

## PAPER

# Simulación de la Dinámica de una Mempool de Blockchain: Un Enfoque Basado en Eventos Discretos

Simón Tadeo Ocampo \*

Universidad Tecnológica Nacional, Av. Del Petróleo Argentino 417, Berisso, Provincia de Buenos Aires, Argentina

\*Corresponding author: simontadeoocampo@alu.frlp.utn.edu.ar

## Abstract

La tecnología blockchain ha introducido nuevos paradigmas en los sistemas distribuidos, siendo la gestión de transacciones un componente crítico para su rendimiento. Este trabajo se centra en la mempool, el área de espera para transacciones no confirmadas en blockchains como Bitcoin. A diferencia de sistemas de colas convencionales, la dinámica de la mempool es analíticamente intratable debido a su disciplina de cola por prioridad basada en comisiones, su mecanismo de servicio por lotes y sus tiempos de servicio estocásticos. Ante esta complejidad, se propone un modelo de simulación de eventos discretos como la herramienta más adecuada para su análisis. El objetivo es caracterizar el rendimiento del sistema bajo diferentes condiciones operativas, evaluando el impacto de factores como la tasa de llegada de transacciones sobre métricas clave como el tiempo de confirmación y el nivel de congestión.

**Keywords:** simulación de eventos discretos; blockchain; mempool; teoría de colas; bitcoin

## 1. Introducción

### 1.1 Contexto: La Era de los sistemas distribuidos y la tecnología Blockchain

La tecnología blockchain se ha consolidado como una evolución fundamental en el campo de los sistemas distribuidos, funcionando como un registro contable digital que es, por diseño, inmutable, descentralizado y compartido. Su valor trasciende la aplicación en criptomonedas, proveyendo un mecanismo para generar confianza y transparencia en redes de múltiples participantes sin requerir intermediarios centralizados. Esta capacidad ha impulsado su adopción en diversas industrias, desde la gestión de cadenas de suministro hasta las finanzas descentralizadas (DeFi). La creciente relevancia de esta tecnología justifica un análisis riguroso de sus componentes internos y de la dinámica operativa que garantiza su funcionamiento, siendo el estudio de la gestión de transacciones pioneras un aspecto crucial para comprender sus límites de rendimiento, resiliencia y escalabilidad.

### 1.2 Definición del sistema: La Mempool como "sala de espera" de la Blockchain

En el núcleo del procesamiento de transacciones de una blockchain como Bitcoin (Nakamoto, 2008) se encuentra la *mempool* (*memory pool*). Este componente funciona como un área de almacenamiento temporal y descentralizada donde residen las transacciones que han sido transmitidas a la red pero que todavía no han sido confirmadas en un bloque. De forma análoga, la mempool actúa como una "sala de espera" virtual, un búfer que desacopla el flujo de llegada de transacciones, a menudo impredecible, del proceso rítmico pero estocástico de confirmación por parte de los mineros. Es importante destacar que no existe una única mempool global; cada nodo completo (*full node*) de la red mantiene su propia versión, que se sincroniza con el resto por medio de la propagación de transacciones.

### 1.3 Justificación del problema: La complejidad analítica de la dinámica de la mempool

A primera vista, la mempool podría ser confundida con un sistema de colas convencional. Sin embargo, su dinámica operativa viola los supuestos fundamentales de los modelos de teoría de colas analíticos (como el modelo M/M/1), haciendo que dichos modelos sean inadecuados para un análisis preciso. La complejidad analítica del sistema surge de tres características clave:

- **Disciplina de cola por prioridad (No-FIFO):** A diferencia de una política "First In, First Out", la mempool opera como un mercado. Los mineros, como agentes económicos, seleccionan las transacciones que maximizan su beneficio, priorizando las que posean una mayor comisión por unidad de peso (sats/vB).
- **Servicio por lotes (Batch Service):** El "servidor" (el minero exitoso) no procesa transacciones de forma individual. En su lugar, agrupa un lote de las transacciones más rentables hasta llenar la capacidad máxima del bloque (4,000,000 de unidades de peso).
- **Tiempos de servicio estocásticos:** El tiempo requerido para "servir" un lote (minar un bloque) es una variable aleatoria que sigue una distribución exponencial con una media de 10 minutos, una propiedad emergente del mecanismo de consenso Proof-of-Work.

