UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN

MATEMÁTICAS APLICADAS Y COMPUTACIÓN

Práctica de concurrencia y paralelismo

Sistema hospitalario

Programación Paralela y Concurrente

Profesor:

José Gustavo Fuentes Cabrera

Alumna:

Pardo Juárez Marlene

Número de cuenta:

42008805-1

Fecha:

Mayo, 2025.

“Por mi raza hablará el espíritu”

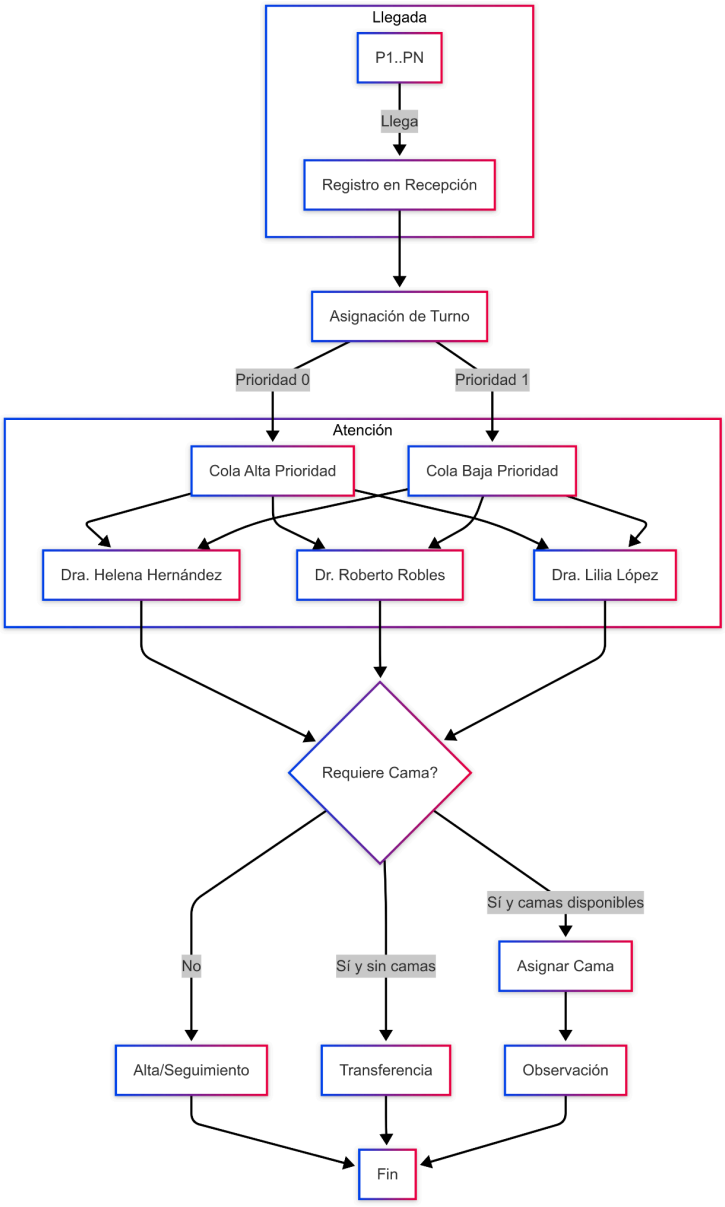
Introducción

Este documento presenta la implementación de concurrencia y paralelismo en la creación de un sistema hospitalario en donde se llevará a cabo la asignación de varias tareas relacionadas con la admisión de pacientes al hospital.

El objetivo es que el sistema aplique los paradigmas de programación paralela, concurrente y asíncrona. Es por ello que usaremos los conceptos que vimos en clase para realizar un sistema automatizado y funcionando de manera realista.

Diagrama del sistema

El funcionamiento del sistema es como se muestra en el siguiente diagrama:



Concurrente: Simula múltiples pacientes llegando y registrándose en recepción en paralelo, controlados con semáforos para reflejar plazas limitadas.

Paralelo: Realiza el cómputo de diagnóstico (simulación de riesgo) en procesos separados para aprovechar múltiples núcleos y reducir latencia.

