Utilización de técnicas y herramientas de aplicación en la informática		<ul style="list-style-type: none"><li>• Aplicar los principios de la Programación Concurrente identificando los fundamentos de la concurrencia y el paralelismo, los problemas inherentes a la programación concurrente, los aspectos de rendimiento de sistemas paralelos así como las medidas de prestaciones de algoritmos paralelos.</li><li>• Programar algoritmos en un entorno de programación paralela bajo el modelo de paso de mensajes a través de un interfaz MPI (Message Passing Interface) estándar y bajo el modelo de memoria compartida utilizando la API OpenMP (Application Program Interface).</li></ul>	Alto
Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.		<ul style="list-style-type: none"><li>• Diseñar, codificar, ejecutar, depurar e interpretar programas utilizando lenguajes de programación concurrente.</li></ul>	Alto



Tabla 4: Formación en Ejes Transversales (continuación)

EJE	ACTIVIDADES	RESULTADOS DE APRENDIZAJE	GRADO DE PROFUNDIDAD EN EL TRATAMIENTO
Fundamentos para el desempeño en equipos de trabajo	<ul style="list-style-type: none"><li>• Actividades de aula.</li><li>• Actividades aplicativas.</li><li>• Talleres.</li></ul>	<ul style="list-style-type: none"><li>• Integrar participativamente el desarrollo de actividades de revisión teórica y actividades de resolución de problemas aportando sus habilidades y conocimientos al equipo de trabajo.</li><li>• Desarrollar los trabajos asignados de manera colaborativa favoreciendo la resolución de los problemas planteados y el desarrollo de la dinámica grupal.</li></ul>	Alto
Fundamentos para la comunicación efectiva	<ul style="list-style-type: none"><li>• Actividades de aula.</li><li>• Actividades aplicativas.</li></ul>	<ul style="list-style-type: none"><li>• Expresar de manera clara, comprensible y libre de errores de ortografía y gramática, en forma oral y escrita, los conceptos teóricos relacionándolos adecuadamente con su práctica.</li><li>• Enfocar la comunicación oral y escrita en el intercambio y transmisión de los conocimientos.</li></ul>	Alto
Fundamentos para la acción ética y responsable.	<ul style="list-style-type: none"><li>• Actividades de aula.</li><li>• Actividades aplicativas.</li><li>• Talleres.</li></ul>	<ul style="list-style-type: none"><li>• Trabajar de forma autónoma desarrollando inquietud para la eficiencia y el rigor, cumpliendo con las modalidades y tiempos de presentación de las actividades teóricas y prácticas.</li><li>• Reaccionar frente al error identificando y verificando nuevas estrategias para alcanzar el logro exigido a fin de completar el desarrollo de la tarea recibida.</li><li>• Realizar la búsqueda creativa de soluciones algorítmicas en grupo e individualmente, con respeto y con responsabilidad ética.</li></ul>	Medio
Fundamentos para el aprendizaje continuo	<ul style="list-style-type: none"><li>• Talleres.</li></ul>	<ul style="list-style-type: none"><li>• Desarrollar y apropiarse de nuevas habilidades y conocimientos como resultado del estudio y la práctica aplicando Programación Concurrente y Paralela.</li></ul>	Alto

### 5.3. Cronograma de formación práctica

ACTIVIDAD	CARGA HORARIA	CRONOGRAMA DE DICTADO
Taller 1: Programación concurrente	12	Semanas 1 a 3
Taller 2: Lenguajes de Programación concurrente	24	Semanas 4 a 9
Taller 3: Bibliotecas de programación paralela y distribuida	24	Semanas 10 a 15
<b>TOTAL</b>	<b>60</b>	

