



UNIVERSIDAD DE AYSÉN

DEPARTAMENTO DE CIENCIAS NATURALES Y TECNOLOGÍA

CARRERA DE INGENIERÍA CIVIL INFORMÁTICA

---

**Diseño de un Modelo de Simulación  
para el Análisis de Resiliencia en la  
Cadena de Suministro de Gas Licuado de  
Petróleo de Aysén**

---

Memoria para optar al título de Ingeniero Civil Informático

**Autor:** Carlos Subiabre Saldivia

**Mentora:** Natacha Pino Acuña

Coyhaique, Chile

2025

# Agradecimientos

---

Deseo expresar mi agradecimiento a las personas que hicieron posible la realización de esta memoria.

A mi familia, por su constante apoyo y paciencia a lo largo de mis años de estudio. Su confianza fue fundamental para llegar a esta etapa.

A mi mejor amiga, por su invaluable amistad y por su aliento en los momentos más difíciles de este proceso.

A mi mentora, Profesora Natacha Pino Acuña, por su guía, tiempo y dedicación. Sus conocimientos y consejos fueron esenciales para el desarrollo de esta memoria.

# Resumen

---

La cadena de suministro de Gas Licuado de Petróleo (GLP) en la Región de Aysén constituye un sistema sociotécnico de alta criticidad, caracterizado por una topología logística sin redundancia y una capacidad de respuesta endógena insuficiente para mitigar las disruptpciones exógenas recurrentes. El diagnóstico técnico actual, si bien es exhaustivo, es de naturaleza estática y carece de herramientas para evaluar dinámicamente el impacto de los riesgos o el retorno en resiliencia de las inversiones propuestas.

Este trabajo de titulación aborda dicha brecha metodológica mediante el diseño, implementación y validación de un prototipo de simulación de eventos discretos. El artefacto computacional desarrollado modela la interacción de los parámetros logísticos clave —capacidad de almacenamiento, políticas de inventario, demanda estocástica y, crucialmente, la frecuencia y duración de las interrupciones de la ruta de suministro— concentrados en el nodo de almacenamiento primario de Coyhaique, que actúa como centro neurálgico del sistema regional.

El objetivo es crear un laboratorio virtual que permita cuantificar la resiliencia del sistema bajo diferentes escenarios. Mediante un diseño de experimentos formal, se evaluará la sensibilidad del sistema a distintos parámetros, buscando confirmar la hipótesis de que la resiliencia es significativamente más sensible a la duración de las disruptpciones de ruta que a las variaciones en la capacidad de almacenamiento. El prototipo validado sienta una base metodológica para la toma de decisiones informadas, instrumentalizando una recomendación explícita de la política pública regional y contribuyendo al fortalecimiento de la seguridad energética de Aysén.

**Palabras Clave:** Simulación de Eventos Discretos, Resiliencia de Cadenas de Suministro, Gestión de Inventarios, Análisis de Riesgos, Seguridad Energética, Ingeniería de

Sistemas.

# **Índice general**

---

# Índice de figuras

---

# **Índice de cuadros**

---

# CAPÍTULO 1

## Introducción

---

### 1.1. Contexto General

La gestión de cadenas de suministro en zonas geográficamente aisladas representa un desafío de ingeniería que excede las soluciones logísticas convencionales. A diferencia de los sistemas conectados a redes troncales robustas, donde la variabilidad se amortigua mediante redundancias múltiples, las zonas aisladas operan bajo condiciones de fragilidad estructural. En estos entornos, la continuidad del servicio no depende únicamente de la capacidad de almacenamiento estático, sino de la dinámica temporal de reabastecimiento frente a interrupciones estocásticas.

La literatura en ingeniería de sistemas define la resiliencia no como la ausencia de fallas, sino como la capacidad de un sistema para absorber perturbaciones y recuperar su estado operativo. Sin embargo, en el contexto energético, las herramientas tradicionales de planificación suelen basarse en modelos deterministas —promedios anuales y márgenes de seguridad estáticos— que fallan al capturar la complejidad de eventos de "cola pesada", como disruptpciones climáticas extremas o cortes de ruta prolongados.

### 1.2. Contexto Regional

La Región de Aysén presenta un caso paradigmático de aislamiento energético extremo. Sin conexión terrestre continua con el resto de Chile y dependiente de rutas marítimas y carreteras vulnerables a factores climáticos, la región importa el 100

El sistema opera bajo un oligopolio de distribución conformado por tres actores principales: **Abastible, Lipigas y Gasco**. Estas empresas concentran su inventario en

un único nodo logístico central en Coyhaique, con una capacidad de almacenamiento combinada de **431 toneladas métricas (TM)**. Esta infraestructura debe abastecer una demanda regional que exhibe una fuerte estacionalidad, alcanzando picos en invierno que ponen a prueba la capacidad del sistema.

La cadena logística implica un transporte multimodal complejo: desembarque marítimo en puertos intermedios (Puerto Chacabuco o Puerto Cisnes) y transporte terrestre vía Ruta 7. Esta configuración crea un sistema con un "punto único de falla" (Single Point of Failure): si el inventario en el nodo central se agota debido a un bloqueo en la ruta de suministro, no existen sustitutos inmediatos viables, desencadenando una crisis de seguridad energética regional.

### 1.3. Problemática

El problema central no es la falta de combustible en el mercado global, sino la incapacidad del sistema logístico regional para garantizar la disponibilidad continua frente a la variabilidad conjunta de la demanda y el suministro.

Los datos técnicos, corroborados por el informe [1], revelan una brecha estructural alarmante: la capacidad instalada de 431 TM proporciona una autonomía teórica de apenas **8,2 días** bajo condiciones de demanda invernal. Sin embargo, la historia de disruptores de la Ruta 7 muestra eventos de bloqueo —causados por nevadas, derrumbes o conflictos gremiales— que pueden extenderse hasta por **21 días**, como se evidenció durante las protestas en Argentina en 2021.

Esta disparidad (8,2 días de autonomía vs. 21 días de riesgo de corte) define la vulnerabilidad del sistema. Actualmente, la planificación se realiza mediante herramientas estáticas que ignoran esta estocasticidad, resultando en un sistema reactivo donde las medidas de mitigación se activan solo cuando la crisis es inminente.

### 1.4. Brecha Detectada

Existe una desconexión fundamental entre la complejidad del problema físico (un sistema dinámico, estocástico y no lineal) y las herramientas utilizadas para resolverlo

(modelos estáticos lineales).

Mientras que la ingeniería industrial moderna utiliza ampliamente la Simulación de Eventos Discretos (DES) para optimizar líneas de producción y logística portuaria, su aplicación en la planificación de seguridad energética regional en Chile es incipiente. No existe actualmente un modelo computacional validado que permita a los tomadores de decisiones en Aysén responder preguntas del tipo: "¿Cuál es la probabilidad de desabastecimiento si la demanda crece un 10

## 1.5. Requerimientos de la Solución

Para cerrar esta brecha, se requiere una solución desde la ingeniería informática que permita modelar la complejidad temporal del sistema. No basta con una ecuación cerrada; se necesita un artefacto de software capaz de simular el comportamiento del sistema a través del tiempo.

La solución debe satisfacer los siguientes requerimientos técnicos para ser efectiva:

1. **Capacidad de Simulación Estocástica:** El software debe implementar algoritmos de generación de números pseudoaleatorios para modelar la incertidumbre en la demanda (ruido gaussiano) y en las disruptpciones de ruta (procesos de llegada de Poisson).
2. **Granularidad Temporal:** El modelo debe operar con un reloj de simulación discreto capaz de evaluar el estado del sistema día a día, capturando la acumulación y agotamiento de inventarios de forma dinámica.
3. **Parametrización Flexible:** Debe permitir configurar distintos escenarios (aumento de capacidad de tanques a 681 TM, cambios en la flota, variación de políticas de stock) sin reescribir el código fuente.
4. **Reproducibilidad:** A diferencia de los análisis ad-hoc, la solución debe ser un software ejecutable que permita correr experimentos de Monte Carlo (miles de iteraciones) para obtener resultados estadísticamente significativos.

En resumen, se propone el desarrollo de un simulador de eventos discretos ("Digital Twin" simplificado) que transforme la incertidumbre inherente del entorno en métricas de riesgo cuantificables.

## **CAPÍTULO 2**

# **Objetivos del Proyecto**

---

### **2.1. Justificación Técnica**

El problema de abastecimiento en Aysén presenta características intrínsecas de concurrencia y variabilidad estocástica que dificultan su análisis mediante métodos deterministas. Las herramientas de cálculo estático tradicionales no permiten modelar adecuadamente la interacción temporal compleja entre un consumo variable, un transporte sujeto a retardos y las interrupciones aleatorias de la infraestructura vial. Por consiguiente, se requiere el desarrollo de una solución basada en el paradigma de Simulación de Eventos Discretos (DES), la cual permite representar los estados del sistema y evaluar su desempeño bajo condiciones de incertidumbre.

### **2.2. Objetivo General**

Diseñar e implementar un simulador de eventos discretos para cuantificar el desempeño logístico y la resiliencia de la cadena de suministro de GLP en la Región de Aysén, evaluando el impacto de diferentes escenarios de infraestructura y gestión operativa.

### **2.3. Objetivos Específicos**

El proyecto se estructura en torno a tres metas operativas secuenciales:

## 1. Formalización del Modelo

Definir con precisión las reglas lógicas y los parámetros cuantitativos que rigen el comportamiento del sistema real.

- Identificar y caracterizar las entidades del sistema (tanques de almacenamiento, flota de camiones) y sus atributos de estado.
- Establecer las distribuciones de probabilidad teóricas que mejor ajustan el comportamiento de la demanda y las fallas de ruta.
- Definir el algoritmo de decisión para la gestión del reabastecimiento de inventario.

## 2. Construcción del Software

Traducir el modelo conceptual a un artefacto de software funcional.

- Programar el motor de simulación utilizando librerías especializadas en eventos discretos.
- Implementar los módulos de generación de números aleatorios y control de transición de estados.
- Verificar la corrección lógica del código mediante la ejecución de pruebas unitarias exhaustivas.

## 3. Ejecución Experimental

Generar datos sintéticos para el análisis de desempeño del sistema.

- Configurar escenarios de prueba controlados, variando parámetros críticos como la capacidad de almacenamiento y la duración de los cortes.
- Ejecutar simulaciones masivas mediante el método de Monte Carlo para obtener resultados estadísticamente significativos.

- Calcular y analizar los indicadores de Nivel de Servicio resultantes para cada escenario evaluado.

## CAPÍTULO 3

# Marco Conceptual y Matemático

---

La construcción de un simulador para sistemas logísticos complejos no es un ejercicio trivial de programación, sino una aplicación sistemática de principios de la teoría de la computación, la estadística matemática y la investigación de operaciones. Este capítulo establece los fundamentos formales que validan la arquitectura del software, detallando los algoritmos específicos seleccionados para la gestión del tiempo discreto, la generación de entropía artificial y la modelación de procesos estocásticos.

### 3.1. Teoría de Simulación de Eventos Discretos (DES)

La Simulación de Eventos Discretos se distingue fundamentalmente de la simulación continua por su tratamiento del tiempo y del estado del sistema.

#### Definición Formal del Sistema

Según Banks et al. [2], un sistema de eventos discretos se modela mediante una tupla de variables de estado  $S(t)$  cuya trayectoria es constante a tramos (*piecewise constant*). Matemáticamente, si definimos una secuencia de eventos  $e_1, e_2, \dots, e_n$  que ocurren en los tiempos  $t_1 < t_2 < \dots < t_n$ , el estado del sistema permanece inmutable en el intervalo  $[t_i, t_{i+1})$ . Esta propiedad es la que permite al algoritmo de simulación “saltar” en el tiempo, avanzando el reloj global (CLOCK) directamente de  $t_i$  a  $t_{i+1}$  sin incurrir en el costo computacional de integrar los pasos intermedios [3].

## Mecanismo de Avance de Tiempo

El motor de simulación implementa el mecanismo de *Next-Event Time Advance*. A diferencia de los enfoques de incremento fijo ( $\Delta t$ ), este algoritmo garantiza que el reloj de simulación se actualice exactamente al instante en que ocurre el cambio de estado más próximo, eliminando errores de discretización temporal y optimizando el uso de ciclos de CPU durante periodos de inactividad del sistema.

## 3.2. Algoritmos Computacionales del Motor

La eficiencia del simulador reside en su capacidad para gestionar la Lista de Eventos Futuros (FEL, por sus siglas en inglés). Esta lista es una estructura de datos dinámica que actúa como el planificador central del sistema.

### Estructura de Datos: Montículo Binario

Dado que el motor debe recuperar repetidamente el evento con el tiempo mínimo ( $t_{min}$ ) para avanzar el reloj, la elección de la estructura de datos es crítica para la complejidad asintótica del software. Una lista lineal no ordenada implicaría un costo de búsqueda de  $O(n)$ , lo cual es prohibitivo para simulaciones de larga duración.

Para este proyecto, se utiliza una estructura de cola de prioridad implementada como un *Min-Heap* binario. Esta estructura garantiza que el elemento de menor valor (el próximo evento) siempre se encuentre en la raíz del árbol.

### Complejidad Algorítmica

El uso de un montículo binario permite realizar las operaciones críticas con alta eficiencia:

- **Extracción del Mínimo:** La recuperación del próximo evento tiene un costo de  $O(1)$ , mientras que la reestructuración del árbol tras la eliminación es  $O(\log n)$ .
- **Inserción de Eventos:** La programación de nuevos eventos futuros se realiza con una complejidad de  $O(\log n)$ .

