

# ►Monitoreo Datacenter

Herramienta de monitoreo de energia y temperatura de sala Datacenter.  
Trabajo Final para Prog. Concurrente UCALP – Prof. Ing. Gustavo Cerveri



► 01  
Introducción

► 02  
Objetivo general

► 03  
Infraestructura

► 04  
Comportamiento

► 05  
Conclusión





# ► 01. Introducción

Presentación y propósito de la propuesta

# Nuestro trabajo

- ▶ El presente trabajo práctico consiste en el desarrollo de un **Simulador de Monitoreo para un Centro de Cómputos Crítico**. El sistema tiene como objetivo principal garantizar la **continuidad del negocio** mediante la gestión automatizada de recursos energéticos y ambientales.  
Con una interfaz grafica desarrollada en Java supervisamos de forma constante tres pilares fundamentales: **Transición automática** de la red eléctrica, **Seguridad térmica** y **Trazabilidad de los eventos** (logs)

# El Lenguaje y el Modelo de Conurrencia

- ▶ El proyecto fue desarrollado íntegramente en **Java**, utilizando su API nativa de **Threads**. El modelo de concurrencia implementado es el de **Memoria Compartida Sincronizada**, basado específicamente en el patrón de **Monitores**.

Múltiples hilos (Servidores, Aires, Clima, Energía) acceden a una misma instancia de datos (el MonitorEnergia). Para evitar que estos hilos choquen o corrompan la información, el objeto Monitor actúa como un "árbitro" o guardián.

# Conurrencia

## Hilos de infraestructura

Cada servidor (PROD, NAS, LAB) y cada Aire Acondicionado corre en su propio hilo independiente (Thread). En la realidad un servidor no espera a que el otro termine para funcionar.

Necesitamos que cada uno tome decisiones autónomas basadas en la energía disponible.

## Hilo de ambiente

Un hilo dedicado a calcular la termodinámica (calor generado vs. frío producido), para que la temperatura cambie de forma fluida y constante sin bloquear la interfaz gráfica del usuario.

# Sincronización y Bloqueo

## Métodos synchronized (Exclusión Mutua):

**Ubicación:** En la clase MonitorEnergia (ej: usarUps(), getFuenteActual()).

**Por qué:** Cuando un hilo de servidor entra a consultar la fuente de energía, "bloquea" el monitor. Si el hilo de la UPS intenta restar batería en ese mismo instante, Java lo pone en espera. Esto evita **Condiciones de Carrera** (que dos hilos modifiquen la misma variable al mismo tiempo).

## Mecanismo de Señalización (notifyAll):

**Ubicación:** En métodos como recargarNafta() o setRedElectrica().

**Por qué:** Los hilos de los servidores suelen estar en espera si no hay energía. Al usar notifyAll(), el monitor les "avisa" a todos los hilos que el estado de los recursos cambió, permitiéndoles reaccionar de inmediato.

# Interfaz Gráfica

Un punto de concurrencia critico se encuentra en la actualización de los colores (JLabel)

**Mecanismo:** SwingUtilities.invokeLater.

**Por qué:** Como Swing no es seguro para multihilo, este método asegura que las actualizaciones visuales se ejecuten en el hilo de despacho de eventos (EDT), evitando que la ventana se tilde o se cierre por errores de concurrencia visual.



# ► 02. Objetivo general

Crear entorno estable y seguro

# Objetivo : Ambiente estable y controlado

- ▶ Diseñar e implementar un sistema de monitoreo (dashboard) para un Centro de Datos, utilizando los mecanismos de programación concurrente en Java para gestionar y monitorizar de forma autónoma la redundancia energética, la seguridad térmica y la alta disponibilidad de servicios críticos ante escenarios de falla.

# ► 03. Infraestructura

Servidores y elementos principales en la sala.

# Equipos instalados en la sala

## Servidores

PROD : ambientes productivos, en el mismo rack modem, router, switch y firewalls ya configurados.

LAB : destinado a ambientes testing y similares.

NAS : almacenamiento de recursos para ambientes no productivos.

