

## UNIVERSIDAD DE MENDOZA – FACULTAD DE INGENIERÍA

<b>CARRERA</b> INGENIERÍA EN INFORMÁTICA	<b>ASIGNATURA</b> COMPUTACIÓN II	<b>CÓDIGO</b> 2038
<b>CURSO</b> TERCER AÑO	<b>ÁREA</b> Tecnologías Aplicadas	<b>ULTIMA REVISIÓN</b> Febrero 2.019
<b>MATERIAS CORRELATIVAS:</b> 2028 – COMPUTACIÓN 2026- SISTEMAS OPERATIVOS		<b>AÑO LECTIVO 2020</b>

Profesor Titular:

Profesor Asociado:

Profesores Adjuntos: Ing. Carlos Taffernaberry

Jefes de trabajos prácticos: Ing. Federico Hernandez

Carga Horaria Semanal: 4

Carga Horaria Total: 120

### **OBJETIVOS:**

- Adquirir dominio en las herramientas de desarrollo de sistemas como versionadores, lenguajes de programación y herramientas colaborativas de trabajo en grupo y orquestación de servicios.
- Comprender distintas formas de paralelizar la programación y sus diferencias.
- Justificar la necesidad de mecanismos de comunicación entre aplicaciones o servicios.
- Conocer la evolución de las arquitecturas de una aplicación para Internet (de cliente-servidor a microservicios y serverless).
- Adquirir dominio en solución de problemas de computación con complejidad.

### **PROGRAMA ANALÍTICO:**

#### **Capítulo I: Conceptos Básicos**

Modelo de ejecución de programas. Distintas arquitecturas, distintos modelos de programación.

Uso de Herramientas de Depuración, Control de versiones y trabajo colaborativo en grupo.

#### **Capítulo II: Programación concurrente**

Concepto procesos e hilos. Problemas clásicos. Sección crítica. Mecanismos de sincronización Interbloqueo y tratamiento.

Mecanismos de Sincronización: Patrones básicos de sincronización. Semáforos y memoria compartida. Aplicación en problemas clásicos.

Paso de mensaje: Conceptos básicos, problema de sección crítica. Resolución de problemas clásicos.

Otros mecanismos de sincronización.

### **Capítulo III: API de Sockets**

Generalidades sobre los *sockets*. Familias de protocolos, tipos de sockets. Creación de sockets de la familia AF\_INET y AF\_INET6, apertura activa y pasiva. Direccionamiento.

Implementación de aplicaciones para servicios TCP/IP.

### **Capítulo IV: Programación paralela**

Justificación de programación paralela. Formas comunes de paralelización. Mecanismos de comunicación y problemas de paralelización (interbloqueo, inanición y condiciones de competencia)

Técnicas de diseño de algoritmos paralelos.

Aplicación de las técnicas en la resolución de problemas clásicos.

Distribución de tareas.

### **Capítulo V: Programación asíncrona**

Paradigma de programación asíncrona. Bloqueo, no bloqueo y operaciones asíncronas. Colas de tarea distribuidas.

Uso de eventos y funciones de sondeo o pooling.

Resolución de problemas asíncronos.

### **Capítulo VI: Herramientas de Orquestación**

Docker: almacenamiento y manejo de imágenes, Servicio de Registry. Manejo de contenedores: comandos básicos, manejo de red y volúmenes. Docker Compose: introducción, lenguaje YAML, comandos. Casos de uso

<b>Formación Práctica</b>	<b>Horas</b>
Resolución de Problemas Rutinarios:	
Laboratorio, Trabajo de Campo:	45
Resolución de Problemas Abiertos:	45
Proyecto y Diseño:	

### **PROGRAMA DE TRABAJOS PRÁCTICOS:**

Práctico No 1: Resolución de problemas de programación concurrente.

Práctico No 2: Resolución de problemas de paradigma cliente servidor.

Práctico No 3: Resolución de problemas de programación paralela.

Práctico No 4: Resolución de problemas de programación asíncrona.

Práctico No 5: Resolución de casos de orquestación de servicios.

### **ARTICULACIÓN HORIZONTAL Y VERTICAL DE CONTENIDOS:**

- Los contenidos abordados en esta materia se basan en conceptos de las siguientes cátedras:

<b>Asignatura</b>	<b>Curso</b>
Informática	1er.
Sistemas Operativos	2do. Año
Computación I	2do. Año

- Comparte e integra elementos horizontalmente con las siguientes cátedras:

<b>Asignatura</b>	<b>Curso</b>
Redes de datos	3er. Año
Programación I	3er. Año
Teoría de Compiladores	3er. Año

- Los contenidos abordados en esta materia aportan conceptos a las siguientes cátedras:

<b>Asignatura</b>	<b>Curso</b>
Teleinformática	4to. Año
Seguridad Informática I	4to. Año

### **CONDICIONES PARA REGULARIZAR LA MATERIA y RÉGIMEN DE EVALUACIÓN:**

La materia se aprobará por medio de un examen final.

Para regularizar la materia el alumno deberá:

- Asistir al 80% de las clases.
- Aprobar el 100% de los trabajos prácticos.

### **BIBLIOGRAFÍA:**

(Título – Autor – Editorial – Año de Edición)

#### **Principal:**

Autor	Título	Editorial	Año Ed.	Dispon.
Allen B. Downey	The Little Book of Semaphores Second Edition	Green Tea Press	2016	SI
David Vallejo Fernández Carlos González Morcillo Javier A. Albusac Jiménez	Programación Concurrente y en Tiempo Real. Tercera Edición	Addison Wesley	2016	SI
Jan Palach	Parallel programming with Python. First edition.	Packt Publishing	2014	SI
Russ McKendrick, Scott Gallagher	Mastering Docker. Third Edition.	Packt Publishing	2018	NO

**De Consulta:**

Autor	Título	Editorial	Año Ed.	Dispon.
Giancarlo Zaccone	Python Parallel Programming Cookbook	Packt	2015	
Quan Nguyen	Mastering Concurrency in Python	Packt	2018	

<https://www.python.org/>

**ESTRATEGIAS DIDÁCTICAS UTILIZADAS:**

- Clases expositivas
- Discusiones grupales
- Trabajos prácticos grupales

**RECURSOS DIDÁCTICOS UTILIZADOS:**

- Medios informáticos
- Pizarrón
- Internet
- Laboratorio
- Apuntes y bibliografía para consulta de los alumnos

**PROGRAMA DE EXAMEN :**

Desarrollo individual, exposición y defensa de Aplicación con la complejidad suficiente para abarcar los puntos mas destacables de la materia. Durante consultas se evalua la maduración de los conceptos, como el uso adecuado de las herramientas de programación.