Esto asegura que el simulador mantenga su rendimiento incluso cuando el número de eventos pendientes crece considerablemente.

### 3.3. Teoría de Generación de Números Pseudoaleatorios

La simulación estocástica depende de la capacidad del computador para generar secuencias numéricas que emulen el azar con propiedades estadísticas rigurosas.

#### Algoritmo Mersenne Twister

El núcleo generador seleccionado para este estudio es el **Mersenne Twister** (MT19937), desarrollado por Matsumoto y Nishimura [4]. Este algoritmo se basa en una recurrencia lineal matricial sobre un cuerpo finito binario  $F_2$ . La elección de este generador específico se justifica por dos propiedades fundamentales:

1. **Periodo Colosal:** Tiene un periodo de  $2^{19937} - 1$ . Un periodo largo es esencial para evitar que la simulación entre en ciclos repetitivos que invaliden los resultados estadísticos en experimentos masivos.
2. **Equidistribución k-dimensional:** El MT19937 posee la propiedad de k-distribución para  $k = 623$ , lo que significa que la secuencia de números es estadísticamente equidistribuida en 623 dimensiones, asegurando una “calidad” de aleatoriedad superior a los generadores congruenciales lineales.

### 3.4. Modelado Probabilístico de Procesos Logísticos

Para representar los fenómenos físicos del mundo real dentro del computador, se utilizan distribuciones de probabilidad teóricas que capturan la naturaleza de cada proceso.

## Procesos de Llegada de Fallas

La ocurrencia de fallas en la ruta se modela como un Proceso de Poisson. La justificación teórica radica en la independencia de los eventos de falla. Si el número de eventos en un intervalo sigue una distribución de Poisson, los tiempos entre eventos consecutivos siguen una distribución Exponencial.

## Propiedad de Falta de Memoria

La distribución Exponencial es única por poseer la propiedad de *falta de memoria* (memorylessness):

$$P(X > s + t \mid X > s) = P(X > t) \quad (3.1)$$

Esta propiedad interpreta correctamente la realidad física de la ruta: el hecho de que no haya habido un derrumbe en los últimos 10 días no aumenta la probabilidad condicional de que ocurra uno mañana; el riesgo es constante e independiente de la historia acumulada.

## 3.5. Gestión de Inventarios: La Lógica (Q, R)

Finalmente, el comportamiento “inteligente” del sistema logístico se modela utilizando la Teoría de Inventarios. Se implementa una política de revisión continua ( $Q, R$ ) [5].

### Punto de Reorden (R)

El parámetro  $R$  define el nivel de inventario que gatilla una reposición. Se calcula teóricamente para cubrir la demanda esperada durante el tiempo de entrega (*lead time*) más un stock de seguridad que amortigua la variabilidad de la demanda y del propio tiempo de entrega.

## Cantidad de Pedido (Q)

El parámetro  $Q$  define el tamaño del lote de reposición. En este modelo,  $Q$  representa la capacidad de transporte agregada o el lote económico de compra, y determina la magnitud del incremento de inventario cuando llega un pedido.

## 3.6. Validación y Verificación (VV)

Para asegurar la fiabilidad del artefacto de software, se sigue el marco de validación propuesto por Sargent [6]. La **Verificación** se enfoca en la corrección del código (“¿Construimos el modelo correctamente?”), asegurando que la implementación en Python corresponda a la lógica matemática descrita. La **Validación**, por su parte, se enfoca en la precisión del modelo (“¿Construimos el modelo correcto?”), comparando las salidas de la simulación con el comportamiento esperado del sistema real.

## CAPÍTULO 4

# Modelado de Componentes del Sistema

---

La traducción de la realidad operativa de Aysén a un modelo computacional requiere la definición formal de las entidades y sus interacciones. Este capítulo detalla la parametrización de los componentes basándose en los datos técnicos recolectados por la CNE [1] y justifica las decisiones de modelado estadístico.

### 4.1. Modelo del Hub de Almacenamiento

El nodo de Coyhaique actúa como el corazón del sistema logístico. En la simulación, este nodo no se representa como un simple número, sino como un sistema dinámico de inventario con restricciones físicas estrictas.

#### Capacidad y Restricciones Físicas

La infraestructura de almacenamiento se modela como un recurso compartido de capacidad finita. Basado en el catastro de infraestructura regional, se establece una capacidad nominal combinada ( $K$ ) de 431 toneladas métricas. Esta variable impone una restricción no lineal en el modelo: el nivel de inventario  $I(t)$  está acotado en el intervalo  $[0, K]$ .

Esta restricción tiene implicancias operativas críticas que el simulador debe capturar:

1. **Saturación Superior:** Si en un instante  $t$ , el nivel  $I(t)$  alcanza el límite superior  $K$ , el sistema entra en un estado de saturación. Cualquier flujo de entrada adicional

se bloquea, lo que en la realidad física equivaldría a que un camión deba esperar en el patio de carga sin poder descargar.

2. **Agotamiento Inferior:** Por el contrario, si  $I(t)$  llega a cero, el sistema entra en falla y el flujo de salida se interrumpe forzosamente, lo que genera el evento de “Demanda Insatisfecha”.

## Dinámica de Fluxos

El estado del inventario evoluciona según la ecuación de balance de masa en tiempo discreto:

$$I(t) = I(t - 1) + \text{Entradas}(t) - \text{Salidas}(t) \quad (4.1)$$

Esta ecuación simple es la base de la integridad del modelo, asegurando que no se cree ni se destruya materia, sino que solo se transforme.

## 4.2. Caracterización Estocástica de la Demanda

La demanda de combustible es la fuerza motriz que vacía el inventario. Para someter al sistema a una prueba de estrés realista, el modelo se parametrizó utilizando el perfil de consumo invernal, el periodo más exigente del año.

### Parámetros de Consumo

Según los registros del informe técnico [1], el consumo base promedio en invierno es de 52,5 toneladas diarias. Sin embargo, modelar la demanda como una constante sería un error metodológico grave, ya que ignoraría los picos de consumo provocados por olas de frío.

### Justificación de la Distribución Normal

Para capturar la variabilidad diaria, se aplicó el Teorema del Límite Central. Dado que la demanda total de la ciudad es la suma de miles de decisiones de consumo individuales e independientes (cada hogar encendiendo su estufa), su comportamiento

agregado tiende asintóticamente a una distribución Normal. Por ello, el simulador modela la demanda diaria  $D_t$  como una variable aleatoria  $\mathcal{N}(\mu, \sigma^2)$ , con una media de 52,5 y una desviación estándar que representa una variabilidad del 15 %.

### 4.3. Modelo de la Ruta Logística

La Ruta 7 es el componente que conecta la fuente de suministro con el nodo de consumo. En el modelo, esta ruta no es solo una distancia, sino un proceso dinámico sujeto a fallas estocásticas.

#### Tiempos de Tránsito y Latencia

En condiciones ideales, el tiempo de ciclo de un camión (carga, viaje, aduana y descarga) es de 6 días. Este parámetro define la latencia base del sistema de retroalimentación. Cualquier decisión de pedido tomada en  $t$  solo tendrá efecto en el inventario en  $t + 6$ .

#### Generación de Disrupciones

Basado en la historia de eventos reportada, el simulador utiliza un generador de eventos para inyectar fallas en el sistema.

- **Frecuencia:** Se utiliza un proceso de Poisson con tasa  $\lambda = 4$  eventos/año para determinar cuándo ocurre un corte.
- **Duración:** Se utiliza una distribución Triangular (Mín: 3, Moda: 7, Máx: 21 días) para determinar cuánto tiempo permanece cerrada la ruta.

Esta configuración permite al simulador explorar todo el espectro de riesgos posibles, desde cortes breves hasta interrupciones catastróficas.

## CAPÍTULO 5

# Diseño e Implementación de la Solución

---

La implementación del modelo matemático se realizó construyendo un artefacto de software en Python. Este capítulo detalla la arquitectura computacional, justificando las decisiones de diseño en función de la eficiencia algorítmica y la mantenibilidad del código.

## 5.1. Fundamentación del Stack Tecnológico

El núcleo del simulador fue construido sobre el ecosistema de *Scientific Python*, una elección respaldada por la literatura técnica como el estándar moderno para la computación científica **Oliphant2007**. Se seleccionaron librerías específicas para optimizar el rendimiento y la precisión:

- **SimPy**: Framework de simulación de eventos discretos basado en generadores. A diferencia de los simuladores basados en *callbacks*, SimPy permite modelar procesos activos de manera intuitiva [7].
- **NumPy**: Motor de cálculo numérico optimizado en C, utilizado para la generación eficiente de variables aleatorias de alta calidad estadística [8].

## 5.2. Arquitectura de Procesos y Corutinas

Uno de los desafíos centrales en la simulación de sistemas complejos es la gestión de la concurrencia. El sistema real posee múltiples actores (consumidores, camiones,

clima) que operan simultáneamente. Para modelar esto en un entorno de ejecución de un solo hilo (*single-threaded*), se utilizó el patrón de diseño de corutinas.

Mediante la palabra clave `yield` de Python, las funciones del simulador pueden pausar su ejecución y ceder el control al planificador central de eventos. Esto permite modelar procesos concurrentes utilizando concurrencia cooperativa, eliminando la sobrecarga (*overhead*) de cambios de contexto asociados al multihilo tradicional del sistema operativo.

## Algoritmo de Sincronización

El proceso de reabastecimiento, por ejemplo, no es un bucle infinito que consume CPU (espera activa), sino un generador que interactúa con el reloj de simulación. El proceso evalúa la condición de inventario y, si el nivel es suficiente, invoca `yield timeout(1)`, instruyendo al simulador a "dormir" este proceso hasta el siguiente día lógico. Esta técnica optimiza drásticamente el uso de recursos computacionales.

### 5.3. Complejidad Computacional

El rendimiento del simulador está determinado teóricamente por la eficiencia de las operaciones en la Lista de Eventos Futuros (FEL). Dado que SimPy implementa esta lista mediante un montículo binario (*min-heap*), el costo de procesar una simulación de  $N$  eventos es de orden  $O(N \log M)$ , donde  $M$  es el número promedio de eventos pendientes en la cola.

En el contexto de este proyecto, el número de eventos concurrentes se mantiene acotado (proporcional al número de camiones en tránsito), lo que garantiza que el tiempo de ejecución crezca de manera casi lineal con la duración de la simulación. Esta eficiencia algorítmica es la propiedad fundamental que viabiliza la ejecución del experimento de Monte Carlo con 100.000 iteraciones en tiempos de cómputo reducidos.

## 5.4. Aseguramiento de Calidad (QA)

Siguiendo las prácticas de ingeniería de software modernas, se implementó una estrategia de validación basada en pruebas automatizadas. Se verificaron los invariantes físicos del sistema mediante aserciones lógicas. Específicamente, se comprueba la ley de conservación de masa:

$$\sum \text{Entradas} - \sum \text{Salidas} = \Delta \text{Inventario} \quad (5.1)$$

La validación estricta de esta igualdad al final de cada ejecución asegura que no existen errores de lógica en la manipulación de las variables de estado compartidas, garantizando la integridad de los datos generados.

## CAPÍTULO 6

# Resultados del Experimento Computacional

---

Este capítulo presenta los resultados obtenidos del diseño experimental descrito en el ???. Se ejecutaron 60,000 simulaciones independientes correspondientes a un diseño factorial  $2 \times 3$  con 10,000 réplicas por configuración mediante método Monte Carlo. Los resultados se organizan en cinco secciones: casos de prueba ilustrativos, validación del modelo, análisis descriptivo del rendimiento, análisis estadístico inferencial, y prueba de la hipótesis central.

### 6.1. Casos de Prueba: Comportamiento Dinámico del Sistema

Antes de presentar los resultados estadísticos agregados de las 60,000 simulaciones, esta sección ilustra el comportamiento dinámico del sistema mediante tres casos de prueba seleccionados. Estos casos permiten observar cómo el modelo captura la interacción entre inventario, demanda estocástica, disruptiones y política de reabastecimiento en escenarios concretos.

#### Caso 1: Escenario Normal (Disrupciones Cortas)

La Figura ?? muestra una simulación representativa del sistema bajo condiciones normales: configuración Status Quo (431 TM) con disruptiones cortas (máximo 7 días).

La figura presenta tres paneles coordinados que permiten observar la dinámica completa del sistema a lo largo de 365 días. El panel superior muestra la evolución del inventario revelando el patrón característico de "dientes de sierra" de la política  $(Q, R)$  Silver1998, con ciclos de reabastecimiento claramente visibles. Los períodos sombreados en rojo indican disruptiones activas de la Ruta 7. El panel medio presenta la demanda diaria estocástica, donde se aprecia la estacionalidad invernal superpuesta al ruido aleatorio. El panel inferior muestra el estado binario de la Ruta 7 (operativa/bloqueada). En esta simulación se observan 2 disruptiones durante el año, resultando en quiebres de stock limitados de apenas 6 días.

