

**UNIVERSIDAD NACIONAL
DE
SANTIAGO DEL ESTERO**

**FACULTAD DE CIENCIAS EXACTAS Y
TECNOLOGÍAS**

PLANIFICACIÓN ANUAL 2024

ASIGNATURA: PROGRAMACIÓN AVANZADA

**LICENCIATURA EN SISTEMAS DE
INFORMACIÓN**

Plan de Estudio: 2011

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: 2024.

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 | ----- |
| CARGA HORARIA TOTAL DE LA ACTIVIDAD CURRICULAR | 90 horas |

Tabla 1: Carga horaria por trayecto

1.6.3. Correlativas

1.6.3.1. Anteriores:

Débiles: Sistemas Operativos Distribuidos.

Fuertes: Redes II.

1.6.3.2. Posteriores:

Débil: 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.

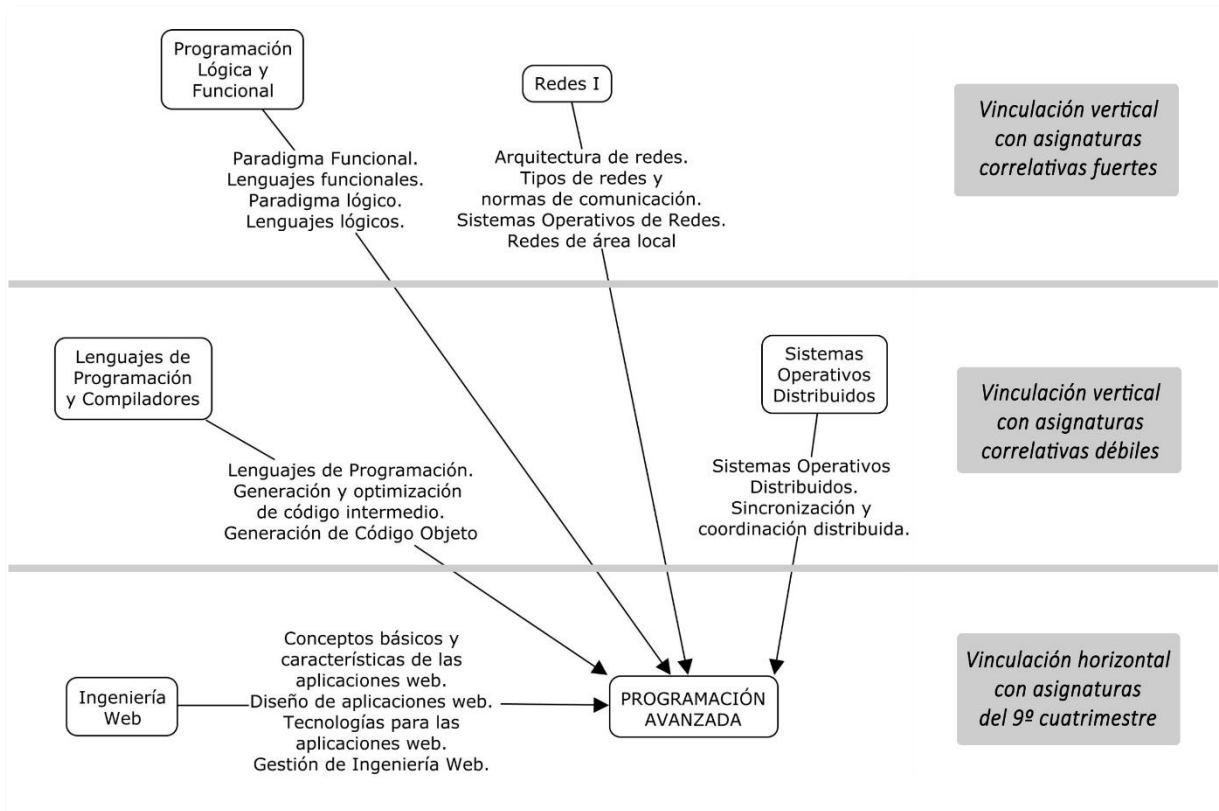


Figura 1. Integración horizontal y vertical de Programación Avanzada.



3. OBJETIVOS

- Que el alumno desarrolle las siguientes competencias específicas:
 - C1. Analizar problemas y formular soluciones algorítmicas utilizando la modelación de procesos concurrentes.
 - C2. Desarrollar programas utilizando la programación concurrente.
 - C3. Ejecutar y verificar proyectos de programación aplicando los principios de programación concurrente.
 - C4. Utilizar herramientas modernas de desarrollo de software que faciliten el desarrollo de programas concurrentes.
 - C5. Capacidad para desarrollar proyectos aplicando la programación concurrente, innovando en el diseño y el desarrollo.
- Que el alumno desarrolle las siguientes competencias transversales:
 - C6. Capacidad para colaborar en el desarrollo de actividades grupales de integración entre teoría y práctica considerando la programación concurrente y paralela.
 - C7. Capacidad para comunicar de manera clara y precisa conceptos, procedimientos y soluciones relacionados con la programación concurrente y paralela.
 - C8. Capacidad para identificar y evaluar los impactos éticos asociados con su práctica de programación concurrente, aplicando un pensamiento crítico y reflexivo.
 - C9. Capacidad para desarrollar un aprendizaje compartido y continuo en el ámbito de la programación concurrente tomando iniciativa para adquirir nuevos conocimientos y habilidades de manera autónoma.

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.

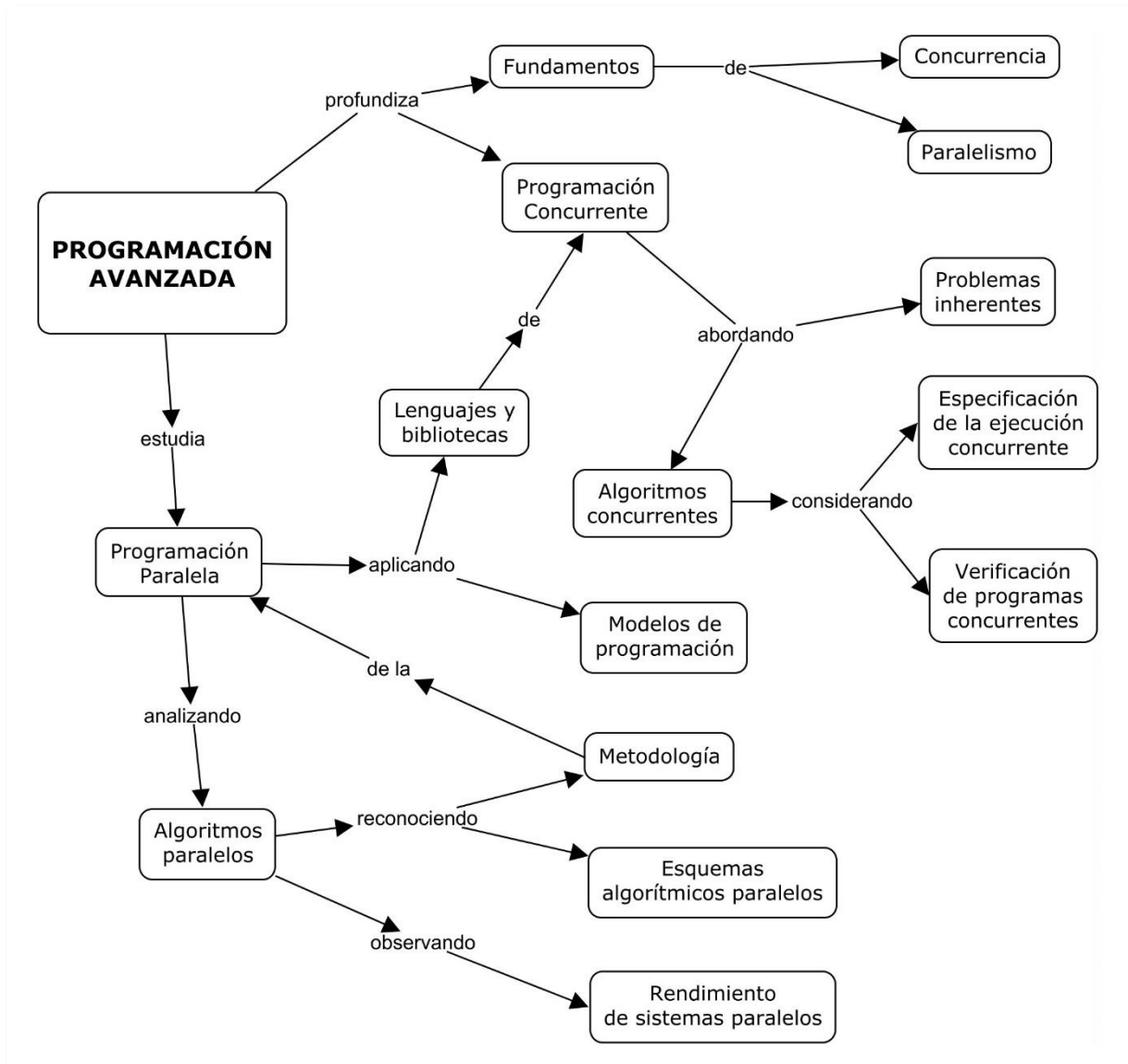
Procesamiento paralelo. Modelos de programación paralela. Programación distribuida. Rendimiento de sistemas paralelos. Lenguajes y bibliotecas de programación concurrente y distribuida.

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 |
| TOTAL | 30 | |

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 concurrentes.
- Taller 3: Bibliotecas de programación concurrente y distribuida.

Aspectos comunes a todos los Talleres

Metodología

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.

Evaluación

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).

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).

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: APLICACIÓN DE LENGUAJES CONCURRENTES.

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.

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.

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 3: Formación en Ejes Transversales

| EJE | ACTIVIDADES | RESULTADOS DE APRENDIZAJE | GPT |
|---|---|---|------|
| Identificación, formulación y resolución de problemas de informática | A1. Modelar procesos concurrentes especificando la concurrencia en lenguaje FSP (Finite State Processes). | RA.1. Desarrolla especificaciones detalladas y precisas de procesos concurrentes asegurando que las soluciones sean comprensibles y replicables usando lenguaje FSP (Finite State Processes). | Alto |
| Concepción, diseño y desarrollo de proyectos de informática | A2. Construir aplicaciones concurrentes e implementar algoritmos utilizando lenguajes concurrentes. | RA.2. Implementa correctamente soluciones algorítmicas utilizando programación concurrente. | Alto |
| Gestión, planificación, ejecución y control de proyectos de informática | A3. Aplicar las técnicas de programación concurrente en un entorno de proyecto real. | RA. 3. Ejecuta y verifica proyectos de programación concurrente gestionando múltiples hilos/procesos | Alto |
| Utilización de técnicas y herramientas de aplicación en la informática | A4. Usar entornos de desarrollo integrado (IDEs), para la codificación, depuración y prueba de programas concurrentes. | RA.4. Depura programas concurrentes en un IDE, identificando y resolviendo problemas relacionados con la concurrencia, como condiciones de carrera y bloqueos, utilizando las herramientas de depuración disponibles. | Alto |
| Generación de desarrollos tecnológicos y/o innovaciones tecnológicas. | A5. Aplicar los conocimientos de concurrencia en un entorno de desarrollo real, promoviendo la creatividad y la innovación en el diseño y la implementación de la solución. | RA. 5. Demuestra capacidad para innovar en el diseño y desarrollo de aplicaciones de software, aplicando la concurrencia y paralelismo. | Alto |
| Fundamentos para el desempeño en equipos de trabajo | A6. Colaborar con su equipo en el desarrollo de tareas de integración entre teoría y práctica considerando la programación concurrente y paralela. | RA.6. Colabora participativamente en el desarrollo de actividades de revisión teórica y actividades de resolución de problemas aportando al equipo de trabajo sus habilidades y conocimientos sobre la programación concurrente y paralela. | Alto |
| Fundamentos para la comunicación efectiva | A7. Elaborar y exponer presentaciones de análisis y revisión sobre soluciones de programación concurrente y paralela. | RA.7. Realiza presentaciones escritas y orales en las que expone y defiende soluciones de programación concurrente y paralela, demostrando dominio del contenido y capacidad de respuesta a preguntas técnicas. | Alto |