La combinación de estas características hace que la *simulación de eventos discretos* se presente como la herramienta de análisis más apropiada y potente para estudiar y comprender la dinámica de la mempool.

### 1.4 Objetivos del estudio

El objetivo general de este trabajo es desarrollar un modelo de simulación de eventos discretos de la mempool de una blockchain para caracterizar su rendimiento bajo diferentes condiciones operativas. Los objetivos específicos son:

- Modelar el proceso de llegada de transacciones, ajustando distribuciones de probabilidad a datos empíricos.
- Implementar la lógica de selección de un minero, incorporando la disciplina de cola por prioridad y la restricción de capacidad del bloque.
- Simular el proceso estocástico de descubrimiento de bloques.
- Evaluar el impacto de factores clave (tasa de llegada, distribución de comisiones) sobre métricas de rendimiento (tiempo de confirmación, tamaño de la mempool).
- Identificar las condiciones operativas que conducen a la congestión de la red.

## 2. Marco teórico

### 2.1 Fundamentos de simulación de sistemas discretos

Un sistema se define por sus *atributos* (propiedades) y *actividades* (procesos que generan cambios). En un *sistema discreto*, como el que se aborda en este estudio, las variables de estado cambian de valor de forma instantánea en puntos específicos del tiempo. La dinámica de la mempool es un ejemplo claro: el estado del sistema (el conjunto de transacciones en espera) cambia únicamente cuando ocurre un *evento*: la llegada de una nueva transacción o el minado de un nuevo bloque.

La evolución del modelo se gestiona mediante un *reloj de simulación*. Acorde a la naturaleza del sistema, se utiliza el mecanismo de *intervalos de tiempo variable*, también conocido como "avance al siguiente evento". En este método, el reloj no avanza en incrementos fijos, sino que salta directamente al instante del suceso más próximo en una lista de sucesos futuros. En nuestro modelo,

el reloj avanzará al instante de la próxima *Llegada<sub>Transaccion</sub>* o del próximo *Descubrimiento<sub>Bloque</sub>*, lo que ocurra primero. Las *entidades* son los objetos que fluyen a través del sistema. En este caso, las entidades primarias son las *transacciones*, cada una con atributos clave como su *fee<sub>rate</sub>* y su *vsize<sub>bytes</sub>*. La construcción del modelo se realizará en un *Sistema de Modelado Visual Interactivo* (VIMS), en este caso Simulink.

## 2.2 Teoría de colas como contrapunto

La teoría de colas ofrece un marco matemático para analizar sistemas de espera, utilizando métricas fundamentales como la *tasa de llegada* ( $\lambda$ ) y la *tasa de servicio* ( $\mu$ ). El modelo M/M/1 es un pilar de esta teoría, describiendo un sistema con llegadas de Poisson, tiempos de servicio exponenciales y un único servidor con disciplina FIFO.

Si bien es útil como punto de partida conceptual, el modelo M/M/1 es manifiestamente inadecuado para representar la mempool. La siguiente tabla resume las discrepancias entre los supuestos del modelo y la realidad del sistema a estudiar (Tabla 1).

**Table 1.** Comparación de supuestos del modelo M/M/1 vs. realidad de la Mempool.

Supuesto M/M/1	Descripción	Validez en el funcionamiento de la Mempool
Llegadas (M)	Llegadas Poisson	Parcialmente válido con matices. Pueden ser "rafagosas".
Servicio (M)	Servicio individual	Inválido. El servicio es por lotes (bloques).
Servidores (1)	Un único servidor	Inválido. Miles de mineros compiten en paralelo.
Disciplina (FIFO)	Primero en llegar	Completamente inválido. Es una cola de prioridad por comisión.
Capacidad	Cola infinita	Inválido. Capacidad finita, se descartan transacciones.