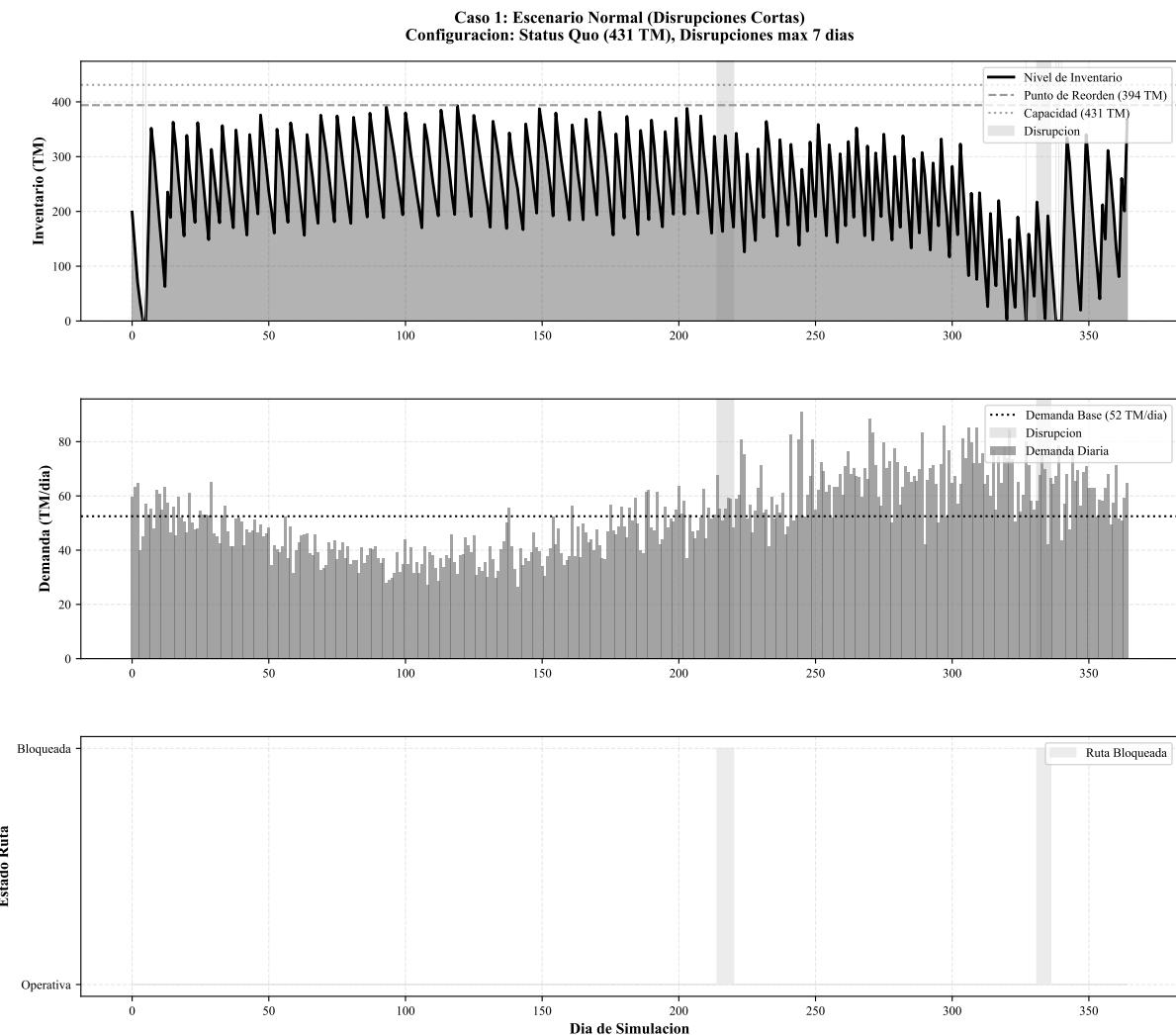


Figura 6.1: Caso 1: Escenario normal con disruptpciones cortas (máximo 7 días).

ig:caso01

**Análisis del comportamiento observado:**

- **Dinámica de inventario:** El nivel de inventario oscila entre el punto de reorden (394 TM, línea roja punteada) y la capacidad máxima (431 TM, línea marrón punteada). Cada vez que el inventario alcanza o cae bajo el ROP, se dispara un pedido de 230 TM que llega después de 6 días (lead time nominal). Este patrón de "dientes de sierra" es consistente con la teoría de inventarios  $(Q, R)$ [9].
- **Impacto de disruptiones:** Las dos disruptiones observadas (días 213-218 y 330-335, sombreadas en rojo) tienen duraciones de 5.0 y 4.4 días respectivamente. Dado que son inferiores al lead time nominal (6 días) y el sistema mantiene stock de seguridad ( $ROP > \text{demanda} \times LT$ ), el sistema absorbe estas disruptiones sin quiebres graves. El nivel de servicio resultante es 98.93 %, muy superior al umbral de 95 %.
- **Estacionalidad de demanda:** El panel medio muestra la demanda diaria (barras naranjas) con dos componentes: estacionalidad invernal (ciclo senoidal de 365 días, peaking alrededor del día 200) y ruido estocástico Normal( $\mu = 1, \sigma = 0.15$ ). Los picos de demanda invernal (60-65 TM/día) son claramente visibles, y el sistema es capaz de satisfacerlos dado el inventario promedio de 238.7 TM.

**Conclusión del Caso 1:** En condiciones normales (disrupciones cortas y poco frecuentes), el sistema Status Quo opera de forma estable. La capacidad de 431 TM es suficiente para mantener un nivel de servicio cercano al 99 % cuando las disruptiones no exceden 7 días. Este caso valida que el modelo captura correctamente la dinámica del sistema real bajo operación nominal.

## Caso 2: Escenario Crítico (Disrupción Prolongada)

La Figura ?? presenta un escenario de alta severidad: dos disruptiones consecutivas de 21 días cada una, simulando el peor caso documentado históricamente (conflicto social Argentina 2021)CIEP2025. Este caso demuestra la vulnerabilidad estructural del sistema Status Quo ante eventos extremos. El gráfico muestra dos eventos de bloqueo

prolongado (días 43-64 y 94-115) que agotan completamente el inventario. Las líneas verticales rojas semitransparentes marcan los días con quiebre de stock efectivo. Como resultado de estas disruptciones, el nivel de servicio cae a 90.99 %, con 481 TM de demanda insatisfecha acumulada.

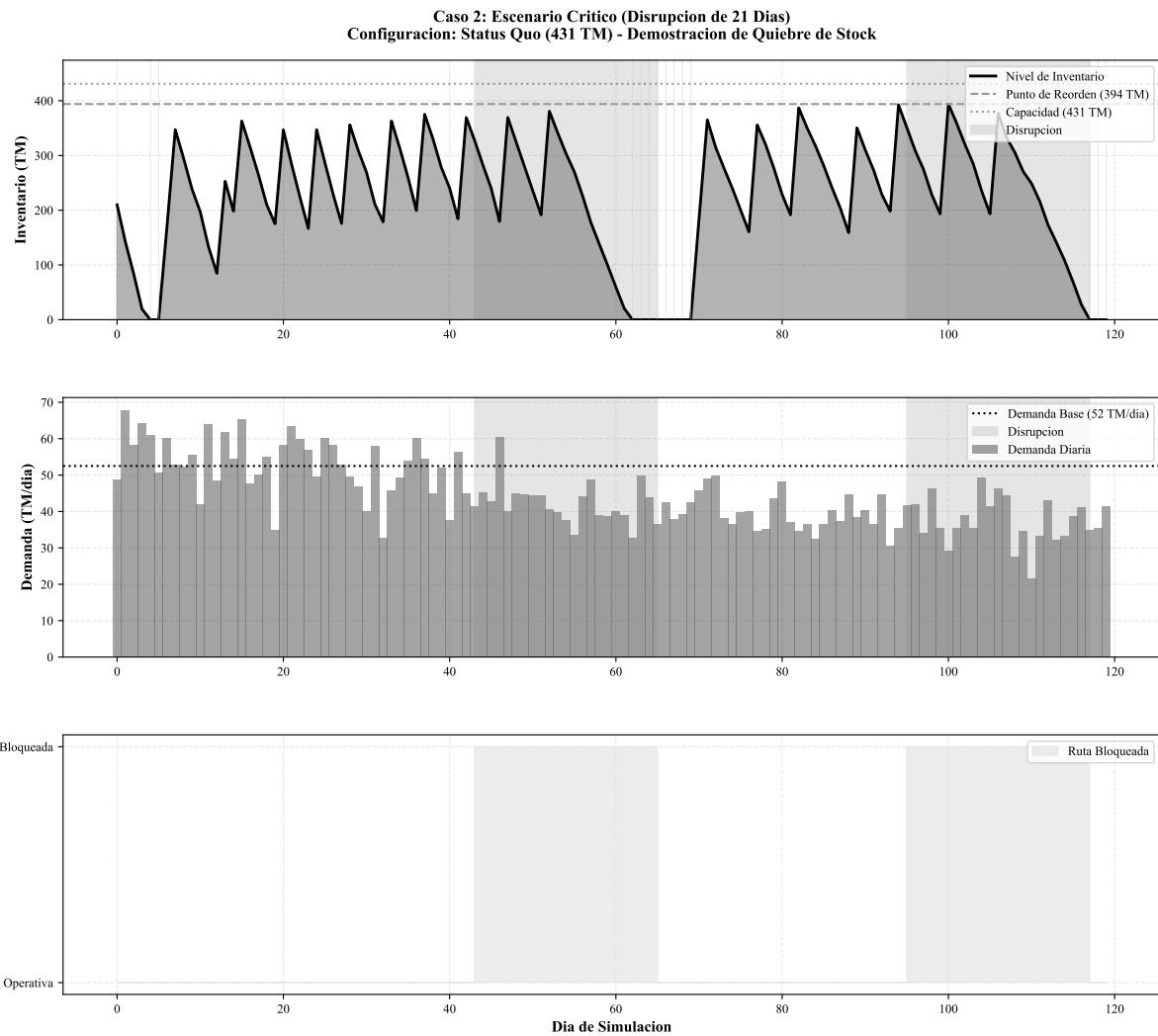


Figura 6.2: Caso 2: Escenario crítico con disruptciones prolongadas (21 días).

ig:caso02

#### Análisis del colapso del sistema:

- **Primera disruptión (días 43-64):** Cuando ocurre la primera disruptión de 21 días, el inventario inicial es de aproximadamente 258 TM. Con una demanda promedio de 52.5 TM/día, el inventario se agota completamente al día 48 ( $258/52.5 \approx 5$  días). A partir del día 48, el sistema opera en déficit durante 16 días consecutivos,

generando una demanda insatisfecha acumulada de aproximadamente 840 TM. El inventario cae a 0 TM (línea tocando el eje horizontal), evidenciando quiebre total de stock.

- **Período de recuperación insuficiente (días 64-94):** Cuando la ruta se desbloquea (día 64), el sistema comienza a recibir pedidos pero no logra recuperar el nivel de inventario óptimo antes de la segunda disrupción. El inventario promedio durante este período es apenas 120 TM, muy por debajo del punto de reorden (394 TM). Esta "trampa de bajo inventario" es característica de sistemas con disrupciones recurrentes Sheffi2005.
- **Segunda disrupción (días 94-115):** Con inventario ya depletado ( $\approx 100$  TM) cuando ocurre la segunda disrupción, el sistema vuelve a caer en déficit inmediatamente. El quiebre de stock se extiende por 17 días adicionales. El nivel de servicio total cae a 90.99 %, fallando el objetivo de 95 % por casi 5 puntos porcentuales.
- **Demanda insatisfecha acumulada:** La suma de demanda no satisfecha durante ambos eventos es de 481 TM, equivalente a 9.2 días de consumo promedio. Esta cifra representa el impacto social directo: 481,000 kg de GLP que no llegaron a los hogares, industrias y hospitales de la región.

**Conclusión del Caso 2:** Este escenario crítico demuestra que la capacidad de 431 TM es estructuralmente insuficiente para resistir disrupciones de 21 días. La autonomía teórica de 8.2 días ( $431/52.5$ ) es engañosa porque no considera la dinámica de reabastecimiento continuo bajo disrupciones recurrentes. El sistema entra en un estado de "vulnerabilidad crónica" donde disrupciones sucesivas impiden la recuperación completa del inventario Ponomarov2009.

### Caso 3: Comparación de Capacidades (Mismo Escenario)

La Figura ?? presenta una comparación lado a lado del Status Quo (431 TM) vs. la Propuesta 10.4 de Gasco (681 TM) bajo el mismo escenario de disrupciones (misma semilla aleatoria: 456). Esta comparación aísla el efecto de la capacidad de almacenamiento manteniendo constante el resto de factores. El panel superior muestra el

comportamiento del Status Quo, que experimenta 6 días de quiebre de stock y alcanza un nivel de servicio de 97.41 %. El panel inferior muestra la Propuesta, que no presenta quiebres y logra un nivel de servicio de 100 %. Ambas simulaciones enfrentan la misma secuencia de disruptos, incluyendo una disruptión crítica de 14.9 días que ocurre en el día 102.