## Energia

UPS : duración 3 horas (100%)

Generador electrico : duración 24 horas (100%)

## Aires acondicionados

Dos unidades, una sola unidad conectada a la línea del generador eléctrico

# ►04. Comportamiento de infraestructura

Funcionalidades y visualizacion

# Corte de suministro eléctrico

Pasa a cambiarse a la linea conectada al generador, apagandose el aire acondicionado 2, quedando up un aire acondicionado, PROD, LAB y NAS

Cuando el generador de agota (24hs), pasa a usarse la linea de la UPS, se apaga el unico aire acondicionado que esta encendido, para mantenericamente el servidor PROD por 2hs mas.

Cuando se agotan estas dos lineas adicionales se produce una **caida** total del Sistema, se dispara switcheo de ambiente "CABA".

Cuando el suministro electrico vuelve, la UPS empieza a cargar y se encienden todos los servicios. (Aires acondicionados, PROD NAS LAB)

# Redundancia electrica

Aquí se encuentran dos líneas de suministro eléctrico adicional para garantizar la disponibilidad de los distintos servicios.

**El generador** eléctrico, tiene conectados únicamente un aire acondicionado y los tres servidores, lo que permite prolongar el suministro eléctrico por hasta 24 horas. Funciona con nafta, la cual debe cargarse manualmente.

**La UPS**, tiene conectado únicamente el servidor PROD y la red eléctrica, con el objetivo de garantizar dos horas adicionales de funcionamiento para los servicios más importantes de los ambientes productivos. Esta empezara a consumirse cuando este el generador agotado, cuando la UPS no está en uso, sus baterías se mantienen en carga.

# Servidores PROD LAB NAS

Los tres servidores estan conectados a **la linea electrica**, cuando se corta el suministro electrico se pasa a usar la linea electrica del **generador**, cuando este se agota pasa a usarse la linea conectada a la **UPS**, **en ese momento se envian señales de apagado a los server LAB y NAS quedando solamente el servidor PROD conectado a la linea UPS**, garantizando mas tiempo de uptime de los servicios productivos.

# Aires acondicionados

Hay dos unidades en la sala, solo una de ellas esta conectada al generador para consumir menos energia en caso de corte de suministro. Adicional a ellos hay un display de temperatura ambiente

# Alta disponibilidad en emergencia

Si el servicio cae completamente se activa DisasterRecovery, este esta configurado para pasar el servicio productivo (PROD), cambia la ruta DNS al servidor PROD instalado en CABA, redirigiendo el trafico.

# Log de eventos

Se escribirán logs de los distintos eventos que se sucedan, tales como cambios de suministro, apagados/encendidos de equipos, variaciones de temperatura, etc. Pueden utilizarse para generar gráficos estadísticos o ser exportadas para una mejor visualización.

# Temperatura ambiente

La temperatura ambiente de operación estable para el funcionamiento de los servidores es de hasta 30 °C. Superado este valor, se produce un apagado automático de los servidores como medida de precaución. Cuando la temperatura desciende por debajo de los 24 °C, los servidores se encienden nuevamente de forma automática. Todo esto está monitorizado con un display.

# ►05.

## Conclusión

Resultados y recomendaciones

# Obtenemos

## Monitoreo ◀

Suministro electrico,  
temperatura, servidores

## Logs ◀

Estadisticas y monitoreo  
de los eventos



## ► Temperaturas

Max 30\*C , Estable <  
24\*C

## ► Disponibilidad

UPS, Generador,  
Disaster Recovery

# Chequeos

## Siempre verificar

- ▶ Bateria : niveles de carga UPS y generador
- ▶ Carga nafta de Generador / reemplazo baterias UPS
- ▶ Linea de suministro de uso actual
- ▶ Temperaturas de ambiente promedio < 26\*
- ▶ Funcionamiento PROD LAB NAS y aires acc.
- ▶ Disaster Recovery CABA

# Gracias!

Consultas, dudas?

[manu\\_pasquali@hotmail.com](mailto:manu_pasquali@hotmail.com)  
+54 221 6746261  
[jpasquali.com](http://jpasquali.com)