Asíncrono: Gestiona colas de pacientes por prioridad (grave/crítico vs. leve/moderado), asegurando que los casos urgentes se atienden primero.

Implementación

A continuación, se presenta el código utilizado para implementar el sistema planteado. Para poder ver la ejecución siga el enlace: https://colab.research.google.com/drive/1FmFlKDK-nFDq9-GQt-_uX9yf_c9Bxihh?usp=sharing

Código Propuesto:

```
import threading
import multiprocessing
import random
import time
from queue import PriorityQueue
from concurrent.futures import ThreadPoolExecutor

# ------------------------------------------------------------------------------------------------------
# Se establecen diagósticos posibles de ejemplo
# ------------------------------------------------------------------------------------------------------
diagnosticos_posibles = [
    "Gripe leve", "Infección respiratoria", "Fractura de brazo", "Apendicitis",
    "COVID-19", "Hipertensión", "Diabetes no controlada",
    "Traumatismo craneoencefálico", "Inconsciente", "Requiere más estudios"
]

#Doctores en el turno
doc_doctores = ["Dra. Helena Hernández", "Dr. Roberto Robles", "Dra. Lilia López"]

sem_registro = threading.Semaphore(3)  # puestos de recepción
mutex_camas = threading.Lock()
camas_disponibles = 5  # número de camas disponibles en el hospital
cola_turnos = PriorityQueue()

# Pool de procesos global para diagnóstico ligero
pool = multiprocessing.Pool(processes=2)

#Se asigna un riesgo y un diagnostico aleatorio
def funcion_diagnostico(paciente_id):
    riesgo = random.random()
    diagnostico = random.choice(diagnosticos_posibles)
    print(f"Paciente {paciente_id}: diagnóstico -> {diagnostico}, riesgo: {riesgo:.2f}")
    return diagnostico, riesgo

# ------------------------------------------------------------------------------------------------------
# Ejecuta diagnóstico en paralelo
# ------------------------------------------------------------------------------------------------------
# Ejecuta diagnóstico en paralelo
def diagnostico_intensivo(id_paciente):
    return pool.apply(funcion_diagnostico, args=(id_paciente,))
```

```
# ---------------------------------------------------------------------------------------------------------
# Determina el estado del paciente según diagnóstico y riesgo, en algunos es específico
# ---------------------------------------------------------------------------------------------------------
def obtener_estado(diagnostico, riesgo):
    if diagnostico == "Inconsciente":
        return "crítico"
    if diagnostico == "Traumatismo craneoencefálico":
        return "grave"
    if diagnostico == "COVID-19":
        return random.choice(["grave"]*3 + ["crítico"])
    if diagnostico == "Fractura de brazo":
        return "grave"
    if diagnostico == "Requiere más estudios":
        return random.choice(["leve", "moderado", "grave", "crítico"])
    if riesgo > 0.8:
        return "crítico"
    if riesgo > 0.6:
        return "grave"
    if riesgo > 0.4:
        return "moderado"
    return "leve"

# ---------------------------------------------------------------------------------------------------------
# Registro en recepción
# ---------------------------------------------------------------------------------------------------------
def registrar_paciente(id_paciente):
    with sem_registro:
        print(f"Paciente {id_paciente}: registrándose en recepción...")
        time.sleep(random.uniform(0.3, 0.6))
        print(f"Paciente {id_paciente}: registro completado")

# ---------------------------------------------------------------------------------------------------------
# Encolar paciente con prioridad, dependiendo de su estado y diagnostico
# ---------------------------------------------------------------------------------------------------------
def encolar_paciente(id_paciente):
    diagnostico, riesgo = diagnostico_intensivo(id_paciente)
    estado = obtener_estado(diagnostico, riesgo)
    if diagnostico == "Requiere más estudios":
        cita = time.strftime("%Y-%m-%d", time.localtime(time.time() + 86400 * random.randint(1,7)))
        print(f"Paciente {id_paciente}: cita de seguimiento para {cita}")
    prioridad = 0 if estado in ("grave", "crítico") else 1
    cola_turnos.put((prioridad, id_paciente, diagnostico, riesgo, estado))
    print(f"Paciente {id_paciente}: encolado con estado '{estado}' (prioridad={prioridad})")
```

```
# ------------------------------------------------------------------------------------------------------------
# Atención de pacientes por doctor
# ------------------------------------------------------------------------------------------------------------
def atender_pacientes(doctor):
    global camas_disponibles
    while True:
        prioridad, id_paciente, diagnostico, riesgo, estado = cola_turnos.get()
        if id_paciente is None:
            cola_turnos.task_done()
            break
        print(f"{doctor}: atendiendo paciente {id_paciente} -> {diagnostico} (estado: {estado})")
        time.sleep(random.uniform(0.8, 1.2))
        if estado in ("grave", "crítico"):
#asignacion de camas disponibles y actualizacion
            with mutex_camas:
                if camas_disponibles > 0:
                    camas_disponibles -= 1
                    print(f"Paciente {id_paciente}: asignada cama (restantes: {camas_disponibles})")
                else:
                    # No hay camas disponibles
                    if diagnostico == "Fractura de brazo": #eleccion del paciente en este caso
                        decision = random.choice(["esperar cama", "transferencia"])
                        print(f"Paciente {id_paciente}: sin camas disponibles. Decide {decision}")
                        if decision == "transferencia":
                            print(f"Paciente {id_paciente}: solicitando ambulancia y transfiriendo a otro hospital")
#cuando no hay camas se transfiere el paciente en estado grave a otro hospital
                    else:
                        print(f"Paciente {id_paciente}: sin camas disponibles. Solicitando ambulancia y transfiriendo")
            print(f"Paciente {id_paciente}: en observación intensa o en traslado")
        elif diagnostico == "Requiere más estudios":
            print(f"Paciente {id_paciente}: alta temporal, esperar estudios")
        else:
            print(f"Paciente {id_paciente}: medicación administrada, alta")
        print(f"{doctor}: finalizó paciente {id_paciente}\n")
        cola_turnos.task_done()

# Flujo de cada paciente
def proceso_paciente(id_paciente):
    registrar_paciente(id_paciente)
    encolar_paciente(id_paciente)
```

```
# ------------------------------------------------------------------------------------------------------------
# Funcion principal
# ------------------------------------------------------------------------------------------------------------

if __name__ == "__main__":
#Se hace la prueba con 20 pacientes
   pacientes = range(1, 21)

#se inicia un temporizador para calcular el tiempo total del proceso
   inicio = time.time()
   # Iniciar hilos de doctores
   hilos_doctores = []
   for doc in doc_doctores:
      t = threading.Thread(target=atender_pacientes, args=(doc,))
      t.start()
      hilos_doctores.append(t)

   # Registrar y encolar todos los pacientes
   with ThreadPoolExecutor(max_workers=5) as executor:
      executor.map(proceso_paciente, pacientes)

   # Enviar sentinelas para terminar a doctores
   for _ in doc_doctores:
      cola_turnos.put((2, None, None, None, None))

   cola_turnos.join()

   # Esperar a que terminen los hilos de doctores
   for t in hilos_doctores:
      t.join()

#Se para el temporizador
   fin = time.time()

#Se imprime el tiempo que tardó el proceso al finalizar.
   print(f"\nSimulación completada en {fin - inicio:.2f} segundos")
```

Pruebas:

Se realiza la prueba con 20 pacientes y los resultados son:

```
Paciente 1: registrándose en recepción...
Paciente 2: registrándose en recepción...
Paciente 3: registrándose en recepción...
Paciente 3: diagnóstico -> Diabetes no controlada, riesgo: 0.70
Paciente 2: diagnóstico -> Hipertensión, riesgo: 0.54
Paciente 1: diagnóstico -> COVID-19, riesgo: 0.57
Paciente 3: registro completado
Paciente 4: registrándose en recepción...
Paciente 3: encolado con estado 'grave' (prioridad=0)
Paciente 2: registro completado
Paciente 5: registrándose en recepción...
Dra. Helena Hernández: atendiendo paciente 3 -> Diabetes no controlada (estado: grave)
Paciente 2: encolado con estado 'moderado' (prioridad=1)
Dr. Roberto Robles: atendiendo paciente 2 -> Hipertensión (estado: moderado)
Paciente 1: registro completado
Paciente 6: registrándose en recepción...
Paciente 1: encolado con estado 'grave' (prioridad=0)
Dra. Lilia López: atendiendo paciente 1 -> COVID-19 (estado: grave)
Paciente 4: diagnóstico -> Hipertensión, riesgo: 0.90
Paciente 4: registro completado
Paciente 7: registrándose en recepción...
Paciente 4: encolado con estado 'crítico' (prioridad=0)
Paciente 5: diagnóstico -> Inconsciente, riesgo: 0.01
Paciente 6: diagnóstico -> Diabetes no controlada, riesgo: 0.02
Paciente 7: diagnóstico -> Diabetes no controlada, riesgo: 0.24
Paciente 5: registro completado
Paciente 8: registrándose en recepción...


Paciente 5: encolado con estado 'crítico' (prioridad=0)
Paciente 6: registro completado
Paciente 9: registrándose en recepción...
Paciente 6: encolado con estado 'leve' (prioridad=1)
Paciente 7: registro completado
Paciente 10: registrándose en recepción...
Paciente 7: encolado con estado 'leve' (prioridad=1)
Paciente 9: diagnóstico -> Inconsciente, riesgo: 0.55
Paciente 10: diagnóstico -> Requiere más estudios, riesgo: 0.68
Paciente 9: registro completado
Paciente 11: registrándose en recepción...
Paciente 9: encolado con estado 'crítico' (prioridad=0)
Paciente 10: registro completado
Paciente 12: registrándose en recepción...
Paciente 10: cita de seguimiento para 2025-05-17
Paciente 10: encolado con estado 'crítico' (prioridad=0)
Paciente 2: medicación administrada, alta
Dr. Roberto Robles: finalizó paciente 2

Dr. Roberto Robles: atendiendo paciente 4 -> Hipertensión (estado: crítico)
Paciente 3: asignada cama (restantes: 4)
Paciente 3: en observación intensa o en traslado
Dra. Helena Hernández: finalizó paciente 3

Dra. Helena Hernández: atendiendo paciente 5 -> Inconsciente (estado: crítico)
Paciente 8: registro completado
Paciente 13: registrándose en recepción...
Paciente 8: diagnóstico -> Infección respiratoria, riesgo: 0.82
Paciente 8: encolado con estado 'crítico' (prioridad=0)


Paciente 1: asignada cama (restantes: 3)
Paciente 1: en observación intensa o en traslado
Dra. Lilia López: finalizó paciente 1

Dra. Lilia López: atendiendo paciente 8 -> Infección respiratoria (estado: crítico)
Paciente 12: diagnóstico -> Fractura de brazo, riesgo: 0.44
Paciente 11: diagnóstico -> Hipertensión, riesgo: 0.88
Paciente 12: registro completado
Paciente 14: registrándose en recepción...
Paciente 12: encolado con estado 'grave' (prioridad=0)
Paciente 11: registro completado
Paciente 15: registrándose en recepción...
Paciente 11: encolado con estado 'crítico' (prioridad=0)
Paciente 15: diagnóstico -> Traumatismo craneoencefálico, riesgo: 0.96
Paciente 13: diagnóstico -> Inconsciente, riesgo: 0.16
Paciente 14: diagnóstico -> Traumatismo craneoencefálico, riesgo: 0.08
Paciente 15: registro completado
Paciente 16: registrándose en recepción...
Paciente 13: registro completado
Paciente 17: registrándose en recepción...
Paciente 15: encolado con estado 'grave' (prioridad=0)
Paciente 13: encolado con estado 'crítico' (prioridad=0)
Paciente 14: registro completado
Paciente 18: registrándose en recepción...
Paciente 14: encolado con estado 'grave' (prioridad=0)
Paciente 18: diagnóstico -> Infección respiratoria, riesgo: 0.29
Paciente 17: diagnóstico -> Hipertensión, riesgo: 0.70
Paciente 4: asignada cama (restantes: 2)
Paciente 4: en observación intensa o en traslado
Dr. Roberto Robles: finalizó paciente 4
```

```
Dr. Roberto Robles: atendiendo paciente 9 -> Inconsciente (estado: crítico)
Paciente 5: asignada cama (restantes: 1)
Paciente 5: en observación intensa o en traslado
Dra. Helena Hernández: finalizó paciente 5

Dra. Helena Hernández: atendiendo paciente 10 -> Requiere más estudios (estado: crítico)
Paciente 18: registro completado
Paciente 19: registrándose en recepción...
Paciente 18: encolado con estado 'leve' (prioridad=1)
Paciente 17: registro completado
Paciente 20: registrándose en recepción...
Paciente 17: encolado con estado 'grave' (prioridad=0)
Paciente 16: diagnóstico -> Traumatismo craneoencefálico, riesgo: 0.12
Paciente 16: registro completado
Paciente 16: encolado con estado 'grave' (prioridad=0)
Paciente 20: diagnóstico -> Gripe leve, riesgo: 0.44
Paciente 8: asignada cama (restantes: 0)
Paciente 8: en observación intensa o en traslado
Dra. Lilia López: finalizó paciente 8

Dra. Lilia López: atendiendo paciente 11 -> Hipertensión (estado: crítico)
Paciente 20: registro completado
Paciente 20: encolado con estado 'moderado' (prioridad=1)
Paciente 19: diagnóstico -> COVID-19, riesgo: 0.80
Paciente 19: registro completado
Paciente 19: encolado con estado 'grave' (prioridad=0)
Paciente 9: sin camas disponibles. Solicitando ambulancia y transfiriendo
Paciente 9: en observación intensa o en traslado
Dr. Roberto Robles: finalizó paciente 9


Dr. Roberto Robles: atendiendo paciente 12 -> Fractura de brazo (estado: grave)
Paciente 10: sin camas disponibles. Solicitando ambulancia y transfiriendo
Paciente 10: en observación intensa o en traslado
Dra. Helena Hernández: finalizó paciente 10

Dra. Helena Hernández: atendiendo paciente 13 -> Inconsciente (estado: crítico)
Paciente 11: sin camas disponibles. Solicitando ambulancia y transfiriendo
Paciente 11: en observación intensa o en traslado
Dra. Lilia López: finalizó paciente 11

Dra. Lilia López: atendiendo paciente 14 -> Traumatismo craneoencefálico (estado: grave)
Paciente 12: sin camas disponibles. Decide esperar cama
Paciente 12: en observación intensa o en traslado
Dr. Roberto Robles: finalizó paciente 12

Dr. Roberto Robles: atendiendo paciente 15 -> Traumatismo craneoencefálico (estado: grave)
Paciente 13: sin camas disponibles. Solicitando ambulancia y transfiriendo
Paciente 13: en observación intensa o en traslado
Dra. Helena Hernández: finalizó paciente 13

Dra. Helena Hernández: atendiendo paciente 16 -> Traumatismo craneoencefálico (estado: grave)
Paciente 14: sin camas disponibles. Solicitando ambulancia y transfiriendo
Paciente 14: en observación intensa o en traslado
Dra. Lilia López: finalizó paciente 14

Dra. Lilia López: atendiendo paciente 17 -> Hipertensión (estado: grave)
Paciente 15: sin camas disponibles. Solicitando ambulancia y transfiriendo
Paciente 15: en observación intensa o en traslado
Dr. Roberto Robles: finalizó paciente 15


Dr. Roberto Robles: atendiendo paciente 19 -> COVID-19 (estado: grave)
Paciente 16: sin camas disponibles. Solicitando ambulancia y transfiriendo
Paciente 16: en observación intensa o en traslado
Dra. Helena Hernández: finalizó paciente 16

Dra. Helena Hernández: atendiendo paciente 6 -> Diabetes no controlada (estado: leve)
Paciente 17: sin camas disponibles. Solicitando ambulancia y transfiriendo
Paciente 17: en observación intensa o en traslado
Dra. Lilia López: finalizó paciente 17

Dra. Lilia López: atendiendo paciente 7 -> Diabetes no controlada (estado: leve)
Paciente 19: sin camas disponibles. Solicitando ambulancia y transfiriendo
Paciente 19: en observación intensa o en traslado
Dr. Roberto Robles: finalizó paciente 19

Dr. Roberto Robles: atendiendo paciente 18 -> Infección respiratoria (estado: leve)
Paciente 6: medicación administrada, alta
Dra. Helena Hernández: finalizó paciente 6

Dra. Helena Hernández: atendiendo paciente 20 -> Gripe leve (estado: moderado)
Paciente 7: medicación administrada, alta
Dra. Lilia López: finalizó paciente 7

Paciente 18: medicación administrada, alta
Dr. Roberto Robles: finalizó paciente 18

Paciente 20: medicación administrada, alta
Dra. Helena Hernández: finalizó paciente 20


Simulación completada en 7.50 segundos
```

Conclusión

El uso de hilos es muy conveniente para agilizar ciertos procesos, pero también es más conveniente realizar otros mediante procesos paralelos y de forma asíncrona. La simulación del sistema hospitalario permitió aplicar e integrar de forma exitosa los tres paradigmas de programación: la concurrencia para manejar múltiples pacientes que ingresan y son registrados simultáneamente, la paralelización para ejecutar tareas intensivas como el diagnóstico automatizado, y la programación asíncrona para simular respuestas de servicios externos, como modelos de IA (que podrían ser tomados como apoyo) o servicios de ambulancia.

El sistema desarrollado refleja un flujo hospitalario realista, con un proceso ordenado desde el registro hasta el alta o transferencia, considerando variables como el estado de salud del paciente, la disponibilidad de camas y la asignación equitativa de doctores. Se establecieron prioridades médicas coherentes, como dar atención urgente a pacientes inconscientes o con COVID-19 en estado crítico. Todos éstos son de fácil escalabilidad en caso de que se requiera agregar ya sea doctores, camas, más diagnósticos, etc.

El proyecto permitió reforzar conceptos clave de la asignatura, usando los conceptos que vimos, lo cual a pesar de eso, fue un reto. Este código tiene muchas áreas de oportunidad de mejora, pero su desarrollo sirvió de aprendizaje.

Bibliografía

Este documento se realizó con ayuda del material proporcionado por el profesor, para la asignatura, así como el apoyo de inteligencias artificiales e internet para pulirlo, algunos prompts usados fueron:

> Realiza modificaciones al código considerando que el diagnóstico "Traumatismo craneoencefálico" es un estado "grave", el diagnóstico "COVID-19" en la mayoría de los casos también es grave o crítico, la "fractura de brazo" también es "grave", agrega el diagnóstico "inconsciente" con estado "crítico" en todos los pacientes con este diagnóstico, así que el nivel de prioridad debe de ser el principal y debería atenderse con urgencia. Agrega un diagnóstico que sea "Requiere más estudios" y se le agende una cita de seguimiento para otro día, el estado de dicho diagnóstico puede ser aleatorio.

Hubo varios similares o en los que una parte del código no funcionaba bien.

> Modifica el código para que cuando el paciente esté en estado "grave" o "crítico" y el hospital no tenga camas disponibles, se le solicite una ambulancia y se transfiera a otro hospital, esto no aplica para "fractura de brazo", para este diagnóstico se le deja a decisión del paciente si se quiere ir a otro hospital o esperar a que se desocupe una cama. Aumenta el número de camas a 5, siguen siendo 3 doctores.

> Ahora, me marca error en:
> cola_turnos.put((1, None, None, None, None))
> y me muestra: TypeError: '<' not supported between instances of 'NoneType' and 'int'
> Esta línea también hace que el proceso se detenga:
> cola_turnos.join()
> Resuelve esto

En general, la IA me ayudó a corregir errores y agregar consideraciones realistas para el hospital, ya que aunque a veces resultan triviales las modificaciones, estando trabajando, recibir ayuda es muy gratificante.