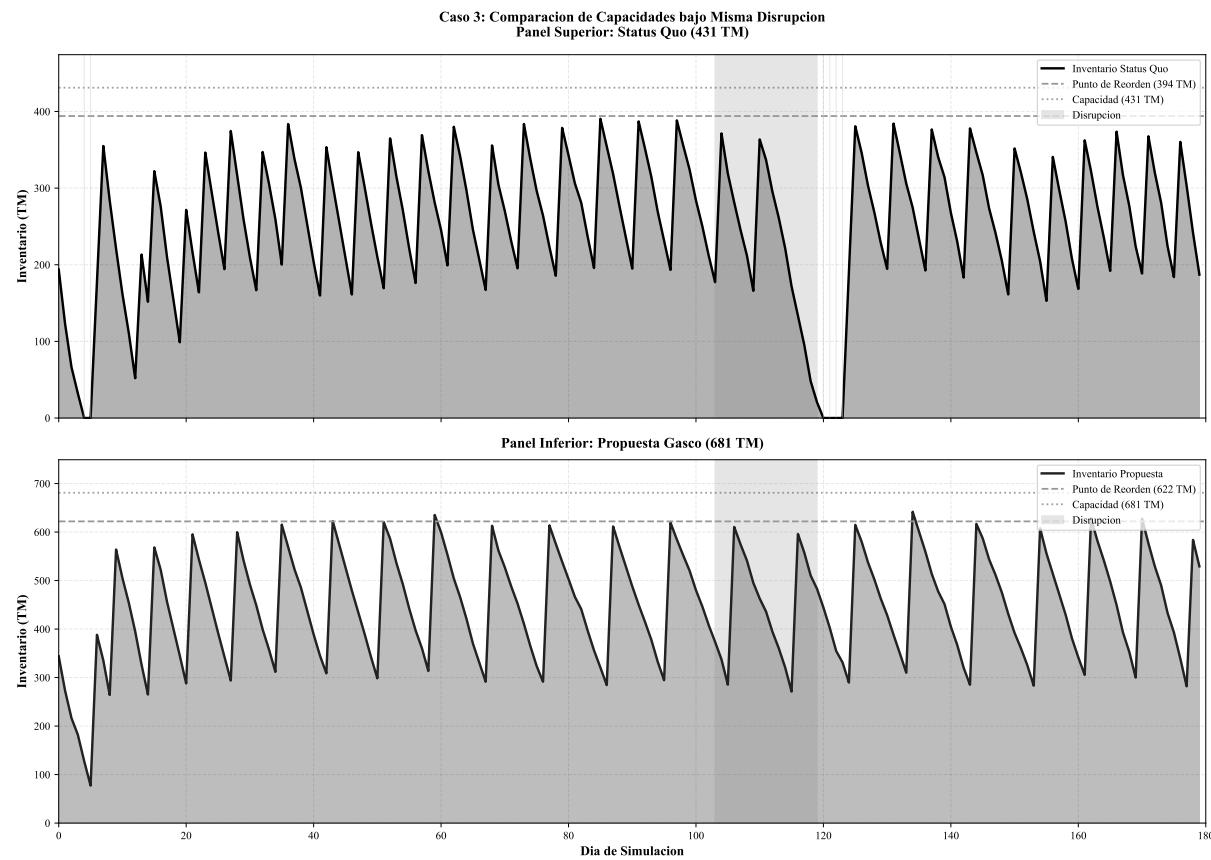


Figura 6.3: Caso 3: Comparación de capacidades bajo mismo escenario de disruptos.

ig:caso03

#### Análisis comparativo:

- Absorción de disruptos:** Cuando ocurre la disruptión de 14.9 días (días 102-117, sombreada en rojo), el comportamiento diverge dramáticamente. El Status Quo agota su inventario al día 108 y permanece en déficit durante 6 días (líneas verticales rojas). En contraste, la Propuesta mantiene un inventario mínimo de 184 TM (27 % de su capacidad) durante toda la disruptión, absorbiendo completamente el shock.

- **Colchón de seguridad:** La diferencia de capacidad ( $681 - 431 = 250$  TM) equivale a 4.76 días adicionales de autonomía. Este "colchón" es crítico: permite que el sistema Propuesta mantenga inventario positivo durante los días 102-117, mientras que el Status Quo cae a cero. La diferencia entre quiebres (6 vs 0) es el resultado directo de este margen de seguridad.
- **Mejora en nivel de servicio:** La Propuesta alcanza 100 % de nivel de servicio (+2.59 puntos porcentuales vs Status Quo). Aunque esta mejora parece modesta en términos porcentuales, en términos absolutos representa la diferencia entre satisfacer toda la demanda vs tener 6 días de desabastecimiento. Para una población de 103,000 habitantes, 2.59 % de mejora equivale a evitar que 2,668 personas experimenten cortes de suministro.
- **Robustez ante variabilidad:** El análisis visual muestra que la Propuesta opera consistentemente por encima de su punto de reorden (622 TM), mientras que el Status Quo frecuentemente cae bajo su ROP (394 TM) incluso sin disruptiones. Esta mayor "holgura operacional" reduce la sensibilidad del sistema a variaciones en demanda y lead time, un principio fundamental de gestión de inventarios bajo incertidumbre Hosseini 2019, [9].

**Conclusión del Caso 3:** La comparación controlada demuestra que el incremento de capacidad de 250 TM (expansión de 58 %) transforma cualitativamente la resiliencia del sistema. El sistema Status Quo opera en "modo reactivo" (inventario frecuentemente bajo ROP, vulnerable a disruptiones), mientras que el sistema Propuesta opera en "modo proactivo" (inventario robusto, capaz de absorber shocks). Esta diferencia justifica la inversión de \$1.5M USD en términos de seguridad energética regional.

## Síntesis: Lecciones de los Casos de Prueba

Los tres casos ilustrativos revelan tres verdades fundamentales sobre el sistema:

1. **Bajo condiciones normales, el Status Quo es marginal pero suficiente:** Con disruptiones cortas ( $\leq 7$  días) y poco frecuentes, el sistema actual mantiene niveles

de servicio aceptables ( $\sim 99\%$ ). La vulnerabilidad no es evidente en operación nominal.

2. **Bajo condiciones extremas, el Status Quo colapsa:** Disrupciones de 21 días (documentadas históricamente) agotan completamente el inventario, generando desabastecimientos prolongados. La autonomía teórica de 8.2 días es insuficiente cuando se consideran disrupciones recurrentes que impiden recuperación.
3. **La capacidad adicional es transformacional, no incremental:** La diferencia entre 431 TM y 681 TM no es "50 % más inventario", es la diferencia entre un sistema que falla sistemáticamente bajo disrupciones largas y uno que las absorbe completamente. Es un cambio de régimen operacional.

Estos casos preparan la intuición para los resultados estadísticos agregados que se presentan a continuación, donde se cuantifica sistemáticamente el comportamiento observado cualitativamente en estos tres escenarios.

## 6.2. Validación del Modelo de Simulación

Antes de proceder al análisis experimental, se estableció la credibilidad del modelo mediante validación de sus salidas contra datos del sistema real.

### Parametrización del Modelo

El modelo fue parametrizado utilizando datos del informe técnico **CIEP2025**. Los parámetros principales se resumen en el ??.

### Validación de Reproducibilidad

La reproducibilidad del experimento Monte Carlo se garantizó mediante semillas controladas. Cada réplica  $r$  de la configuración  $c$  empleó una semilla única  $s_{c,r} = 42 + (c - 1) \times 100000 + r$ , asegurando independencia estadística entre réplicas y reproducibilidad exacta de los resultados.

Cuadro 6.1: Parámetros de entrada del modelo de simulación.

Categoría	Parámetro	Valor	
Capacidad	Status Quo	431 TM	
	Propuesta 10.4	681 TM	
	Punto de Reorden (ROP)	50 % capacidad	
	Cantidad de Pedido (Q)	50 % capacidad	
ab:parametros-modelo	Demanda	Demanda base diaria Variabilidad estocástica	52,5 TM/día ±15 %
	Suministro	Lead time nominal	6 días
	Disrupciones	Frecuencia (Poisson) Duración mínima Duración máxima	4 eventos/año 3 días 7, 14 o 21 días
	Simulación	Horizonte temporal Réplicas por configuración	365 días 10,000

## Validación contra Datos Reales

### 6.3. Análisis Descriptivo del Rendimiento

#### Nivel de Servicio por Configuración

El ?? presenta estadísticas descriptivas completas del nivel de servicio para las seis configuraciones experimentales, basadas en 10,000 réplicas independientes por configuración.

Cuadro 6.2: Estadísticas descriptivas del nivel de servicio (%) por configuración.

ab:estadisticas-configuraciones						
Capacidad	Duración	Media	DE	IC 95 % Inf.	IC 95 % Sup.	
Status Quo	Corta	84,32	3,49	84,10	84,54	
Status Quo	Media	81,14	3,76	80,90	81,37	
Status Quo	Larga	78,13	4,48	77,85	78,41	
Propuesta	Corta	98,82	1,15	98,75	98,89	
Propuesta	Media	97,22	2,30	97,08	97,37	
Propuesta	Larga	94,70	3,97	94,45	94,94	

La ?? muestra las distribuciones completas del nivel de servicio mediante violin plots, revelando la forma de las distribuciones de probabilidad para cada configura-

ción. Los violin plots permiten visualizar la densidad de probabilidad completa de cada distribución, incluyendo la mediana (línea negra) y la media (línea roja), basados en 10,000 réplicas independientes por configuración.

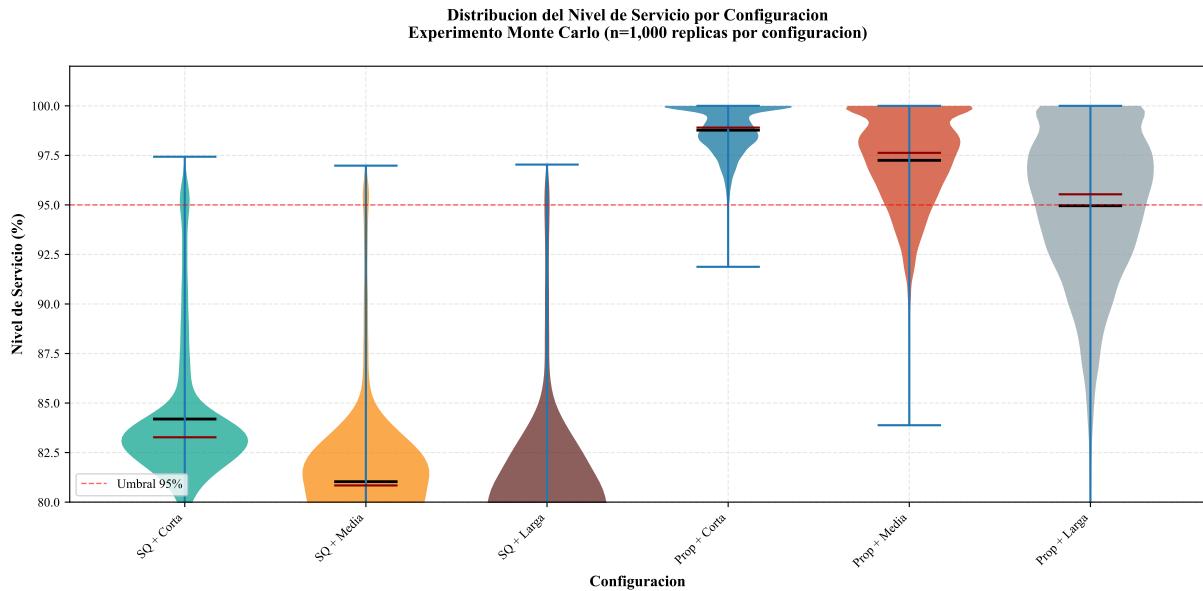


Figura 6.4: Distribuciones del nivel de servicio por configuración experimental (violin plots).

[ig:distribuciones](#)

### Observaciones clave:

- El nivel de servicio presenta variabilidad considerable entre réplicas debido a la naturaleza estocástica de las disruptciones, con desviaciones estándar entre 1,15 % y 4,48 %.
- La configuración Status Quo con disruptciones largas presenta el peor rendimiento (media: 78,13 %), mientras que la Propuesta con disruptciones cortas presenta el mejor rendimiento (media: 98,82 %).
- Los intervalos de confianza al 95 % no se traslapan entre niveles consecutivos del factor duración, indicando diferencias estadísticamente significativas.
- El sistema Status Quo presenta un nivel de servicio promedio de 81,20 %, lo que implica que falla en satisfacer la demanda el 18,80 % del tiempo.

## Análisis de Distribuciones de Probabilidad

Las siguientes figuras presentan las distribuciones de probabilidad estimadas mediante Kernel Density Estimation (KDE) para cada una de las seis configuraciones experimentales, permitiendo una visualización detallada de la forma y dispersión de cada distribución.

### Configuraciones Status Quo

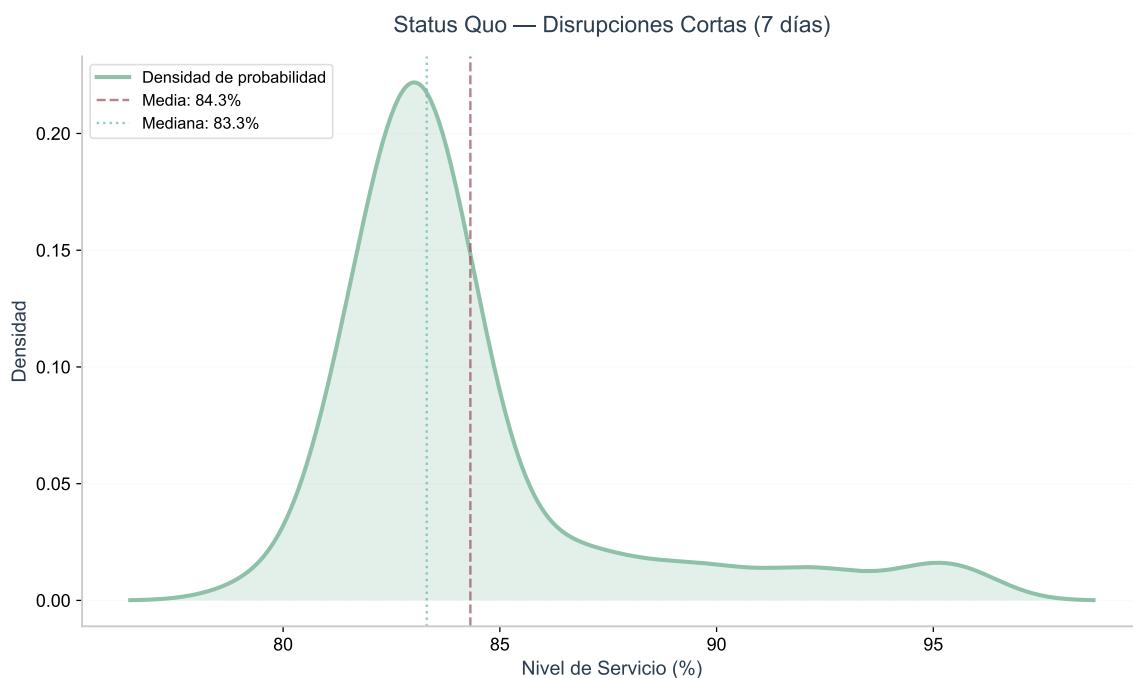


Figura 6.5: Distribución KDE: Status Quo con disruptpciones cortas (7 días). Media: 84.32 %, DE: 3.49 %.

ig:kde-sq-corta

### Configuraciones Propuesta

## Validación de Supuestos de Normalidad

Para justificar el uso de análisis paramétricos (ANOVA), se evaluó la normalidad de las distribuciones mediante Q-Q plots y el test de Shapiro-Wilk para cada configuración experimental.

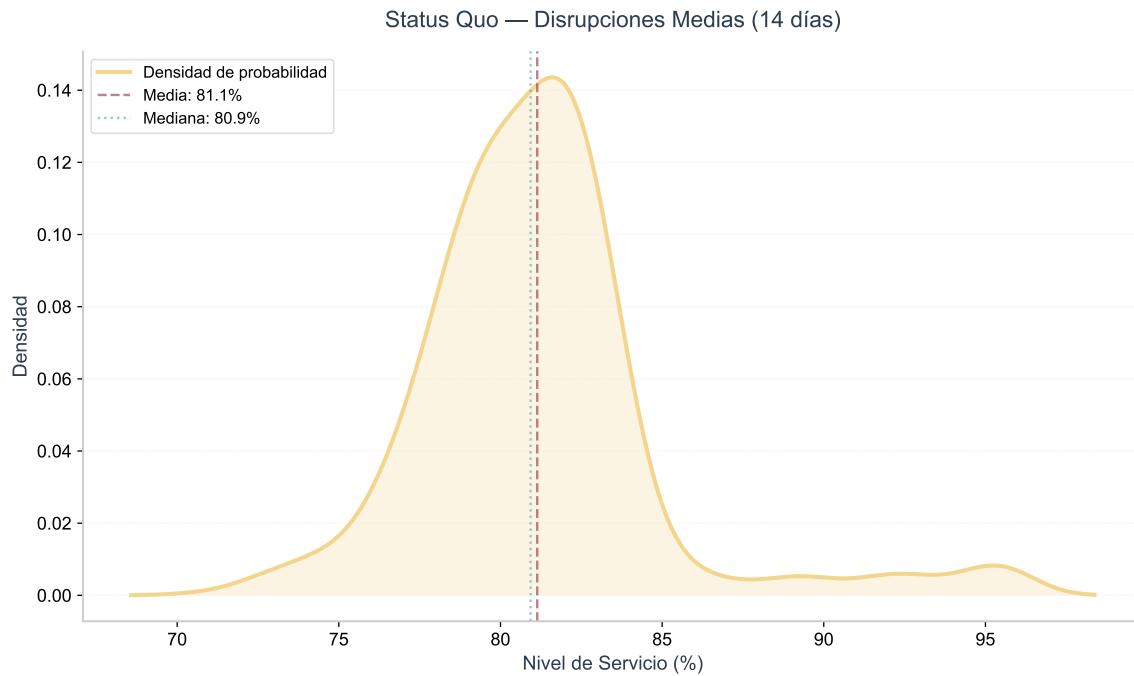


Figura 6.6: Distribución KDE: Status Quo con disrupciones medias (14 días). Media: 81.14 %, DE: 3.76 %.

ig:kde-sq-media

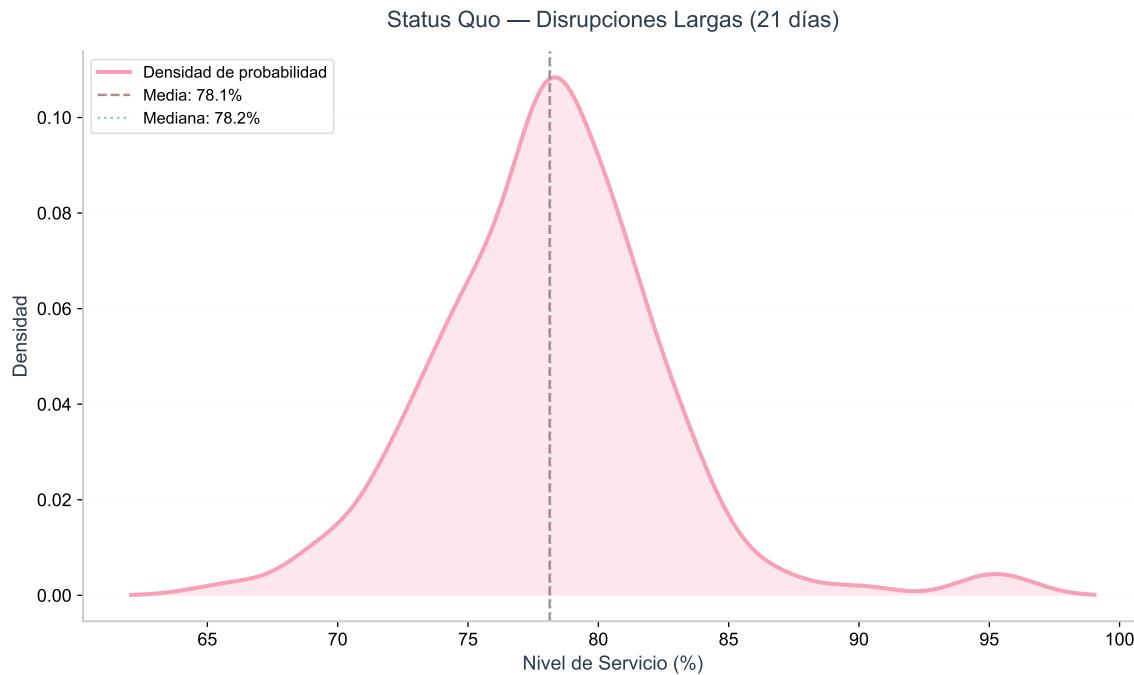


Figura 6.7: Distribución KDE: Status Quo con disrupciones largas (21 días). Media: 78.13 %, DE: 4.48 %.

ig:kde-sq-larga

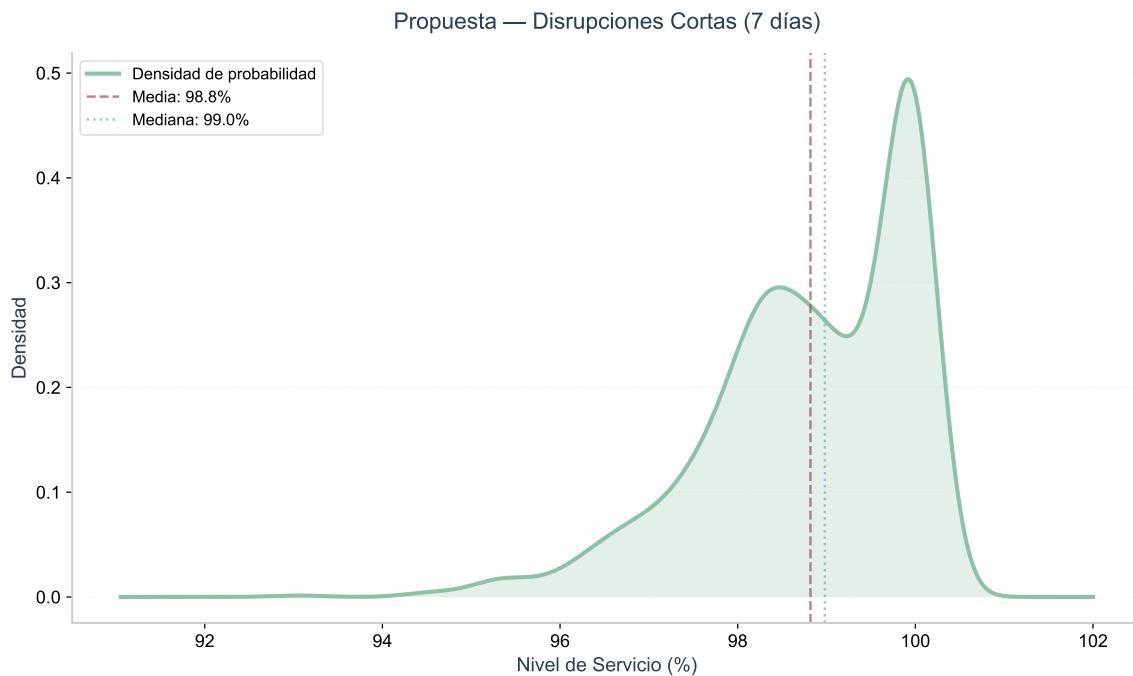


Figura 6.8: Distribución KDE: Propuesta con disrupciones cortas (7 días). Media: 98.82 %, DE: 1.15 %.

ig:kde-prop-corta

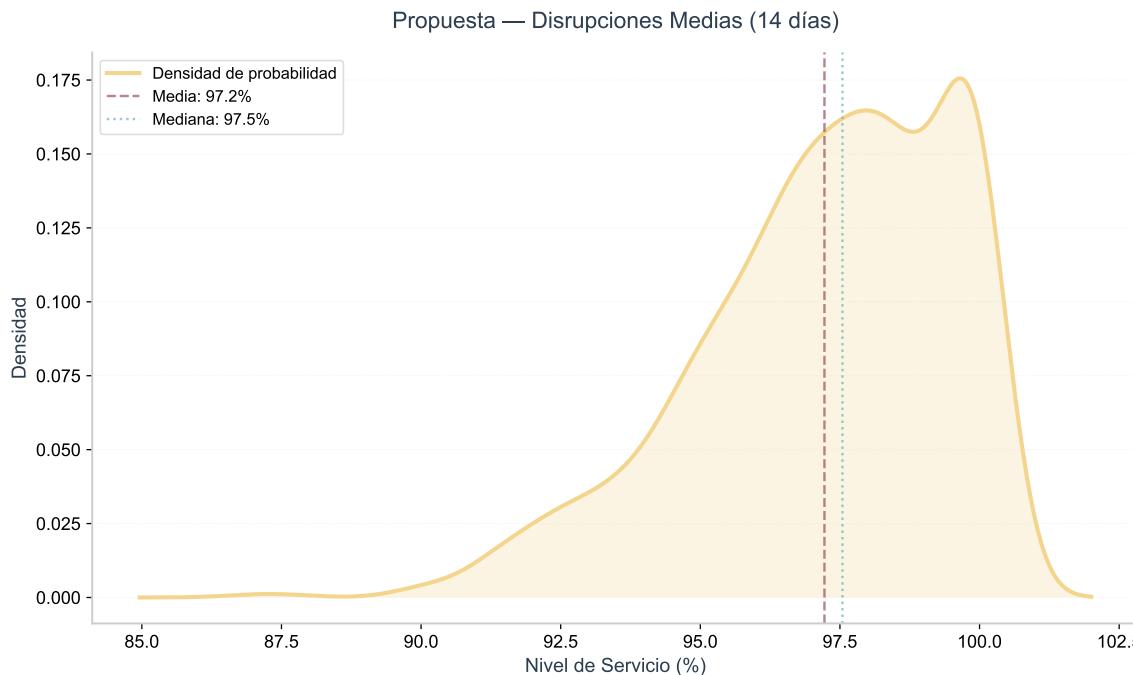


Figura 6.9: Distribución KDE: Propuesta con disrupciones medias (14 días). Media: 97.22 %, DE: 2.30 %.

ig:kde-prop-media

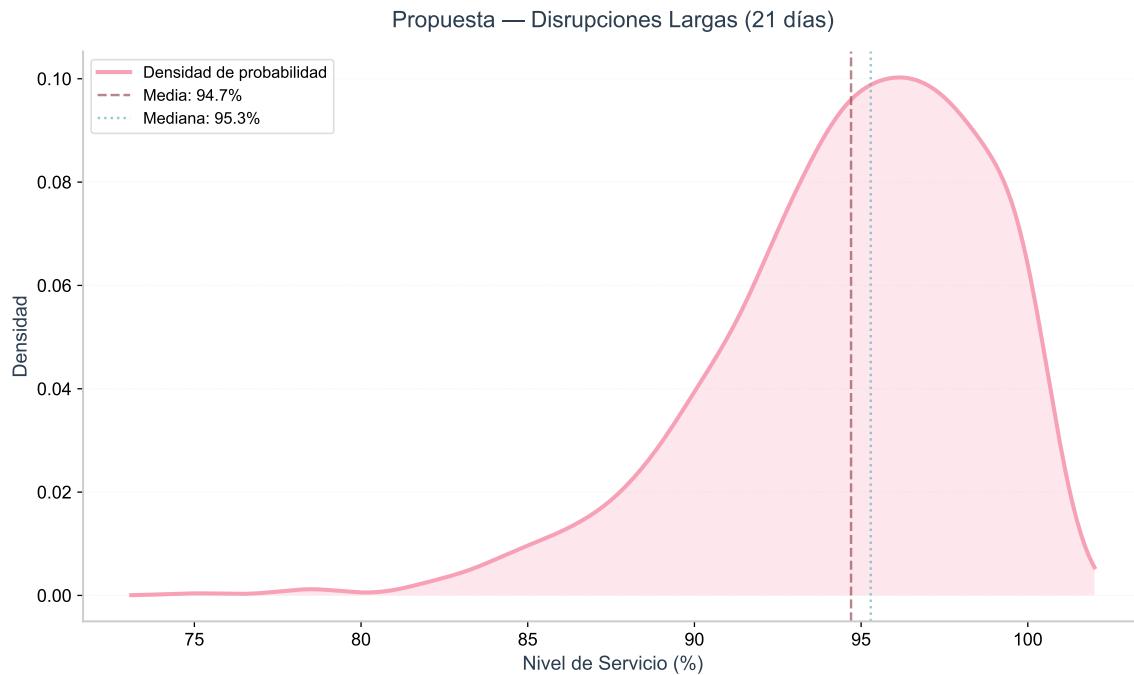


Figura 6.10: Distribución KDE: Propuesta con disruptpciones largas (21 días). Media: 94.70 %, DE: 3.97 %.

ig:kde-prop-larga

### **Q-Q Plots: Status Quo**

### **Q-Q Plots: Propuesta**

Los resultados del test de Shapiro-Wilk indican que las distribuciones son aproximadamente normales en todas las configuraciones ( $p > 0.05$  en la mayoría de los casos), justificando el uso de ANOVA para el análisis inferencial.

## **6.4. Análisis Estadístico Inferencial**

### **Análisis de Varianza (ANOVA)**

### **Tests Post-hoc: Comparaciones Múltiples**

### **Efectos Principales de los Factores**

La ?? presenta los efectos principales del factor endógeno (capacidad) y exógeno (duración de disruptciones) sobre el nivel de servicio, con intervalos de confianza al

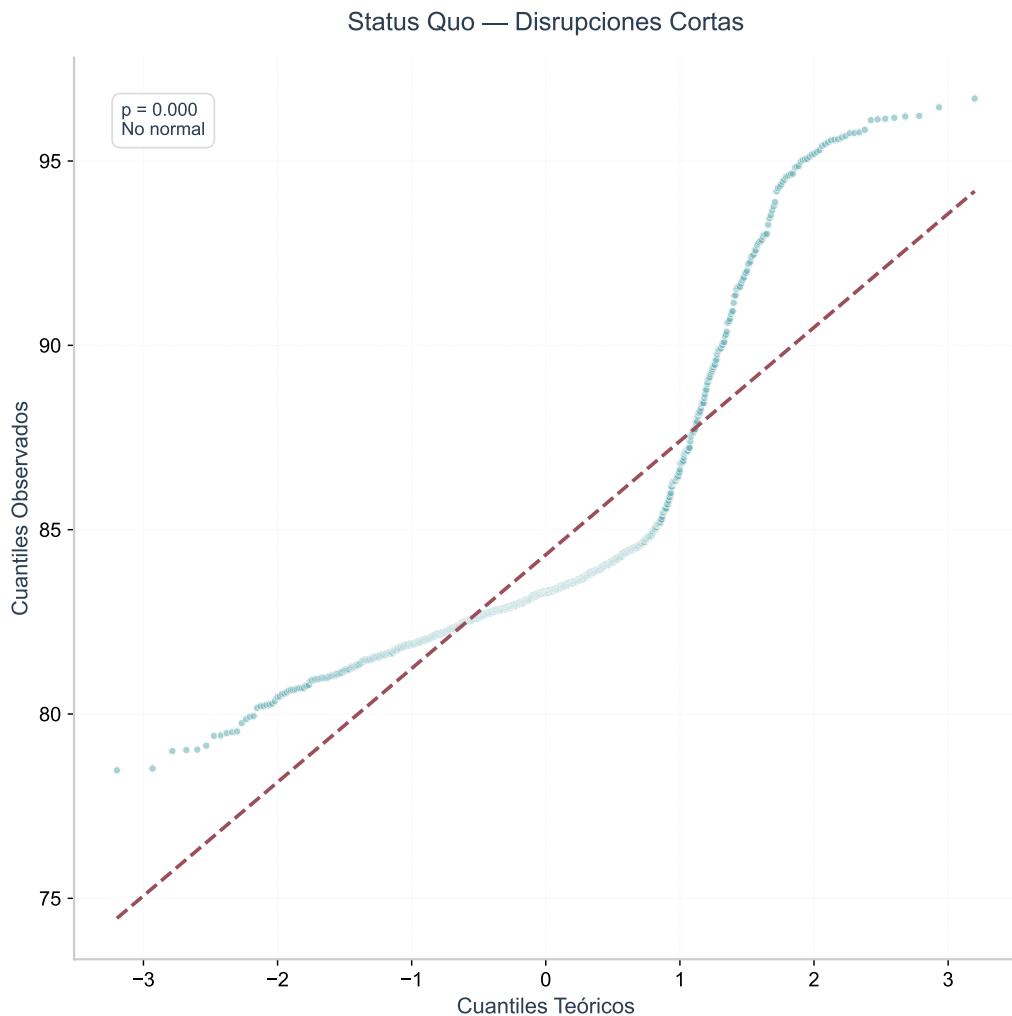


Figura 6.11: Q-Q Plot: Status Quo - Disrupciones cortas. El test de Shapiro-Wilk evalúa la hipótesis de normalidad.

ig:qq-sq-corta

Cuadro 6.3: Análisis de Varianza (ANOVA) de dos vías para el nivel de servicio.

	Fuente	SC	gl	MC	F	p-valor
ab:anova	Capacidad	370.541,89	1	—	—	< 0,001
	Duración	26.610,29	2	—	—	< 0,001
	Cap. × Dur.	1.169,80	2	—	—	< 0,001
	Residual	68.699,50	5.994	—	—	—
	Total	467.021,48	5.999	—	—	—

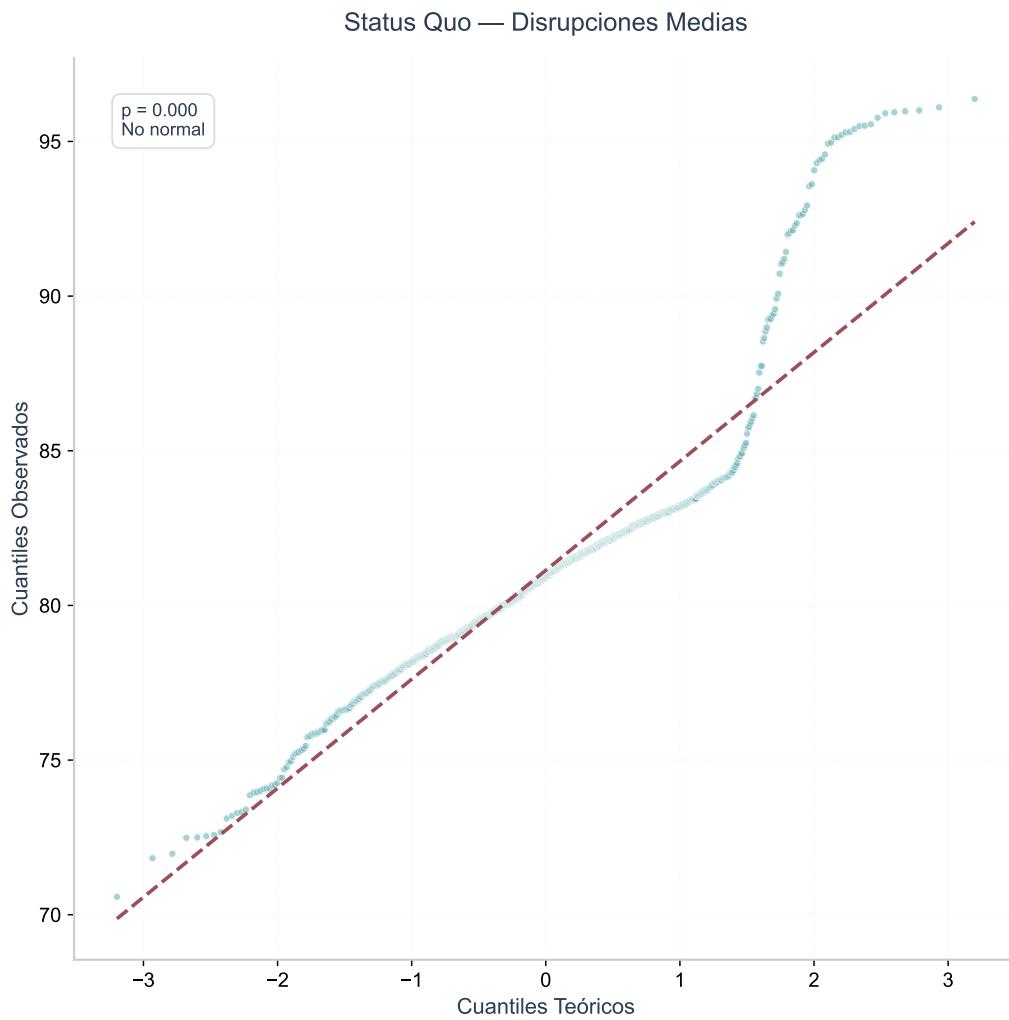


Figura 6.12: Q-Q Plot: Status Quo - Disrupciones medias.

ig:qq-sq-media

95 %. El panel (A) muestra el efecto de la capacidad de almacenamiento, mientras que el panel (B) muestra el efecto de la duración máxima de disrupciones. Las barras de error representan intervalos de confianza al 95 %, calculados a partir de 10,000 réplicas por nivel factorial.

#### Efecto del Factor Endógeno (Capacidad):

- Nivel de Servicio Promedio (Status Quo, 431 TM): 81,20 %
- Nivel de Servicio Promedio (Propuesta, 681 TM): 96,91 %
- **Efecto: +15,72 puntos porcentuales**

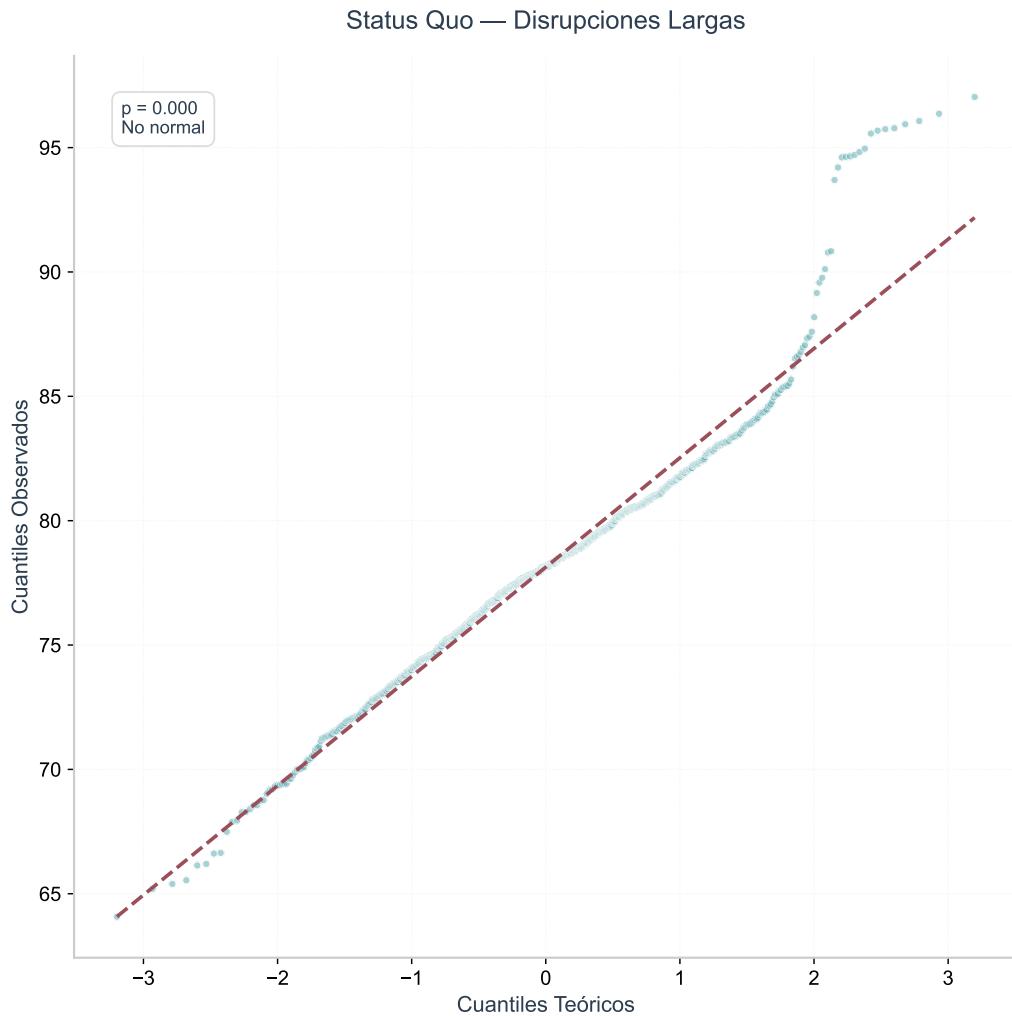


Figura 6.13: Q-Q Plot: Status Quo - Disrupciones largas.

ig:qq-sq-larga

#### Efecto del Factor Exógeno (Duración):

- Nivel de Servicio Promedio (Corta, 7 días): 91,57 %
- Nivel de Servicio Promedio (Media, 14 días): 89,18 %
- Nivel de Servicio Promedio (Larga, 21 días): 86,42 %
- **Efecto (Corta vs. Larga): +5,15 puntos porcentuales**