Tabla 5: Cronograma para el desarrollo de las Actividades Prácticas



## 6. BIBLIOGRAFÍA.

TÍTULO	AUTOR/ES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Introducción a la computación paralela	Aguilar, J., & Leiss, E.	Venezuela: Gráficas Quinteto	ResearchGate	2004
	<b>ENLACE DE ACCESO</b>	<a href="https://www.researchgate.net/publication/267367623">https://www.researchgate.net/publication/267367623</a>		
Sistemas de tiempo real y Lenguajes de Programación	Alan Burns y Andy Wellings	Pearson Education	1 en Centro de Documentación	2003
Introducción a la Programación Paralela	Almeida - Giménez - Mantas - Vidal	Paraninfo	1 en Centro de Documentación	2008
Ejercicios de programación paralela con OpenMP y MPI	Alvarruiz, F. & Román, J. E.	Editorial de la Universidad Politécnica de Valencia	Biblioteca Digital eLibro	2018
	<b>ENLACE DE ACCESO</b>	<a href="https://elibro.net/es/lc/unsebiblio/titulos/57462">https://elibro.net/es/lc/unsebiblio/titulos/57462</a>		
Aplicaciones distribuidas en Java con tecnología RMI	Caballé, S. & Xhafa, F.	Delta Publicaciones	Biblioteca Digital eLibro	2008
	<b>ENLACE DE ACCESO</b>	<a href="https://elibro.net/es/lc/unsebiblio/titulos/168236">https://elibro.net/es/lc/unsebiblio/titulos/168236</a>		
Programación Concurrente en Java	Doug Lea	Addison Wesley	1 en Centro de Documentación	2001
Programación Concurrente y Tiempo Real	Fernández, D. V., Morcillo, C. G., & Jiménez, J. A. A.	Edita David Vallejo Fernández	Biblioteca de retro.com.es	2016
	<b>ENLACE DE ACCESO</b>	<a href="https://www.retro.com.es/_media/programacion/ada/programacion-concurrente-y-tiempo-real.pdf">https://www.retro.com.es/_media/programacion/ada/programacion-concurrente-y-tiempo-real.pdf</a>		
Programación orientada a objetos usando Java	Flórez Fernández, H. A.	Ecoe Ediciones	Biblioteca Digital eLibro	2012
	<b>ENLACE DE ACCESO</b>	<a href="https://elibro.net/es/lc/unsebiblio/titulos/69236">https://elibro.net/es/lc/unsebiblio/titulos/69236</a>		
Java 2: manual de usuario y tutorial (5a. ed.)	Froufe Quintas, A.	RA-MA Editorial	Biblioteca Digital eLibro	2013
	<b>ENLACE DE ACCESO</b>	<a href="https://elibro.net/es/lc/unsebiblio/titulos/106400">https://elibro.net/es/lc/unsebiblio/titulos/106400</a>		

Tabla 6: Bibliografía



Tabla 6: Bibliografía (continuación)

TÍTULO	AUTOR/ES	EDITORIAL	EJEMPLARES DISPONIBLES	AÑO DE EDICIÓN
Concepts, Techniques, and Models of Computer Programming	Haridi, S., & Van-Roy, P	Cambridge: MIT Press	Academia.edu	2004
	<b>ENLACE DE ACCESO</b>	<a href="https://www.academia.edu/34862716/Concepts_Techniques_and_Models_of_Computer_Programming">https://www.academia.edu/34862716/Concepts_Techniques_and_Models_of_Computer_Programming</a>		
Arquitecturas de computadores avanzadas	Jiménez González, D., Guim Bernat, F., & Rodero Castro, I.	Universitat Oberta de Catalunya	Universitat Oberta de Catalunya. Repositorio digital.	2015
	<b>ENLACE DE ACCESO</b>	<a href="http://hdl.handle.net/10609/79549">http://hdl.handle.net/10609/79549</a>		
Programación en Java 2 - Algoritmos, estructuras de datos y Programación Orientada a Objetos	Joyanes Aguilar - Zahonero Martinez.	Mc Graw Hill.	1 en Centro de Documentación	2002
State models and java programs.	Magee, J., & Kramer, J.	Hoboken: wiley	City, University of London Staff Personal Pages	1999
	<b>ENLACE DE ACCESO</b>	<a href="http://www.staff.city.ac.uk/c.kloukinas/concurrency/concurrency-handouts.pdf">http://www.staff.city.ac.uk/c.kloukinas/concurrency/concurrency-handouts.pdf</a>		
Concurrencia y sistemas distribuidos..	Muñoz Escóí, F. D.	Editorial de la Universidad Politécnica de Valencia	Biblioteca Digital eLibro	2013
	<b>ENLACE DE ACCESO</b>	<a href="https://elibro.net/es/lc/unsebiblio/titulos/57365">https://elibro.net/es/lc/unsebiblio/titulos/57365</a>		
Computación de altas prestaciones	Rodero Castro, I., & Guim Bernat, F.	Universitat Oberta de Catalunya	Universitat Oberta de Catalunya. Repositorio digital.	2012
	<b>ENLACE DE ACCESO</b>	<a href="http://hdl.handle.net/10609/63485">http://hdl.handle.net/10609/63485</a>		
Java 17: programación avanzada	Vegas Gertrudix, J. M.	RA-MA Editorial	Biblioteca Digital eLibro	2021
	<b>ENLACE DE ACCESO</b>	<a href="https://elibro.net/es/lc/unsebiblio/titulos/222668">https://elibro.net/es/lc/unsebiblio/titulos/222668</a>		

## 7. ESTRATEGIAS METODOLÓGICAS.

### 7.1.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 mediante Actividades Aplicativas



---

y actividades de Taller.

