

# Paralelismo en java para todos

Ing. Aluis Marte  
@jconf2019



# UN POCO SOBRE MI

Ingeniero de sistemas y computación egresado de la PUCMM con más de 4 años de experiencia en paralelismo, HPC programando en Java, C y C++. Gerente de proyectos (con metodologías SCRUM y tradicionales), profesor vigente en el TEP PUCMM y amante de los gatos.

Teléfono: +1 (809) 968-7812

GitHub: <https://github.com/aluismarte>

BitBucket:  
<https://bitbucket.org/aluismarte/>

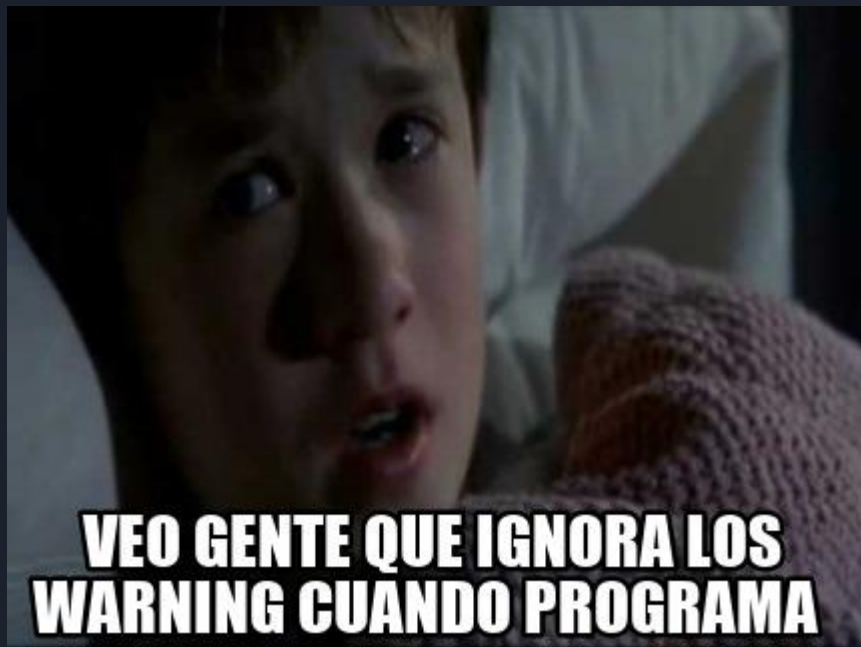
**Odio los Warning**



# Temas a tratar

- Conceptos
  - Concurrente y Paralelo
  - Hilos
  - Piscinas
- ¿Como atacar el problema?
  - Naturaleza
  - Operaciones
  - Tarea o tareas a realizar
- Identificación de las variables y el problema
  - Atomización de los puntos de trabajo
  - Segmentación de las tareas
  - Asignación de trabajo
- Carga y división del trabajo:
  - CPU
  - RAM
  - DISK

Si, toda una hoja para esto



“Mi código funciona así”

“Es solo un warning” -\_-#

“No se otra forma de hacerlo”

Cuidado con el perro. Tranquilo, entra a la casa, es solo un Warning



# Paralelismo y concurrencia

Se tiende a pensar que son lo mismo, pero... ¡No! no son lo mismo, aunque para simplificar la charla, lo asumimos.

**PISCINA DE HILOS:** Es un elemento organizado que maneja hilos bajo criterios dados para aceptar múltiples procesos y estos a su vez trabajan en cola para ser asignados a los hilos disponibles.

**CONCURRENCIA:** 2 o más procesos corren simultáneamente en un mismo núcleo del CPU compartiendo lapsos de tiempo del mismo para dar la impresión que están corriendo el mismo tiempo.

**PARALELISMO:** 2 o más procesos corren al mismo tiempo ejecutándose cada uno en un núcleo independiente.

**HILO:** es simplemente una tarea que puede ser ejecutada al mismo tiempo que otra tarea.



# Una nota de developer

El operador FINAL es vital para proteger los estados de los objetos, evitar null points y evitar que los developers programen sin un estándar...

```
public void exampleFinal(final String data) {  
    final User user = new User();  
    elementUI.setOnClickListener(event -> {  
        user.setName("Hola final");  
        user.setData(data);  
    });  
}
```

Por qué usar final cuando uso hilos?

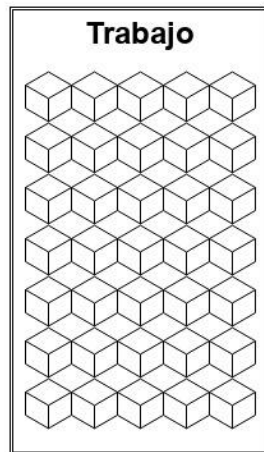
# ¿Como atacar el problema?

La pregunta “¿Más hilos es mejor?” puede que no sea la esperada si tomamos en cuenta estos conceptos:

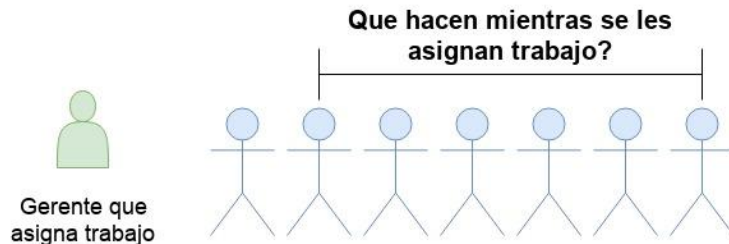
- Tiempo de administración
- Fatiga del CPU
- Fatiga de otros recursos
- Sincronía con otros recursos
- Tipo de datos

El caos quizás no es tan malo y aprender a usarlo a favor puede hacer que tenga orden y no lo aparente.

## ¿Como puedo saber si mi programación es realmente ideal para el problema?



Por que tengo todo el trabajo cargado si le asignare una caja a cada uno?,







¡GRACIAS!