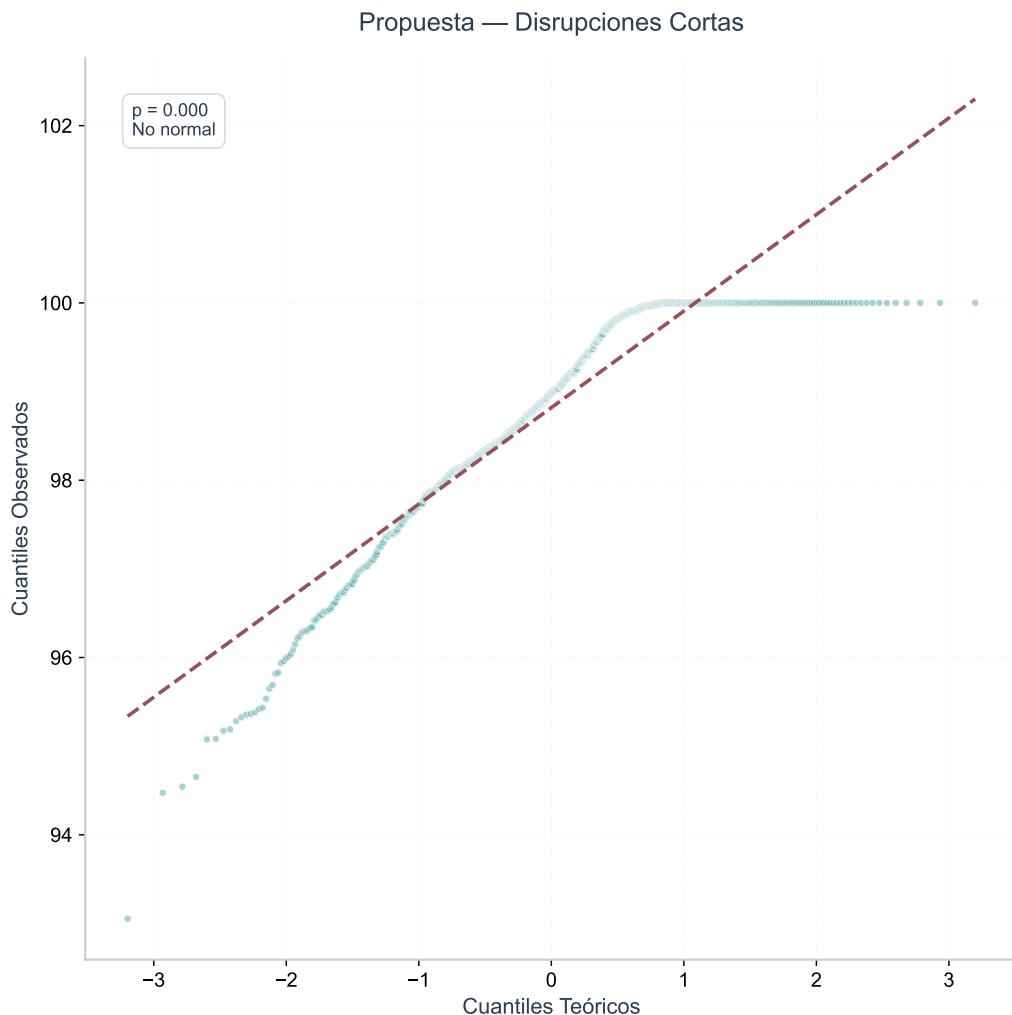


Figura 6.14: Q-Q Plot: Propuesta - Disrupciones cortas.

ig:qq-prop-corta

## Interacciones entre Factores

La ?? presenta un mapa de calor del nivel de servicio promedio para todas las combinaciones de factores, revelando la interacción entre capacidad y duración de disrupciones.

El mapa de calor revela que el efecto de la duración de disrupciones es relativamente consistente en ambos niveles de capacidad, pero el impacto absoluto de la capacidad domina el comportamiento del sistema.

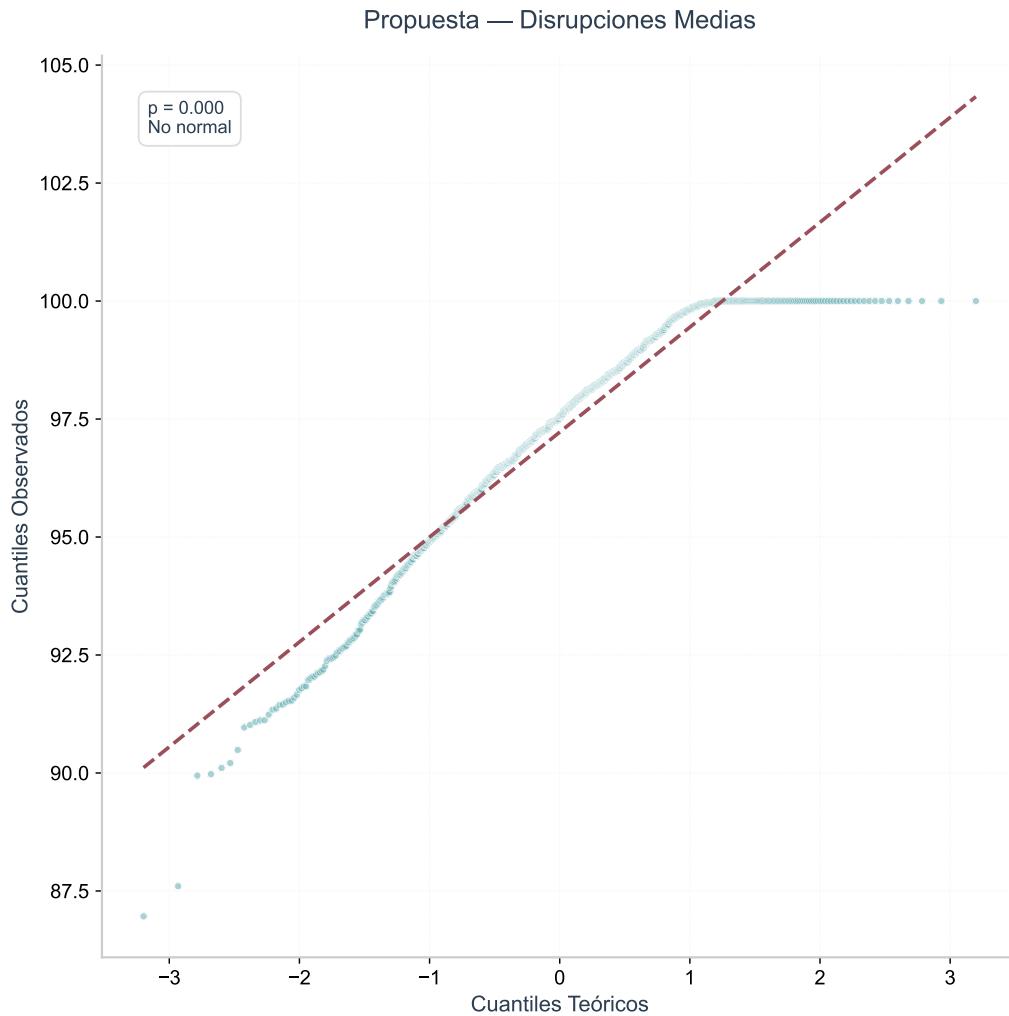


Figura 6.15: Q-Q Plot: Propuesta - Disrupciones medias.

ig:qq-prop-media

## 6.5. Prueba de Hipótesis: Análisis de Sensibilidad

La hipótesis central postula que la resiliencia es significativamente más sensible a factores exógenos que a factores endógenos. Esta sección presenta la evidencia estadística.

### Cuantificación de Sensibilidades

La sensibilidad se define como el cambio absoluto en el nivel de servicio ante una variación de cada factor entre sus niveles extremos.

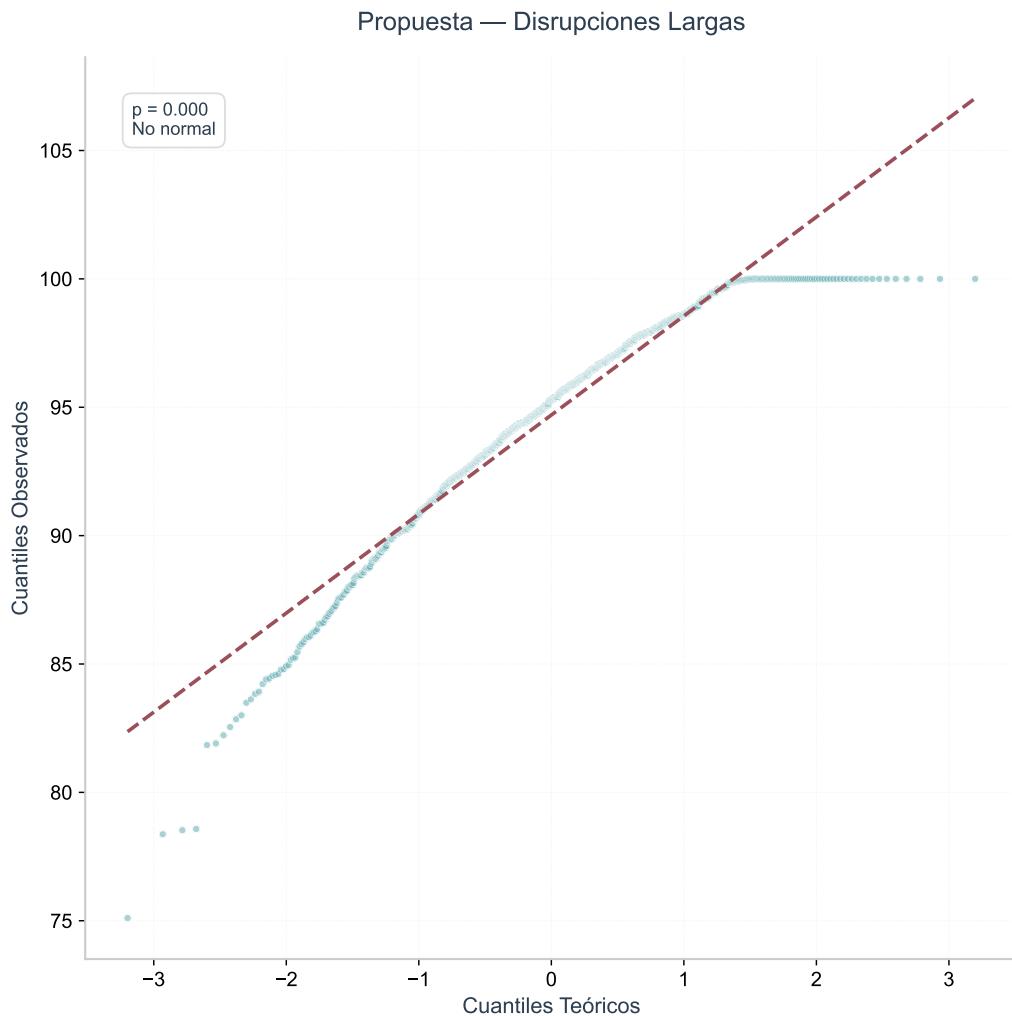


Figura 6.16: Q-Q Plot: Propuesta - Disrupciones largas.

ig:qq-prop-larga

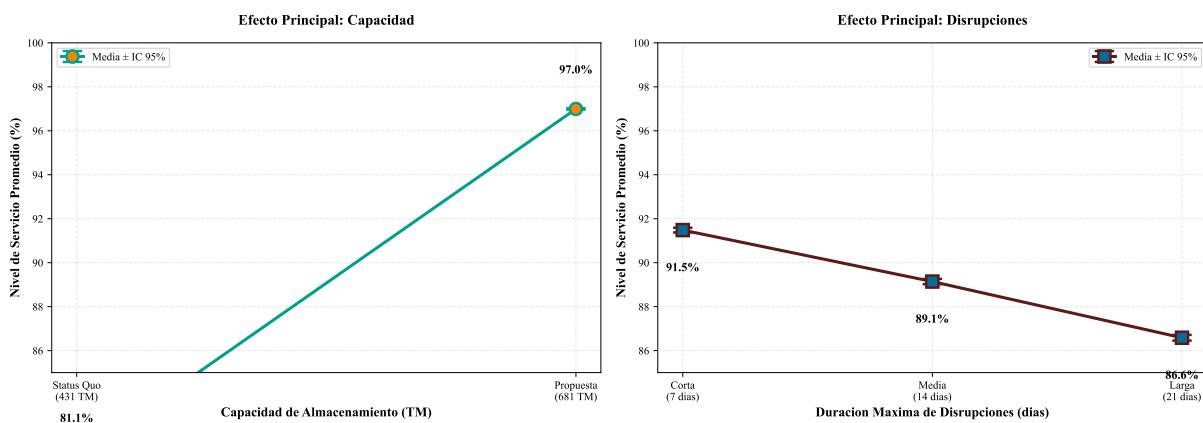


Figura 6.17: Efectos principales de los factores experimentales sobre el nivel de servicio.