Tabla 3: Formación en Ejes Transversales

| EJE | ACTIVIDADES | RESULTADOS DE APRENDIZAJE | GPT |
|---|---|--|-------|
| Fundamentos para la acción ética y responsable. | A8. Evaluar las decisiones tomadas en el diseño e implementación de programas aplicando la programación concurrente y paralela, analizando su alineación con principios éticos. | RA.8. Establece conclusiones sobre su práctica de programación, considerando los impactos éticos de las soluciones desarrolladas aplicando programación concurrente y paralela. | Medio |
| Fundamentos para el aprendizaje continuo | A9. Analizar, debatir y discutir sobre temas relacionados con la programación concurrente y paralela colaborando en la resolución de problemas y fomentando un aprendizaje compartido y continuo. | RA.9. Desarrolla habilidades de aprendizaje compartido y continuo en el campo de la programación concurrente y paralela, identificando y adquiriendo nuevos conocimientos de manera autónoma y de manera colaborativa. | Alto |

(*) GPT: Grado de Profundidad en el Tratamiento.

5.3. Cronograma de formación práctica

| ACTIVIDAD | CARGA HORARIA | CRONOGRAMA DE DICTADO |
|--|---------------|-----------------------|
| Taller 1: Programación concurrente | 12 | Semanas 1 a 3 |
| Taller 2: Aplicación de lenguajes concurrentes | 24 | Semanas 4 a 9 |
| Taller 3: Bibliotecas de programación paralela y distribuida | 24 | Semanas 10 a 15 |
| TOTAL | 60 | |

Tabla 4: 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 |
| | ENLACE DE ACCESO | https://www.researchgate.net/publication/267367623 | | |
| Sistemas de tiempo real y Lenguajes de Programación | Alan Burns y Andy Wellings | Pearson Education | 1 en Centro de Documentación | 2003 |

Tabla 5: Bibliografía



Tabla 5: Bibliografía (continuación)

| TÍTULO | AUTOR/ES | EDITORIAL | EJEMPLARES DISPONIBLES | AÑO DE EDICIÓN |
|--|--|---|---|----------------|
| 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 |
| | ENLACE DE ACCESO | https://elibro.net/es/lc/unsebiblio/titulos/57462 | | |
| Aplicaciones distribuidas en Java con tecnología RMI | Caballé, S. & Xhafa, F. | Delta Publicaciones | Biblioteca Digital eLibro | 2008 |
| | ENLACE DE ACCESO | https://elibro.net/es/lc/unsebiblio/titulos/168236 | | |
| 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 |
| | ENLACE DE ACCESO | https://www.retro.com.es/_media/programacion/ada/programacion-concurrente-y-tiempo-real.pdf | | |
| Programación orientada a objetos usando Java | Flórez Fernández, H. A. | Ecoe Ediciones | Biblioteca Digital eLibro | 2012 |
| | ENLACE DE ACCESO | https://elibro.net/es/lc/unsebiblio/titulos/69236 | | |
| Java 2: manual de usuario y tutorial (5a. ed.) | Froufe Quintas, A. | RA-MA Editorial | Biblioteca Digital eLibro | 2013 |
| | ENLACE DE ACCESO | https://elibro.net/es/lc/unsebiblio/titulos/106400 | | |
| Concepts, Techniques, and Models of Computer Programming | Haridi, S., & Van-Roy, P | Cambridge: MIT Press | Academia.edu | 2004 |
| | ENLACE DE ACCESO | https://www.academia.edu/34862716/Concepts_Techniques_and_Models_of_Computer_Programming | | |
| Arquitecturas de computadores avanzadas | Jiménez González, D., Guim Bernat, F., & Rodero Castro, I. | Universitat Oberta de Catalunya | Universitat Oberta de Catalunya. Repositorio digital. | 2015 |
| | ENLACE DE ACCESO | http://hdl.handle.net/10609/79549 | | |
| Prog. en Java 2 - Algoritmos, estructuras de datos y POO | Joyanes Aguilar - Zahonero Martinez. | Mc Graw Hill. | 1 en Centro de Documentación | 2002 |



