

## **Programa Analítico de la Cátedra Programación Concurrente**

**Año 2.024**

**Carrera Ingeniería Informática Plan de Estudios 2.022**

### ***Unidad 1: Programación Orientada a Objetos (POO)***

Definición. Razones y Características. Estructura de un objeto: relaciones. Tipos de relaciones. Propiedades. Métodos.

Clases, objetos, mensajes. Relaciones (asociación, agregación, generalización y herencia, dependencia).

Encapsulamiento y ocultación. Organización jerárquica de los objetos.

Polimorfismo. Herencia. Beneficios y problemas derivados de la utilización de la POO.

### ***Unidad 2: Programación Concurrente***

Introducción. Programas concurrentes. Indicadores de concurrencia. Ventajas y desventajas de la utilización de programas concurrentes. Arquitecturas concurrentes.

Comunicación y sincronización. Tipos de comunicación. Tipos de sincronización.

Intercambio de mensajes. Instrucciones atómicas. Funcionamiento correcto.

Regiones críticas. Exclusión mutua.

Semáforos. Regiones críticas. Regiones críticas condicionales. Monitores.

Propiedades de los programas concurrentes. Problema del productor y consumidor. Bloqueos.

Monitores. Introducción. Características.

### ***Unidad 3: Hilos***

Introducción. Programas y procesos. Hilos.

Implementación en un lenguaje de programación: tipos de datos, operadores, estructura, IDEs.

Terminación. Estados. Herramientas de Planificación de Hilos. Multiprocesamiento.

### ***Unidad 4: Monitores***

Introducción. Excepciones con try-catch. Bloqueo de un hilo. Notificación de espera en hilos. Wait(), Notify() y Notifyall() como cola de espera. Modelo Productor/Consumidor. Interrupción de un hilo. Implementación.

### ***Unidad 5: Implementación de Semáforos y otras variantes de implementación de Hilos***

Runnable: introducción. Implementación del método run(). Creación y ejecución de tareas. Implementación.

Semáforos: introducción. Métodos acquire y release. Diferencias con wait y notify. Implementación de semáforos con herencia con thread y runnable.

Executor Framework: introducción. Implementación de la Interfaz Executor Service. Métodos. Otras implementaciones de la clase Executors.

Firma:



Aclaración:

Ing. José Faifán H.