


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

			PRÁCTICA DE LABORATORIO		
CARRERA: COMPUTACION			ASIGNATURA: COMPUTACION PARALELA		
NRO. PRÁCTICA:	4	TÍTULO PRÁCTICA: Desarrollo e implementación de aplicaciones de cómputo paralelo basado procesos			
OBJETIVOS					
<ul style="list-style-type: none">• Diseñar e implementar una aplicación concurrente que simule un sistema urbano de control de tráfico vehicular.• Aplicar técnicas de paralelismo basado en procesos para coordinar múltiples semáforos.• Utilizar mecanismos de comunicación y sincronización entre procesos en Python.• Incorporar una interfaz gráfica (GUI) para visualizar el funcionamiento de la simulación.					
INSTRUCCIONES		Simulación de control de tráfico vehicular con paralelismo basado en procesos <p>En esta práctica se desarrollará una simulación de tráfico vehicular en una intersección urbana de la ciudad de Cuenca (a elección de los estudiantes) usando multiprocessing en Python. Cada semáforo de la intersección funcionará como un proceso independiente, que debe sincronizarse con los otros para evitar colisiones y congestión.</p> <p>La intersección contiene cuatro vías (Norte, Sur, Este, Oeste), cada una con un semáforo. En cada ciclo, solo una vía o conjunto de vías no conflictivas puede estar en verde, mientras las otras están en rojo o amarillo.</p> <p>Cada semáforo debe:</p> <ol style="list-style-type: none">1. Operar como un proceso independiente.2. Comunicarse con un controlador central que determina qué vías pueden avanzar.3. Simular vehículos que llegan, esperan y avanzan según el color del semáforo. <p>Nota: se debe incluir una interfaz gráfica que muestre el estado actual de cada semáforo y la circulación de vehículos. El sistema debe operar en ciclos simulando el flujo de tráfico realista, con lógica de control de prioridades y espera de vehículos.</p>			
		Requerimientos funcionales <p>General</p> <ol style="list-style-type: none">1. Estructura basada en clases: Implementar clases como Semaforo, ControladorTrafico, Vehiculo, etc.2. Paralelismo con procesos: Usar multiprocessing.Process para ejecutar cada semáforo.3. Sincronización de procesos: Utilizar mecanismos como Barrier, Lock, Semaphore para coordinar estados.4. Interfaz gráfica: Desarrollar la GUI usando Tkinter (recomendado) o PyQt para visualizar los semáforos y vehículos.			

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

	Detalles específicos <ol style="list-style-type: none"> 1. Crear una clase Semaforo: <ul style="list-style-type: none"> ○ Controla su propio estado (rojo, amarillo, verde). ○ Cambia de estado cíclicamente. ○ Se comunica con el controlador central para obtener permiso de paso. 2. Crear una clase ControladorTrafico: <ul style="list-style-type: none"> ○ Supervisa todos los semáforos. ○ Evita colisiones activando solo los semáforos no conflictivos. ○ Usa un ciclo de tiempo y barreras para controlar el flujo. 3. Vehículos simulados: <ul style="list-style-type: none"> ○ Cada vía debe tener una lista dinámica de vehículos en espera. ○ Visualizar su avance cuando el semáforo está en verde. 4. Interfaz gráfica: <ul style="list-style-type: none"> ○ Mostrar la intersección desde una vista superior. ○ Reflejar el estado de cada semáforo con colores. ○ Representar los vehículos como bloques que avanzan. 5. Generar un reporte final (en consola): <ul style="list-style-type: none"> ○ Número total de vehículos que cruzaron. ○ Tiempos de espera promedio por vía. ○ Ciclos completados correctamente. 												
	Guía de evaluación (5 puntos totales) <table> <tr> <th>Criterio</th><th>Puntos</th></tr> <tr> <td>Uso adecuado de clases y métodos</td><td>1.0</td></tr> <tr> <td>Implementación de procesos y sincronización</td><td>1.0</td></tr> <tr> <td>Comunicación entre procesos (colas, pipes)</td><td>1.0</td></tr> <tr> <td>Desarrollo funcional de la GUI</td><td>1.0</td></tr> <tr> <td>Presentación, informe técnico, resultados y reporte de simulación</td><td>1.0</td></tr> </table>	Criterio	Puntos	Uso adecuado de clases y métodos	1.0	Implementación de procesos y sincronización	1.0	Comunicación entre procesos (colas, pipes)	1.0	Desarrollo funcional de la GUI	1.0	Presentación, informe técnico, resultados y reporte de simulación	1.0
Criterio	Puntos												
Uso adecuado de clases y métodos	1.0												
Implementación de procesos y sincronización	1.0												
Comunicación entre procesos (colas, pipes)	1.0												
Desarrollo funcional de la GUI	1.0												
Presentación, informe técnico, resultados y reporte de simulación	1.0												
ACTIVIDADES POR DESARROLLAR													
<ol style="list-style-type: none"> 1. Diseñar las clases Semaforo, ControladorTrafico y las estructuras de vehículos. 2. Implementar los procesos y su sincronización. 3. Desarrollar la GUI para representar la simulación. 4. Ejecutar la simulación por al menos 10 ciclos completos. 5. Generar y subir el informe al AVAC de la práctica con el desarrollo de cada uno de los puntos descritos anteriormente, incluyendo: capturas de pantalla, código explicativo, resultados, conclusiones, firma de los integrantes. 													
RESULTADO(S) OBTENIDO(S): <ul style="list-style-type: none"> • Diseña e implementa aplicaciones en paralelo. • Desarrollar aplicaciones en paralelo con control visual de procesos concurrentes. • Integrar programación de procesos con visualización gráfica en tiempo real. 													
CONCLUSIONES: <ul style="list-style-type: none"> • Los estudiantes podrán interactuar con el análisis y diseño de algoritmos paralelos. • El estudiante podrá implementar y visualizar sistemas complejos distribuidos. 													

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- Se fortalecerán conceptos de comunicación, sincronización y visualización en paralelismo.

RECOMENDACIONES:

Docente: Ing. Gabriel León Paredes, PhD.



Firma: