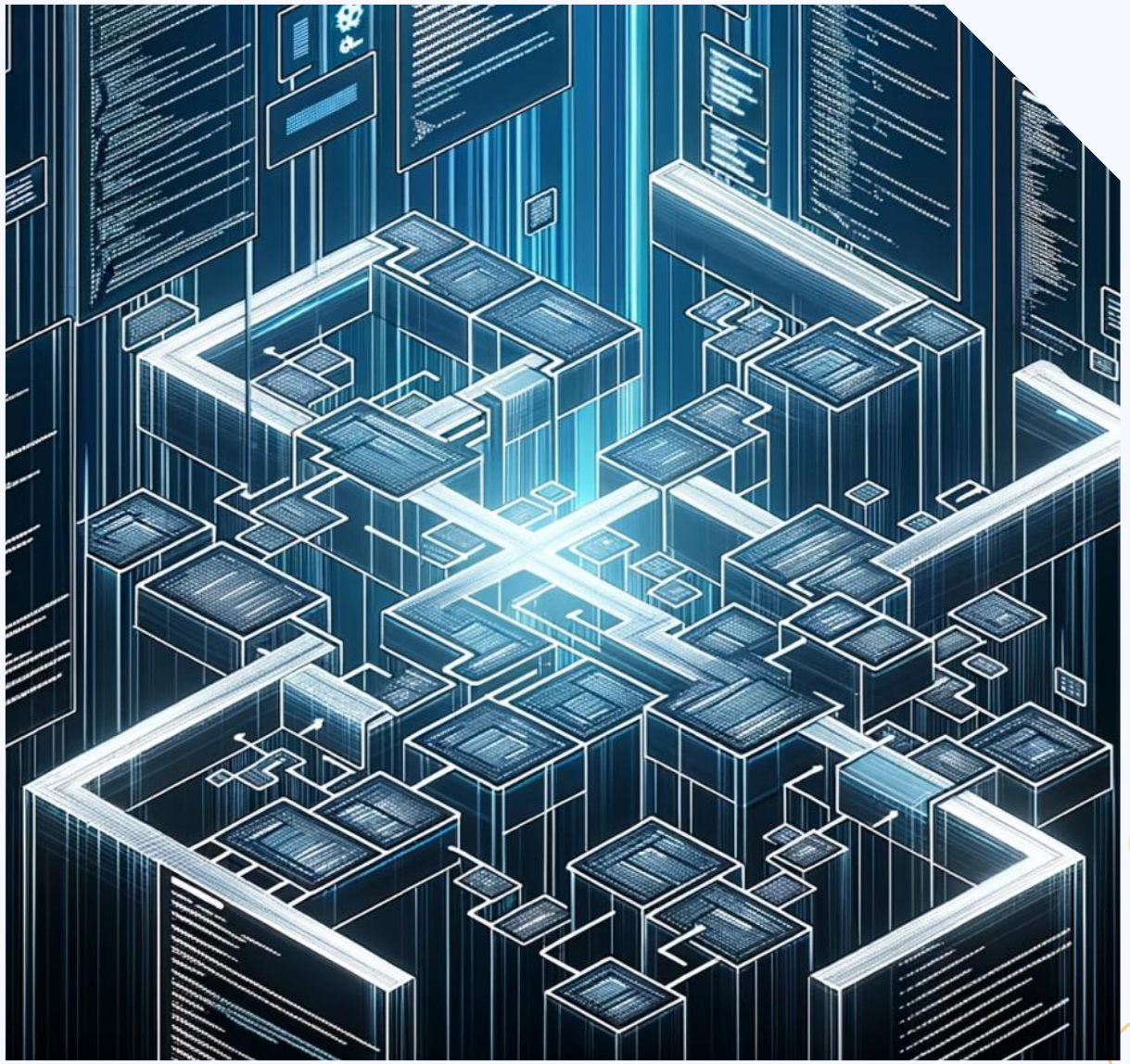
Verificación de algoritmos

Fernanda Potosi Reyes
Emmanuel Paz Cabezas
Helmer Andres Mosquera Lerma

¿Qué es la verificación de algoritmos?

La **verificación de algoritmos** es el proceso de demostrar formal o empíricamente que un algoritmo es lógicamente correcto; es decir, que cumple exactamente con las especificaciones matemáticas y restricciones de diseño para las que fue creado. En términos simples, responde a la pregunta: *"¿Construimos el algoritmo correctamente?"*.



Verificación vs. Validación

Verificación

Comprueba la corrección matemática y lógica de la estructura interna. *"¿Estamos resolviendo la ecuación de la manera correcta?"* (Ej. El algoritmo ordena los datos de menor a mayor sin perder registros).

Validación

Comprueba la utilidad real respecto a las necesidades del mundo real. *"¿Estamos resolviendo el problema correcto?"* (Ej. ¿El sistema realmente necesitaba ordenar estos datos, o el usuario final prefería una barra de búsqueda?).

Métodos de Verificación

Verificación Formal (Demostraciones Matemáticas)
Utiliza lógica estricta para probar que el algoritmo es correcto para *todas* las combinaciones de entrada posibles en el universo, sin necesidad de ejecutarlo de forma empírica.



Verificación Formal (Demostraciones Matemáticas)

1. **Lógica de Hoare:** Define precondiciones (el estado inicial que se exige antes de arrancar) y postcondiciones (el estado final que el algoritmo garantiza entregar).

2. **Invariantes de bucle:** Propiedades matemáticas que se diseñan para mantenerse absolutamente verdaderas antes, durante y después de cada iteración de un ciclo (como un while o un for).

3. **Model Checking (Verificación de Modelos):** Una exploración algorítmica exhaustiva y automatizada de todos los estados posibles que puede alcanzar un sistema concurrente para asegurar que nunca ocurran estados de peligro (ej. interbloqueos o *deadlocks*).

Verificación Empírica (Testing)

Consiste en ejecutar el algoritmo con un conjunto finito de datos de prueba cuidadosamente seleccionados (casos base y casos límite, como números negativos, nulos o matrices vacías). Como advirtió el pionero Edsger Dijkstra: *"Las pruebas pueden demostrar la presencia de errores, pero nunca su ausencia"*.



Análisis Estático

Herramientas de software automatizadas que analizan el código fuente del algoritmo en busca de patrones defectuosos, fugas de memoria o bucles infinitos, inspeccionando la estructura gramatical sin llegar a compilarla ni ejecutarla.



¿Por qué es tan crítica la verificación formal?

Mientras que el algoritmo de recomendación de una red social puede depender solo de *testing* básico, la **verificación formal y exhaustiva** es obligatoria por ley en los **sistemas de misión crítica**. Un error algorítmico no detectado puede ser catastrófico en:



Un error algorítmico no detectado puede ser catastrófico en:



Aviación y Aeroespacial

Algoritmos de Navegación autónoma y control de vuelo.



Criptografía y Finanzas

Contratos inteligentes (*Smart contracts*) de blockchain y protocolos bancarios donde un fallo lógico permite el robo de millones de dólares.



Sistemas Médicos

Software de control de dosificación en máquinas de radioterapia o marcapasos (donde un bucle mal calculado cuesta vidas).

Gracias por su atención