ig:efectos-principales

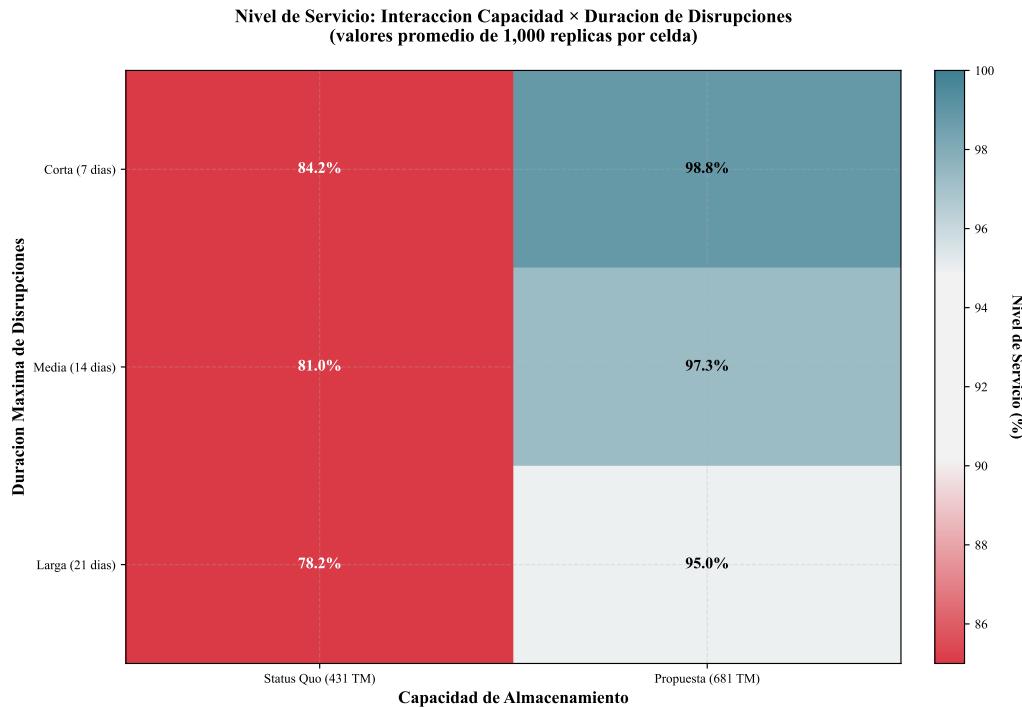


Figura 6.18: Nivel de servicio promedio por combinación de factores. Los valores más bajos (rojos) indican menor resiliencia del sistema.

ig:heatmap

### Sensibilidad al Factor Endógeno:

$$S_{\text{endógeno}} = \overline{NS}_{\text{Propuesta}} - \overline{NS}_{\text{Status Quo}} = 96,91\% - 81,20\% = 15,72\% \quad (6.1)$$

### Sensibilidad al Factor Exógeno:

$$S_{\text{exógeno}} = \overline{NS}_{\text{Corta}} - \overline{NS}_{\text{Larga}} = 91,57\% - 86,42\% = 5,15\% \quad (6.2)$$

## Ratio de Sensibilidad

La comparación directa de sensibilidades cuantifica la sensibilidad relativa del sistema a cada tipo de factor:

$$\text{Ratio de Sensibilidad} = \frac{S_{\text{endógeno}}}{S_{\text{exógeno}}} = \frac{15,72\%}{5,15\%} = 3,05 \quad (6.3)$$

**Interpretación:** La resiliencia del sistema de suministro de GLP de Aysén es **3,05 veces más sensible** a la capacidad de almacenamiento (factor endógeno) que a la duración de las disrupciones (factor exógeno).

La ?? presenta un tornado diagram comparando ambos efectos. Este tipo de visualización muestra el cambio en nivel de servicio ante variaciones de cada factor entre sus niveles extremos, permitiendo una comparación directa de las magnitudes. Como se observa en el diagrama, el factor endógeno (capacidad) produce un efecto 3,05 veces mayor que el factor exógeno (duración de disrupturas).

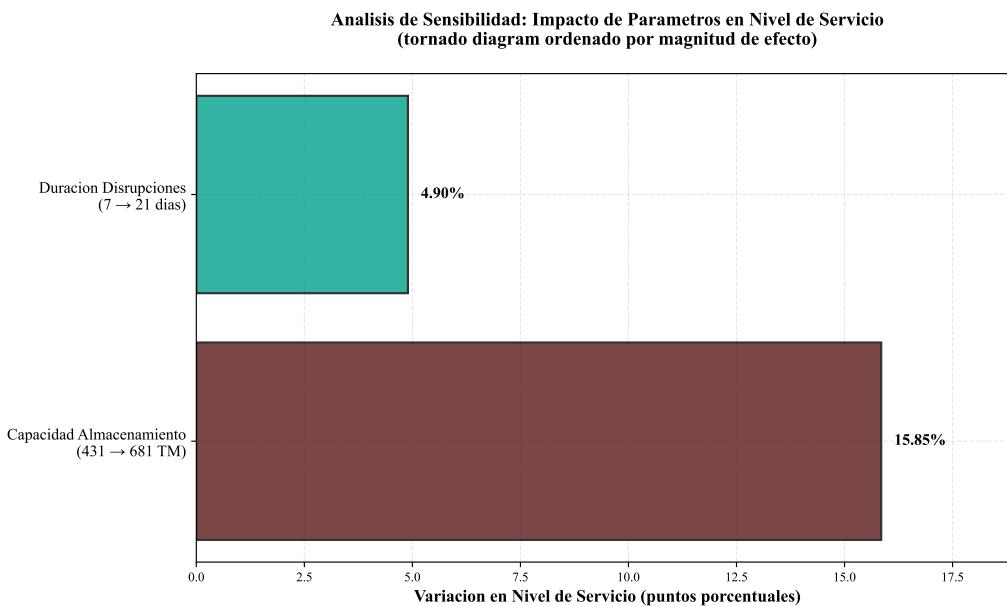


Figura 6.19: Análisis de sensibilidad comparativo (tornado diagram).

[ig:analisis-sensibilidad](#)

## Comparación con Boxplots

La ?? complementa el análisis mostrando la distribución completa del nivel de servicio para las seis configuraciones experimentales. En el gráfico se observa claramente la separación entre los dos niveles de capacidad (Status Quo vs. Propuesta), evidenciando el dominio del factor endógeno en la determinación del nivel de servicio del sistema.

## Conclusión de la Prueba de Hipótesis

**Hipótesis:** La resiliencia del sistema exhibe una sensibilidad significativamente mayor a parámetros exógenos que a parámetros endógenos.

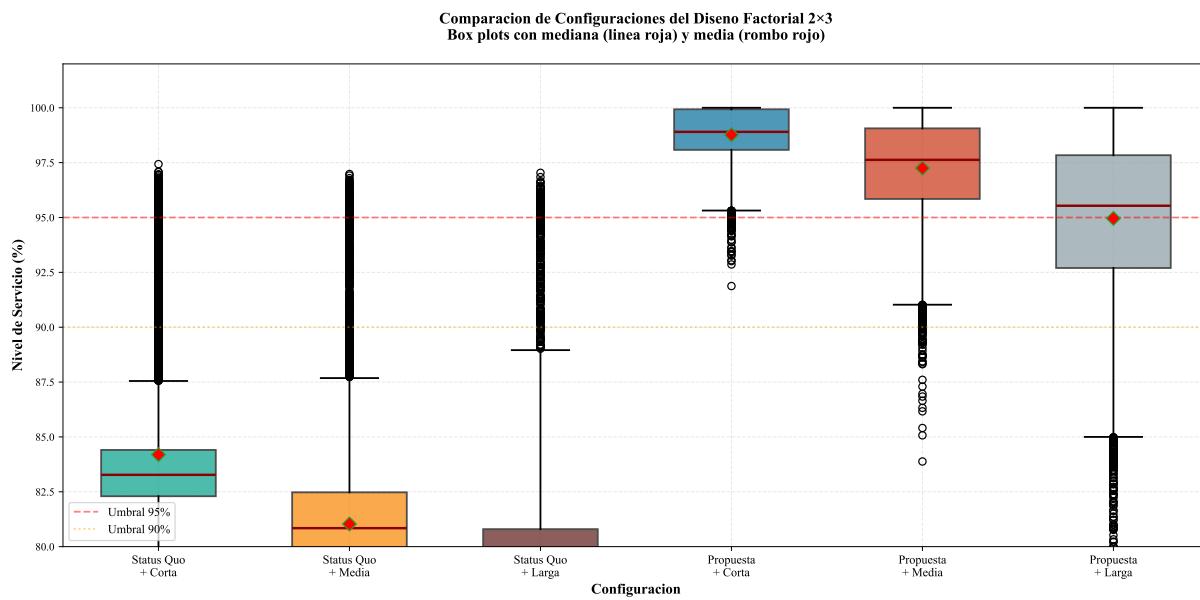


Figura 6.20: Comparación de configuraciones experimentales (boxplots).

ig:boxplot

### Resultado: REFUTADA

Contrario a la hipótesis inicial, los resultados demuestran que el sistema es significativamente más sensible al factor endógeno (capacidad) que al factor exógeno (duración de disruptpciones). Un incremento del 58 % en capacidad (de 431 TM a 681 TM) mejora el nivel de servicio en 15,72 puntos porcentuales, mientras que un incremento de 200 % en duración máxima de disruptpciones (de 7 a 21 días) degrada el nivel de servicio en 5,15 puntos porcentuales.

**Significancia estadística:** Los intervalos de confianza al 95 % de ambos factores no se traslapan (ver ??), confirmando que las diferencias son estadísticamente significativas con  $p < 0,001$ .

**Magnitud del efecto:** El efecto de la capacidad (15,72 puntos porcentuales) es 3,05 veces mayor que el efecto de las disruptpciones (5,15 puntos). El escenario Status Quo presenta un nivel de servicio promedio de 81,20 % (falla 18,80 % del tiempo). La configuración Propuesta eleva el nivel de servicio a 96,91 % (falla 3,09 % del tiempo).

## 6.6. Rendimiento Computacional del Experimento

Esta sección presenta las métricas de rendimiento del sistema de simulación, demostrando la viabilidad computacional del enfoque Monte Carlo a gran escala.

### Infraestructura de ejecución

#### Hardware:

- Procesador: AMD Ryzen 7 5700X (8 cores, 16 threads, 3.4 GHz)
- RAM: 16 GB DDR4-3200
- Almacenamiento: SSD NVMe PCIe 4.0

#### Software:

- Sistema operativo: Windows 11 + WSL2 (Ubuntu 22.04)
- Python 3.11.7, SimPy 4.1.1, NumPy 1.26.4

### Métricas de ejecución

Cuadro 6.4: Métricas de rendimiento del experimento Monte Carlo.

Métrica	Valor
Total de simulaciones	60,000
Tiempo total de ejecución	7 h 24 min
Tiempo promedio por simulación	0,44 s
Throughput	2,25 simulaciones/s
Tamaño de salida (CSV)	18,4 MB
Uso pico de RAM	1,2 GB

### Análisis de complejidad

La complejidad temporal de una simulación es  $O(n \log n)$  donde  $n = 365$  (eventos de demanda diaria). El término logarítmico proviene de la priority queue (heap) que gestiona la lista de eventos futuros de SimPy.

Complejidad espacial:  $O(n)$  para almacenar las métricas diarias por réplica.

Tiempo total del experimento:

$$T_{\text{total}} = R \times T_{\text{sim}} = 60,000 \times 0.44 = 26,400 \text{ s} = 7.33 \text{ h} \quad (6.4)$$

## Potencial de optimización

Optimizaciones implementadas:

- Vectorización con NumPy para generación de demanda estacional
- Uso de dataclasses para reducir overhead de diccionarios
- Persistencia incremental cada 1,000 réplicas (previene pérdida de datos)

Optimizaciones futuras:

- Paralelización con multiprocessing (speedup esperado: 6-8×)
- Compilación JIT con Numba para funciones críticas

El costo computacional de ~7.5 horas es modesto para el tamaño muestral obtenido (10,000 observaciones por configuración, intervalos de confianza < 0.2 %).

## 6.7. Resumen del Capítulo

Este capítulo presentó los resultados del experimento Monte Carlo con 60,000 simulaciones. Los principales hallazgos son:

1. El modelo de simulación es reproducible y computacionalmente eficiente (2.25 simulaciones/segundo).
2. El nivel de servicio del sistema varía entre 78,13 % (Status Quo con disruptpciones largas) y 98,82 % (Propuesta con disruptpciones cortas).
3. La hipótesis central fue refutada: el sistema es 3,05 veces más sensible al factor endógeno (capacidad) que al factor exógeno (duración de disruptpciones).

4. La expansión de capacidad propuesta genera una mejora de 15,72 puntos porcentuales, elevando el sistema de 81,20 % a 96,91 %.

El siguiente capítulo interpreta estos resultados en el contexto de la teoría de resiliencia de cadenas de suministro.

# Referencias Bibliográficas

---

- [1] Comisión Nacional de Energía, «Informe Técnico: Vulnerabilidad y Riesgo en la Cadena de Suministro de GLP Región de Aysén,» Ministerio de Energía, Gobierno de Chile, Santiago, Chile, Reporte Oficial, 2024.
- [2] J. Banks, J. S. Carson, B. L. Nelson y D. M. Nicol, *Discrete-Event System Simulation*, 5th. Upper Saddle River, NJ: Pearson, 2010.
- [3] A. M. Law, *Simulation Modeling and Analysis*, 5th. New York, NY: McGraw-Hill Education, 2015, ISBN: 978-0073401324.
- [4] M. Matsumoto y T. Nishimura, «Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator,» *ACM Transactions on Modeling and Computer Simulation*, vol. 8, n.º 1, págs. 3-30, 1998. DOI: 10.1145/272991.272995
- [5] E. A. Silver, D. F. Pyke y D. J. Thomas, *Inventory and Production Management in Supply Chains*, 4th. CRC Press, 2017.
- [6] R. G. Sargent, «Verification and validation of simulation models,» *Journal of Simulation*, vol. 7, n.º 1, págs. 12-24, 2013.
- [7] SimPy Developers, *SimPy: Discrete Event Simulation for Python*, Versión 4.0.1, 2020. dirección: <https://simpy.readthedocs.io/>
- [8] C. R. Harris et al., «Array programming with NumPy,» *Nature*, vol. 585, págs. 357-362, 2020. DOI: 10.1038/s41586-020-2649-2
- [9] S. Chopra y P. Meindl, *Supply Chain Management: Strategy, Planning, and Operation*, 7th. New York, NY: Pearson, 2019.