Las actividades de estudio tanto prácticas como de tipo teórico serán apoyadas y desarrolladas mediante el aula virtual de la asignatura alojada en la plataforma Moodle del Centro Universitario Virtual (CUV - <http://cuv.unse.edu.ar>). Mediante esta plataforma se acercarán al estudiante diferentes recursos y se utilizarán herramientas como cuestionarios evaluativos y de diagnóstico, así como la presentación digital de las actividades y proyectos propuestos en las prácticas. Cada contenido desarrollado es mediado para su mejor comprensión y con el fin de propiciar el diálogo y la discusión.

Mediante Actividades de Aula y Actividades Aplicativas se propone al alumno la vinculación entre teoría y práctica. En estas actividades se fomenta el desarrollo de habilidades para la comunicación efectiva mediante la expresión oral y/o escrita de los conceptos teóricos relacionándolos con la práctica.

Para cada una de las unidades temáticas de la asignatura, 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 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 estudiados, a través de casos de estudio.

Durante las clases se completarán las actividades requeridas para el abordaje de cada tema y se organizarán tareas individuales y grupales colaborativas que favorezcan la apropiación del estudio, tanto mediante Actividades Aplicativas en las clases teóricas como mediante actividades de Taller en las clases de práctica. De este modo, los conceptos teóricos y los aspectos metodológicos de la práctica requerida son abordados tanto en los encuentros teóricos como en los prácticos.

En el desarrollo de las actividades de la asignatura se promueve por parte del alumno la integración participativa en donde aporte al grupo de estudio tanto sus habilidades como sus conocimientos.

## 7.2. Mecanismos para la integración de docentes

Se realizarán actividades de revisión y coordinación en el Trayecto Algoritmos y Lenguajes en el marco de la Comisión de Seguimiento del Plan de Estudios de la carrera de Licenciatura en Sistemas de Información. Además, se efectuarán reuniones periódicas con las asignaturas del Trayecto con el fin de aunar tareas conjuntas de integración.

La coordinación entre las asignaturas se realiza mediante encuentros anuales en los que se revisan los logros alcanzados en la comprensión y aplicación de los paradigmas de programación estudiados y las actividades prácticas.

La asignatura Sistemas Operativos Distribuidos integra el módulo 8° del 4° año de la carrera de Licenciatura en Sistemas de Información y ahonda exhaustivamente los conceptos de Sincronización y coordinación distribuida: exclusión mutua; control de concurrencia, Manejo de recursos: asignación y protección. La integración con esta asignatura se lleva a cabo



mediante reuniones en las que se intercambia información respecto de las actividades cumplidas y los logros alcanzados por los alumnos, a fin de coordinar las acciones necesarias.

### 7.3. Recursos Didácticos

Se utilizarán diversos recursos didácticos a fin de lograr los objetivos propuestos, a saber:

- Ambiente integrado de desarrollo (IDE) para la programación en lenguaje Java y en lenguaje C/C++.
- Biblioteca MPI (Message Passing Interface, Interfaz de Paso de Mensajes), OpenMPI.
- LTSA - Analizador de sistemas de transición etiquetados.
- Para desarrollar aplicaciones en Java se necesita la herramienta de desarrollo Java Development Kit (JDK). JDK incluye Java Runtime Environment, el compilador Java y las API de Java.
- En las clases teóricas y en las clases prácticas se utilizarán enunciados tomados de las actividades prácticas propuestas, diapositivas y videos elaborados por los docentes de la asignatura, demos, tutoriales, etc.
- Se contará con el espacio de la asignatura en la plataforma Moodle del Centro Universitario Virtual (CUV), que permitirá al alumno acceder a los recursos digitales, apoyo de prácticas, etc. También brindará a los alumnos un canal de comunicación permanente donde podrán acceder a información actualizada de la asignatura: fechas de evaluaciones, resultados de parciales, condición final de la cursada, etc.

## 8. EVALUACIÓN

### 8.1. Evaluación Diagnóstica

Teniendo en cuenta que la evaluación diagnóstica no sólo es una estimación, sino que tiene como propósito contribuir al aprendizaje, se llevará a cabo una única evaluación diagnóstica, al comienzo de las clases, cuya finalidad será determinar el nivel de conocimientos y habilidades previas que permitan encarar el aprendizaje de la asignatura.