## 2.3 Mecanismos fundamentales de la Blockchain de Bitcoin

Para construir un modelo fidedigno y robusto, es necesario comprender los mecanismos que gobiernan la dinámica de la mempool.

- **Consenso por Prueba de Trabajo (Proof-of-Work):** El "servicio" en nuestro sistema es una competencia probabilística. Los mineros resuelven un puzzle criptográfico, y el primero en encontrar una solución añade el bloque. Este proceso asegura que el tiempo entre bloques siga una distribución exponencial con media de 10 minutos.
- **El rol económico de las comisiones (Fees):** El espacio en un bloque es un recurso escaso. Las comisiones funcionan como un mecanismo de precios. Los usuarios ofrecen comisiones más altas para una confirmación más rápida, creando una disciplina de cola por prioridad.
- **Capacidad del lote (peso del bloque):** El protocolo impone un límite máximo al "peso" de un bloque (4,000,000 WU). Un minero seleccionará transacciones con la mayor fee rate hasta alcanzar este límite.

## 2.4 Diseño de Experimentos (DoE) para simulación

El *Diseño de Experimentos* (DoE) es un enfoque estructurado para planificar y analizar las corridas de simulación. Permite entender cómo múltiples variables de entrada, o *factores*, influyen en las métricas de salida, o *respuestas*. Los *niveles* de un factor son los valores específicos que este puede tomar. El Principio de Pareto justifica la decisión de centrarse en los factores que se presume tienen el mayor impacto. El análisis de los resultados se realizará mediante un *Análisis de Varianza* (ANOVA), utilizando la *Prueba F* para evaluar la significancia estadística de los factores.

### 3. Metodología y desarrollo del modelo

#### 3.1 Arquitectura del framework metodológico

El ajuste de distribuciones de probabilidad a datos empíricos constituye un problema fundamental en simulación de eventos discretos. Para la dinámica de la mempool, este proceso es particularmente crítico debido a la alta variabilidad inherente de las variables del sistema y la necesidad de evitar sobreajuste (*overfitting*) que comprometa la validez del modelo de simulación.

El framework implementado integra múltiples técnicas estadísticas en una metodología robusta que previene sobreajuste y garantiza generalización. Sus componentes principales son:

##### 3.1.1 Análisis de características previo

Antes del ajuste, se realiza un análisis estadístico para detectar características específicas de cada variable: alta asimetría ( $|\text{skewness}| > 2$ ), curtosis elevada ( $|\text{kurtosis}| > 7$ ), presencia de colas pesadas ( $P_{99}/P_{95} > 2.5$ ), y análisis robusto de outliers mediante criterio  $3 \times IQR$  (valores fuera de  $Q_1 - 3 \times IQR$  y  $Q_3 + 3 \times IQR$ ). Este análisis guía la preselección de familias de distribuciones teóricamente apropiadas y define el conjunto de candidatos a evaluar mediante criterios de información.

##### 3.1.2 Validación cruzada k-fold

Para evaluar la capacidad de generalización, se implementa validación cruzada 3-fold donde los datos se dividen en 3 subconjuntos. El modelo se ajusta en 2 subconjuntos y se evalúa en el restante usando el test Kolmogorov-Smirnov, repitiendo el proceso 3 veces. Esta técnica detecta memorización de ruido específico de la muestra (Law, 2014).

##### 3.1.3 Criterios de información múltiples

Se emplean tres criterios de información para selección de modelos:

- **BIC (Bayesian Information Criterion):**  $\text{BIC} = -2 \ln(L) + k \ln(n)$  como criterio principal (Schwarz, 1978)
- **AIC (Akaike Information Criterion):**  $\text{AIC} = -2 \ln(L) + 2k$  para comparación (Akaike, 1974)
- **HQIC (Hannan-Quinn Information Criterion):**  $\text{HQIC} = -2 \ln(L) + 2k \ln(\ln(n))$

donde  $L$  es la verosimilitud máxima,  $k$  el número de parámetros y  $n$  el tamaño muestral. El BIC penaliza más fuertemente la complejidad que el AIC, favoreciendo parsimonia estadística (Burnham & Anderson, 2002).