Tabla 5: Bibliografía (continuación)

| TÍTULO | AUTOR/ES | EDITORIAL | EJEMPLARES DISPONIBLES | AÑO DE EDICIÓN |
|--|--------------------------------------|---|---|----------------|
| State models and java programs. | Magee, J., & Kramer, J. | Hoboken: wiley | City, University of London Staff Personal Pages | 1999 |
| | ENLACE DE ACCESO | http://www.staff.city.ac.uk/c.kloukinas/concurrency/concurrency-handouts.pdf | | |
| Concurrencia y sistemas distribuidos.. | Muñoz Escó, F. D. | Editorial de la Universidad Politécnica de Valencia | Biblioteca Digital eLibro | 2013 |
| | ENLACE DE ACCESO | https://elibro.net/es/lc/unsebiblio/titulos/57365 | | |
| Computación de altas prestaciones | Rodero Castro, I., & Guim Bernat, F. | Universitat Oberta de Catalunya | Universitat Oberta de Catalunya. Repositorio digital. | 2012 |
| | ENLACE DE ACCESO | http://hdl.handle.net/10609/63485 | | |
| Java 17: programación avanzada | Vegas Gertrudix, J. M. | RA-MA Editorial | Biblioteca Digital eLibro | 2021 |
| | ENLACE DE ACCESO | https://elibro.net/es/lc/unsebiblio/titulos/222668 | | |

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. | 9 | 20/05 | 20/05 |
| Recuperatorio 1 | | | 10 | 27/05 | 27/05 |
| Parcial 2 | Algoritmos paralelos | | 13 | 24/06 | 24/06 |
| Recuperatorio 2 | | | 14 | 01/07 | 01/07 |

Tabla 6: Programa de Evaluaciones Parciales

8.3.2. Criterios de Evaluación

Respecto de las evaluaciones Parcial Teórico 1, Parcial Teórico 2 y sus respectivos recuperatorios, el criterio de evaluación es el siguiente:

- En relación al Resultado de Aprendizaje RA.7.:

CE.7. El estudiante demuestra conocimiento de la programación concurrente y paralela, explicando detalles complejos con confianza y precisión y respondiendo correctamente las preguntas técnicas planteadas.

Los Criterios de Evaluación generales que serán aplicados en el Taller de Programación Concurrente, el Taller de Aplicación de Lenguajes Concurrentes y el Taller de Bibliotecas de Programación Paralela y Distribuida, son los siguientes:



- En relación al Resultado de Aprendizaje RA.4:

RA.4. Depura programas concurrentes en un IDE, identificando y resolviendo problemas relacionados con la concurrencia, como condiciones de carrera y bloqueos, utilizando las herramientas de depuración disponibles.

- En relación al Resultado de Aprendizaje RA.5:

RA. 5. Demuestra capacidad para innovar en el diseño y desarrollo de aplicaciones de software, aplicando la concurrencia y paralelismo.

- En relación al Resultado de Aprendizaje RA.7:

RA.7. Realiza presentaciones escritas y orales en las que expone y defiende soluciones de programación concurrente y paralela, demostrando dominio del contenido y capacidad de respuesta a preguntas técnicas.

- En relación al Resultado de Aprendizaje RA.9:

RA.9. Desarrolla habilidades de aprendizaje compartido y continuo en el campo de la programación concurrente y paralela, identificando y adquiriendo nuevos conocimientos de manera autónoma y de manera colaborativa.

Los Criterios de Evaluación específicos al Taller de Programación Concurrente, son los siguientes:

- En relación al Resultado de Aprendizaje RA.1:

CE.1. El estudiante desarrolla la especificación de procesos concurrentes de forma clara y precisa, utilizando correctamente la sintaxis del lenguaje FSP.

- En relación al Resultado de Aprendizaje RA.3:

CE.3.1. El estudiante verifica adecuadamente los programas concurrentes aplicando LTSA (Labelled Transition System Analyser) gestionando múltiples hilos/procesos.

Los Criterios de Evaluación específicos al Taller de Aplicación de Lenguajes Concurrentes, son los siguientes:

- En relación al Resultado de Aprendizaje RA.2:

CE.2.1. Aplica de manera correcta y apropiada los constructos concurrentes disponibles en el lenguaje de programación utilizado (hilos, goroutines, STM en Haskell, etc.) reflejando una comprensión sólida de los mecanismos de concurrencia del lenguaje (synchronized en Java, channel en Go, MVar en Haskell, o async en Python).

- En relación al Resultado de Aprendizaje RA.3:

CE.3.2. El estudiante ejecuta y verifica proyectos de programación utilizando los lenguajes Java, Go, Haskell o Python, entre otros, aplicando principios de programación concurrente de manera efectiva.

Los Criterios de Evaluación específicos al Taller de Bibliotecas de Programación Paralela y Distribuida, son los siguientes:



- En relación al Resultado de Aprendizaje RA.2:

CE.2.2. El estudiante implementa correctamente la interfaz MPI en sus programas, demostrando un dominio de las funciones básicas y avanzadas, como MPI_Send, MPI_Recv, MPI_Bcast, MPI_Reduce, entre otras.

CE.2.3. El estudiante implementa correctamente las directivas de OpenMP (como #pragma omp parallel, #pragma omp for, #pragma omp sections, etc.) en su código, asegurando una paralelización efectiva de las tareas

Respecto de las Actividades de Aula y Actividades Aplicativas, los criterios de evaluación son siguientes:

- En relación al Resultado de Aprendizaje RA.6.:

CE.6. El estudiante cumple las tareas asignadas dentro del equipo, entregando trabajos de calidad en los plazos establecidos.

- En relación al Resultado de Aprendizaje RA.7:

RA.7. Realiza presentaciones escritas y orales en las que expone y defiende soluciones de programación concurrente y paralela, demostrando dominio del contenido y capacidad de respuesta a preguntas técnicas.

- En relación al Resultado de Aprendizaje RA.8:

CE. 8. El estudiante identifica y analiza cuestiones éticas relevantes en la práctica de la programación concurrente y paralela. En relación al Resultado de Aprendizaje RA.9:

- En relación al Resultado de Aprendizaje RA.9:

RA.9. Desarrolla habilidades de aprendizaje compartido y continuo en el campo de la programación concurrente y paralela, identificando y adquiriendo nuevos conocimientos de manera autónoma y de manera colaborativa.

Lo descripto se resume en la siguiente Tabla.

Tabla 7: Aplicación de los Criterios de Evaluación

| RESULTADOS DE APRENDIZAJE | CRITERIOS DE EVALUACIÓN | EVALUACIÓN EN LA QUE SE APLICA |
|---|--|------------------------------------|
| RA.1. Desarrolla especificaciones detalladas y precisas de procesos concurrentes asegurando que las soluciones sean comprensibles y replicables usando lenguaje FSP (Finite State Processes). | CE.1. El estudiante desarrolla la especificación de procesos concurrentes de forma clara y precisa, utilizando correctamente la sintaxis del lenguaje FSP. | Taller de Programación Concurrente |