Los contenidos sobre los que se evaluará serán: Paradigma Funcional, Paradigma Lógico, Arquitectura de Redes, Generación y optimización de código, Sincronización y coordinación distribuida y Sistemas Operativos Distribuidos. Se utilizará la herramienta cuestionario de la plataforma Moodle del CUV.

La evaluación diagnóstica será especialmente diseñada, individual, y objetiva.

Se hará una prueba de opción múltiple para que el alumno marque la opción correcta. El nivel de calificación será cuantitativa politómica. Escala: 1 al 10.

### 8.2. 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 actividades teóricas y de taller propuestas.
- Capacidad de resolución y de análisis de los problemas de carácter teórico y de taller que se le presenten al alumno.

Las actividades que se aplicarán para la Evaluación Formativa implicarán el seguimiento del desarrollo de los Talleres y la presentación 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.

La evaluación formativa de los contenidos teóricos se llevará a cabo mediante la presentación y exposición de las Actividades Aplicativas.

En cuanto a las competencias relacionadas a la práctica, la evaluación formativa se llevará adelante aplicando la técnica de casos y se estructurará mediante la entrega regulada de actividades de los talleres.

Mediante la evaluación formativa se motivará, reforzará y proporcionará ayuda a los estudiantes, reconociendo y observando sus aprendizajes, dificultades y posibilidades.

### 8.3. Evaluación Parcial

#### 8.3.1. Programa de Evaluaciones Parciales

EVALUACIÓN	TEMA	MODALIDAD	SEM	FECHA	DEVOLUCIÓN DE RESULTADO
Parcial 1	Programación concurrente y paralela	Individual. Cuestionario multiple choice mediante plataforma del CUV.	8	08/05	11/05
Recuperatorio 1			10	22/05	26/05
Parcial 2	Algoritmos paralelos		13	12/06	15/06
Recuperatorio 2			14	19/06	22/06

Tabla 7: Programa de Evaluaciones Parciales



### 8.3.2. Criterios de Evaluación

ESTRUCTURA	OBJETIVO	CRITERIOS DE EVALUACIÓN
Preguntas teóricas de los siguientes tipos: <ul style="list-style-type: none"><li>• Verdadero-Falso.</li><li>• Respuesta múltiple.</li><li>• Completar gráficas.</li><li>• Completar frases.</li><li>• Relacionar conceptos con sus definiciones.</li></ul>	<b>Parcial 1 / Recuperatorio 1</b> <ul style="list-style-type: none"><li>• Identificar los fundamentos de la concurrencia y el paralelismo así como los problemas inherentes a la programación concurrente.</li><li>• Reconocer las principales características de la especificación de la ejecución concurrente y la verificación de programas concurrentes.</li><li>• Distinguir las diferencias entre los modelos de programación paralela.</li></ul>	<ul style="list-style-type: none"><li>• Capacidad para reconocer aspectos de relevancia en la temática planteada.</li><li>• Habilidad para relacionar, contrastar y vincular los contenidos estudiados.</li></ul>
	<b>Parcial 2 / Recuperatorio 2</b> <ul style="list-style-type: none"><li>• Identificar los aspectos de rendimiento de sistemas paralelos así como las medidas de prestaciones de algoritmos paralelos.</li><li>• Reconocer las principales características de la metodología de la programación paralela y de los esquemas algorítmicos paralelos.</li></ul>	

Tabla 8: Criterios de Evaluación de las Evaluaciones Parciales

### 8.3.3. Escala de Valoración

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.

### 8.4. Evaluación Integradora

No se prevé.

### 8.5. Evaluación Sumativa

#### 8.5.1. Condiciones para lograr la promoción sin Examen Final de la Asignatura. (Rige la Resolución HCD N° 135/00)

No se prevé.

#### 8.5.2. Condiciones para lograr la Regularidad de la Asignatura.

Para regularizar la asignatura el alumno deberá:

- Aprobar las evaluaciones parciales o sus recuperatorios.
- Aprobar los tres (3) Talleres de la asignatura.



---

### 8.6. Examen Final

En el examen final los alumnos serán evaluados sobre los contenidos teóricos previstos en el programa de la asignatura. El examen podrá ser oral o escrito e individual.

### 8.7. Examen Libre

Para el examen Libre el alumno deberá aprobar las dos (2) instancias que se detallan a continuación, siendo cada una de ellas eliminatorias:

#### **Etapas 1.**

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.

#### **Etapas 2.**

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

*Méndez Analía*

.....  
*Profesor responsable de Asignatura*