##### 3.1.4 Bootstrap para estabilidad de parámetros

Se implementa bootstrap con 1000 iteraciones para obtener estimaciones robustas y confiables de la estabilidad de los parámetros evaluados (Efron & Tibshirani, 1993). El umbral  $CV > 0.3$  se fundamenta en la literatura de simulación (Law, 2014), donde coeficientes de variación superiores al 30% en parámetros indican ajuste frágil que compromete la validez generativa del modelo. El número elevado de iteraciones bootstrap garantiza convergencia estadística y reduce la incertidumbre en la evaluación de estabilidad paramétrica.

##### 3.1.5 Detección automática de overfitting

El sistema detecta automáticamente sobreajuste mediante los siguientes umbrales:

- **BIC extremadamente bajo (< -8,000):** Valores BIC inferiores a  $-8,000$  en muestras  $n \approx 10^4$  sugieren log-verosimilitudes anómalamente altas ( $>4,000$ ) que exceden expectativas teóricas para datos continuos, indicando potencial sobreajuste (Burnham & Anderson, 2002).

- **K-S de validación cruzada muy bajo ( $< 0.02$ ):** Para muestras grandes ( $n > 10^3$ ), valores  $D < 0.02$  implican discrepancia inferior al 2%, lo que sugiere memorización de patrones específicos del conjunto de entrenamiento más que captura de la estructura poblacional (Massey Jr, 1951).
- **Inestabilidad paramétrica ( $CV > 0.3$ ):** Coeficientes de variación superiores al 30% indican que pequeñas perturbaciones en los datos producen cambios sustanciales en parámetros, violando el principio de robustez estadística (Law, 2014).
- **Log-verosimilitud positiva:** Indica problemas numéricos en la estimación por máxima verosimilitud.

Cuando se detecta overfitting severo, el sistema aplica fallback automático a la segunda mejor distribución según criterios ajustados, priorizando robustez sobre ajuste nominal.

### 3.1.6 Selección por adecuación teórica

Para evitar selecciones estadísticamente válidas pero conceptualmente inadecuadas, el framework incorpora consideraciones de adecuación teórica post-BIC sin modificar artificialmente los criterios de información. La selección final pondera:

- **Tiempos entre llegadas:** Se favorecen distribuciones con soporte semi-infinito positivo teóricamente apropiadas para procesos de llegada: Inverse Gamma (por flexibilidad en colas), gamma, Weibull, y exponencial (caso especial gamma).
- **Variables económicas/tamaños:** Se priorizan distribuciones capaces de modelar asimetría extrema y colas pesadas: Johnson SU (por versatilidad), log-normal (crecimiento multiplicativo), y Pareto (leyes de potencia).

Esta aproximación conserva la validez estadística del BIC mientras incorpora conocimiento del dominio, evitando la introducción de sesgos arbitrarios mediante bonificaciones numéricas ad hoc.

### 3.1.7 Criterios para segmentación de distribuciones

Para variables con alta heterogeneidad estructural, se implementa segmentación discreta evaluando puntos de corte candidatos. La metodología construye mezclas empíricas de la forma:

$$f_{mix}(x) = w_s f_s(x) + w_l f_l(x)$$

con BIC de mezcla definido como:

$$\text{BIC}_{mix} = -2\ell_{mix} + (k_s + k_l + 1)\ln(n)$$

La aceptación de segmentación requiere cumplimiento simultáneo de tres criterios fundamentados:

1. **Evidencia decisiva en BIC:**  $|\Delta BIC| > 10$  siguiendo las escalas de evidencia de Kass y Raftery (Kass & Raftery, 1995), donde diferencias superiores a 10 unidades constituyen evidencia decisiva.
2. **Estabilidad paramétrica en todos los segmentos:**  $CV < 0.3$  en cada segmento individual, garantizando robustez generativa (Law, 2014).
3. **Calidad de ajuste aceptable:** Estadístico K-S  $D < 0.3$  en todos los segmentos, asegurando representación fiel de la estructura empírica.