Tabla 7: Aplicación de los Criterios de Evaluación

| RESULTADOS DE APRENDIZAJE | CRITERIOS DE EVALUACIÓN | EVALUACIÓN EN LA QUE SE APLICA |
|---|--|---|
| RA.2. Implementa correctamente soluciones algorítmicas utilizando programación concurrente. | CE.2.1. Aplica de manera correcta y apropiada los constructos concurrentes disponibles en el lenguaje de programación utilizado (hilos, goroutines, STM en Haskell, etc.) reflejando una comprensión sólida de los mecanismos de concurrencia del lenguaje (synchronized en Java, channel en Go, MVar en Haskell, o asyncio en Python) | Taller de Aplicación de Lenguajes Concurrentes |
| | CE.2.2. El estudiante implementa correctamente la interfaz MPI en sus programas, demostrando un dominio de las funciones básicas y avanzadas, como MPI_Send, MPI_Recv, MPI_Bcast, MPI_Reduce, entre otras. | Taller de Bibliotecas de Prog. Paralela y Distribuida |
| | CE.2.3. El estudiante implementa correctamente las directivas de OpenMP (como #pragma omp parallel, #pragma omp for, #pragma omp sections, etc.) en su código, asegurando una paralelización efectiva de las tareas | |
| RA. 3. Ejecuta y verifica proyectos de programación concurrente gestionando múltiples hilos/procesos | CE.3.1. El estudiante verifica adecuadamente los programas concurrentes aplicando LTSA (Labelled Transition System Analyser) gestionando múltiples hilos/procesos. | Taller de Prog. Concurrente |
| | CE.3.2. El estudiante ejecuta y verifica proyectos de programación utilizando los lenguajes Java, Go, Haskell o Python, entre otros, aplicando principios de programación concurrente de manera efectiva. | Taller de Aplicación de Lenguajes Concurrentes |
| RA.4. Depura programas concurrentes en un IDE, identificando y resolviendo problemas relacionados con la concurrencia, como condiciones de carrera y bloqueos, utilizando las herramientas de depuración disponibles. | CE.4. El estudiante utiliza de manera eficiente las herramientas y características del IDE, como la gestión de hilos, puntos de interrupción (breakpoints), y monitoreo de recursos, demostrando conocimiento profundo del entorno. | Taller de Prog. Concurrente Taller de Aplicación de Lenguajes Concurrentes |
| RA. 5. Demuestra capacidad para innovar en el diseño y desarrollo de aplicaciones de software, aplicando la concurrencia y paralelismo. | CE.5. El estudiante aplica correctamente técnicas de concurrencia y paralelismo, demostrando un entendimiento profundo de los conceptos y su aplicación práctica. | Taller de Bibliotecas de Prog. Paralela y Distribuida |



Tabla 7: Aplicación de los Criterios de Evaluación

| RESULTADOS DE APRENDIZAJE | CRITERIOS DE EVALUACIÓN | EVALUACIÓN EN LA QUE SE APLICA |
|---|--|--|
| RA.6. Colabora participativamente en el desarrollo de actividades de revisión teórica y actividades de resolución de problemas aportando al equipo de trabajo sus habilidades y conocimientos sobre la programación concurrente y paralela. | CE.6. El estudiante cumple las tareas asignadas dentro del equipo, entregando trabajos de calidad en los plazos establecidos. | Actividades de aula Actividades aplicativas |
| RA.7. Realiza presentaciones escritas y orales en las que expone y defiende soluciones de programación concurrente y paralela, demostrando dominio del contenido y capacidad de respuesta a preguntas técnicas. | CE.7. El estudiante demuestra conocimiento de la programación concurrente y paralela, explicando detalles complejos con confianza y precisión y respondiendo correctamente las preguntas técnicas planteadas. | Taller de Prog. Concurrente Taller de Aplicación de Lenguajes Concurrentes Taller de Bibliotecas de Prog. Paralela y Distribuida Parcial Teórico 1 / Recuperatorio 1 Parcial Teórico 2 / Recuperatorio 2 |
| RA.8. Establece conclusiones sobre su práctica de programación, considerando los impactos éticos de las soluciones desarrolladas aplicando programación concurrente y paralela. | CE. 8. El estudiante identifica y analiza cuestiones éticas relevantes en la práctica de la programación concurrente y paralela. | Actividades de aula Actividades aplicativas |
| RA.9. Desarrolla habilidades de aprendizaje compartido y continuo en el campo de la programación concurrente y paralela, identificando y adquiriendo nuevos conocimientos de manera autónoma y de manera colaborativa. | CE.9. El estudiante participa activamente en discusiones de grupo, talleres o foros, compartiendo conocimientos y colaborando en la resolución de problemas, demostrando capacidad para aplicar los conocimientos adquiridos de manera autónoma y colaborativa en la resolución de problemas de programación concurrente y paralela. | Taller de Prog. Concurrente Taller de Aplicación de Lenguajes Concurrentes Taller de Bibliotecas de Prog. Paralela y Distribuida |



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 10. El puntaje mínimo para aprobar los parciales es de cinco (5) puntos, sobre una calificación máxima de 10.

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 dos (2) 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