El incumplimiento de cualquier criterio resulta en rechazo automático de la segmentación, priorizando robustez sobre mejora nominal de verosimilitud. Esta aproximación jerárquica garantiza que los beneficios estadísticos se traduzcan en ventajas efectivas para simulación.

### 3.2 Criterios de evaluación estadística

La calidad del ajuste se evalúa mediante tests de bondad de ajuste estadísticamente establecidos, siguiendo la práctica estándar en simulación (Law, 2014).

#### 3.2.1 Test Kolmogorov-Smirnov

El estadístico K-S mide la máxima discrepancia entre la función de distribución empírica del conjunto de validación y la distribución teórica ajustada en entrenamiento (Massey Jr, 1951):

$$D_n = \sup_x |F_n(x) - F(x)|$$

En la implementación, el test K-S se emplea mediante validación cruzada (promedio de  $D$  en 3 particiones) para obtener una estimación robusta de la discrepancia. Aunque el test tradicional proporciona un p-valor, el framework se basa directamente en la magnitud de  $D$  por ser más informativa para comparación entre modelos en muestras grandes.

#### 3.2.2 Criterios de información para selección de modelos

Los criterios BIC y AIC, introducidos anteriormente, se emplean aquí únicamente de forma comparativa: BIC como criterio principal por su mayor penalización a la complejidad y AIC como referencia secundaria. Se reporta  $\Delta BIC$  respecto de la mejor alternativa para cuantificar evidencia relativa.

#### 3.2.3 Criterio de aceptabilidad para simulación

En muestras grandes (del orden de  $10^4$  observaciones) las pruebas de bondad de ajuste tienden a rechazar casi cualquier modelo paramétrico debido a su alta potencia asintótica. Por ello se adopta un criterio jerárquico pragmático basado en umbrales de calidad:

1. **Paso 1 (Clasificación por K-S):** El estadístico  $D$  promedio (validación cruzada) se clasifica como excelente ( $D < 0.05$ ), muy bueno ( $D < 0.10$ ), bueno ( $D < 0.20$ ), o aceptable ( $D < 0.30$ ).
2. **Paso 2 (Adecuación teórica):** Se considera la pertinencia conceptual de las familias de distribuciones candidatas según el tipo de variable y sus características empíricas, priorizando aquellas teóricamente apropiadas.
3. **Paso 3 (Estabilidad paramétrica):** Se confirma CV de parámetros (bootstrap de 1000 réplicas)  $\leq 0.3$ . En caso de inestabilidad superior, se activa la detección de overfitting.
4. **Paso 4 (Detección de overfitting):** Se aplica fallback automático si se detectan las señales críticas anteriormente explicadas:  $BIC < -8,000$ ,  $D < 0.02$ , o  $CV > 0.3$ .

La selección final prioriza el menor BIC ajustado entre candidatos estables, garantizando parsimonia y robustez para generación de muestras sintéticas.

### 3.3 Adquisición y análisis de datos empíricos

Habiendo establecido el framework metodológico a utilizar para el ajuste de distribuciones, se procede a su aplicación sistemática sobre datos empíricos de la mempool de Bitcoin. La fidelidad de un modelo de simulación depende críticamente de la calidad de sus datos de entrada, por lo cual el primer paso consistió en la captura de datos empíricos de la mempool de Bitcoin mediante un script en Python robusto con lógica de reconexión automática. Este script, utilizando la librería websockets, establece una conexión en tiempo real con el endpoint público de blockchain.info, suscribiéndose al flujo de transacciones no confirmadas. Este enfoque metodológico es crucial, ya que permite generar un timestamp de alta precisión en el momento exacto de la recepción de cada transacción, garantizando una medición fiel de los tiempos entre llegadas.

Se recolectaron datos en dos ventanas de una hora para capturar *baja* y *alta congestión*: en la madrugada de Argentina (05:40–06:40 ART, valle de actividad global) y al mediodía (11:01–12:01 ART,

pico de actividad que coincide con horarios comerciales de Europa y América). Esta estrategia de muestreo temporal permite caracterizar la dinámica de la mempool bajo condiciones operativas contrastantes: períodos de baja demanda donde las transacciones se procesan rápidamente con comisiones mínimas, versus períodos de alta demanda donde la competencia por espacio en bloques intensifica las comisiones y extiende los tiempos de confirmación. A continuación se analizan ambos escenarios.

### 3.3.1 Resultados del análisis de distribuciones (Alta Congestión)

Los siguientes resultados se obtuvieron del análisis del dataset de alta congestión, compuesto por 16,249 transacciones (tamaños y comisiones) y 13,751 intervalos entre llegadas, utilizando la metodología descrita en la subsección anterior.

**Tiempos entre llegadas:** Se registra asimetría marcada y curtosis elevada, con presencia de intervalos extremadamente cortos intercalados con lapsos más prolongados típicos de ráfagas de validación y propagación. La distribución *Inverse Gamma* fue seleccionada como mejor modelo ( $BIC = -30148.01$ ) con desempeño bueno en validación cruzada ( $D = 0.1357$ ) y parámetros altamente estables ( $CV$  bootstrap = 0.017), lo que indica robustez frente a re-muestreos. La media de intervalos es 0.262 segundos con desviación estándar de 0.568 segundos, mostrando alta variabilidad característica del tráfico de transacciones. La Figura 1 muestra el ajuste y diagnóstico Q-Q.

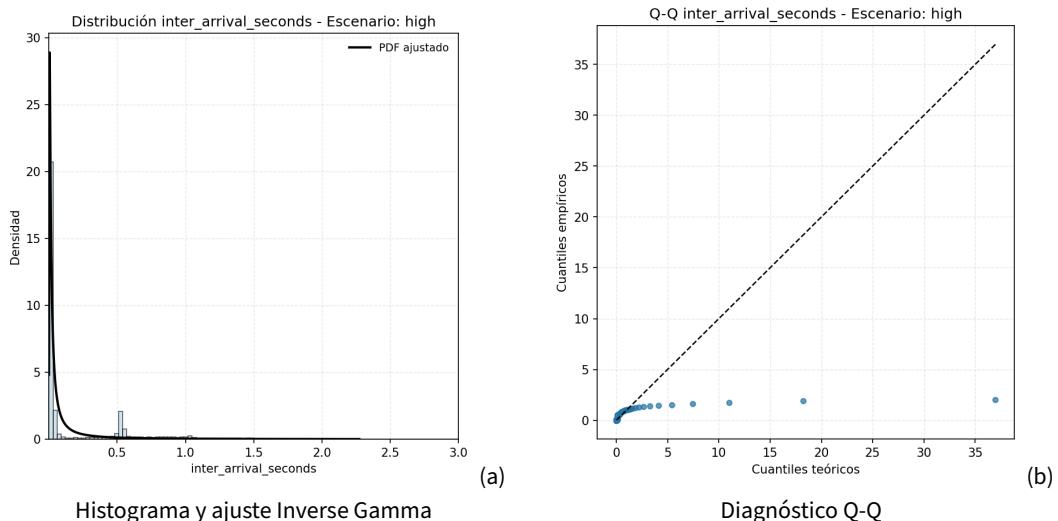
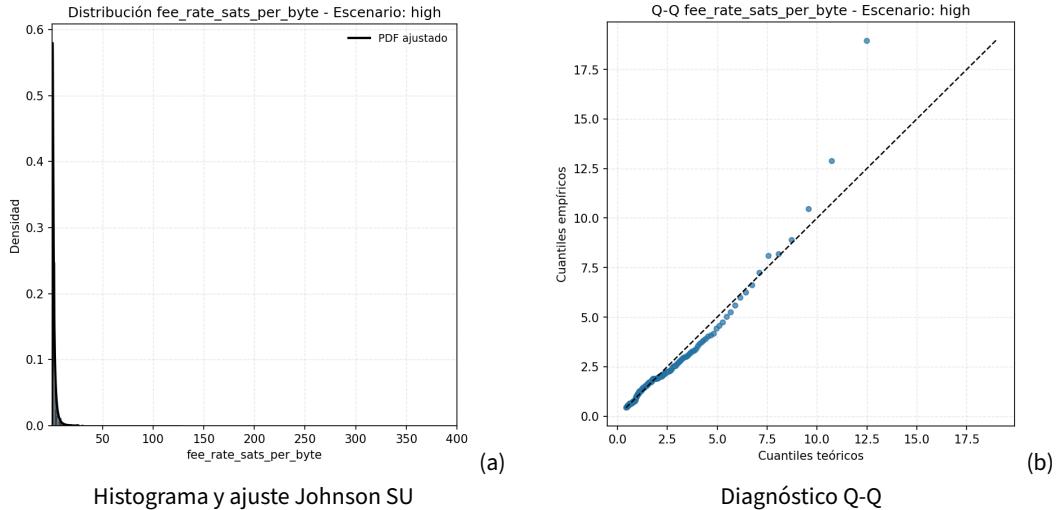


Figure 1. Intervalos entre llegadas (alta congestión): ajuste Inverse Gamma y diagnóstico Q-Q.

**Tasas de comisión:** Se observa asimetría extrema y curtosis muy elevada reflejando presión competitiva por inclusión en bloques. La distribución *Johnson SU* fue seleccionada ( $BIC = 50906.31$ ) con un ajuste muy bueno ( $D = 0.0734$ ) y parámetros estables ( $CV = 0.108$ ). La media de fee rate es 2.42 sats/vB con mediana 1.56 sats/vB y desviación estándar de 6.54 sats/vB, evidenciando la asimetría extrema con valores máximos de 444.31 sats/vB durante períodos de alta congestión. La flexibilidad superior de la transformación hiperbólica de Johnson SU (Johnson, 1949) permite capturar mejor las distorsiones simultáneas en ambas colas. La Figura 2 ilustra el ajuste y diagnóstico.

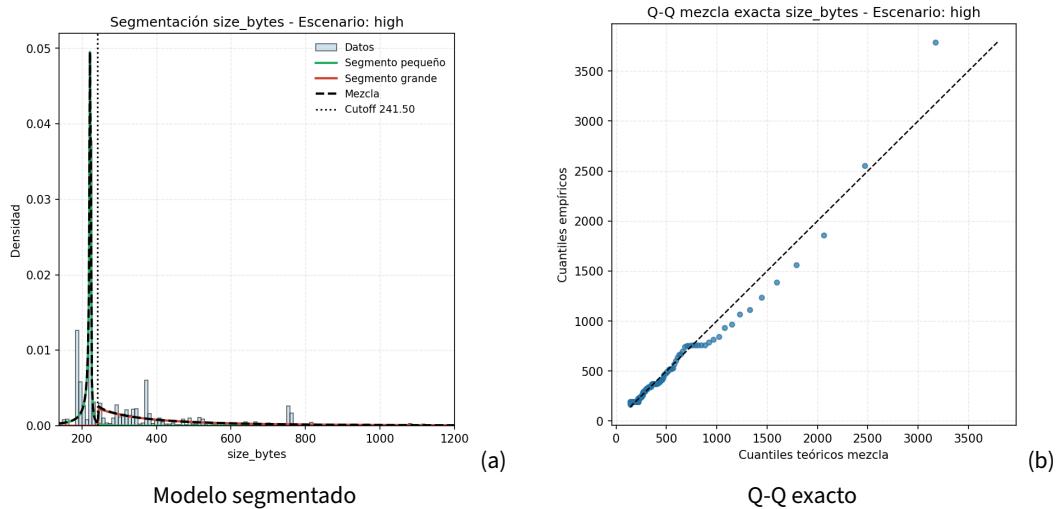


**Figure 2.** Fee rate (alta congestión): ajuste Johnson SU y diagnóstico Q-Q (superior flexibilidad para colas asimétricas).

**Tamaños de transacción:** Se mantiene un patrón de heterogeneidad estructural con mezcla implícita de subpoblaciones (transacciones simples vs. agrupadas) y colas muy pesadas. El mejor modelo único disponible *Johnson SU* ( $BIC = 197065.72$ ,  $KS D = 0.1608$ ) exhibe limitaciones críticas: inestabilidad paramétrica alta ( $CV = 0.649$ ) y ajuste deficiente en la cola pesada, evidenciando estructura latente no capturada adecuadamente por una distribución simple. La media de tamaños es 529.51 bytes con mediana 226.0 bytes y desviación estándar extrema de 2517.54 bytes, reflejando la presencia de transacciones de hasta 96,293 bytes. Esta deficiencia es particularmente problemática para simulación, donde la estabilidad paramétrica es esencial para generar muestras confiables.

**Segmentación exploratoria de tamaños.** Para superar las limitaciones del modelo único, se implementó segmentación discreta evaluando puntos de corte  $c$  candidatos. La metodología ajusta distribuciones independientes a  $x \leq c$  y  $x > c$ , construyendo la mezcla empírica  $f_{mix}(x) = w_s f_s(x) + w_l f_l(x)$  y calculando  $BIC_{mix} = -2\ell_{mix} + (k_s + k_l + 1)\ln(n)$ . El resultado óptimo ( $c = 241.5$  bytes) produce una mejora decisiva: la mezcla alcanza  $BIC = 190463.32$  ( $\Delta BIC = -6602.40$ ), con pesos  $w_s = 0.552$  y  $w_l = 0.448$ . El segmento pequeño ( $x \leq 241.5$ , *Johnson SU*) registra  $D = 0.1847$ ,  $BIC = 68187.65$  y  $CV = 0.076$ ; el segmento grande ( $x > 241.5$ , *Pareto*) mejora el ajuste en cola con  $D = 0.0885$ ,  $BIC = 100138.81$  y  $CV = 0.096$ . Ambos segmentos satisfacen simultáneamente los criterios de estabilidad ( $D < 0.3$ ,  $CV < 0.3$ ), garantizando un generador robusto para simulación.

La Tabla 2 documenta la transformación completa. La *mezcla segmentada* se adopta como generador final, superando decisivamente al modelo único en estabilidad, capacidad predictiva y representación de la heterogeneidad característica de alta congestión. Los cuantiles de cola (P99) muestran errores del modelo único superiores al 1800%, mientras que la mezcla mantiene errores inferiores al 50%. La Figura 3 muestra la comparación y diagnóstico Q-Q.



**Figure 3.** Tamaños de transacción (alta congestión): modelo único Johnson SU vs segmentación ( $c = 241.5$  bytes: modelo segmentado Johnson SU/Pareto) con diagnóstico Q-Q de la mezcla.

**Table 2.** Segmentación de tamaños ( $c = 241.5$  bytes) vs. modelo único (alta congestión).

Modelo	Segmento	Distribución	KS D	BIC	CV	Peso
Único	Global	Johnson SU	0.1608	197065.72	0.649	1.0000
Segmentado	$x \leq 241.5$	Johnson SU	0.1847	68187.65	0.076	0.552
Segmentado	$x > 241.5$	Pareto	0.0885	100138.81	0.096	0.448
Mezcla (BIC global)			–	190463.32	–	–
$\Delta BIC = -6602.40$ (mejora decisiva)						

**Nota:** El punto de corte  $c = 241.5$  reduce drásticamente el BIC y la discrepancia en la cola y elimina la inestabilidad global (CV global 0.649 → CV segmentados 0.076 y 0.096). La mejora ( $|\Delta BIC| = 6602.40 \gg 10$ ) constituye evidencia decisiva.

### 3.3.2 Parámetros para el modelo de simulación (Alta Congestión)

La Tabla 3 resume los parámetros y métricas clave de las distribuciones seleccionadas mediante la metodología robusta para el escenario de alta congestión.

**Table 3.** Parámetros y métricas de ajuste (alta congestión, modelo segmentado adoptado).

Variable	Distribución	Parámetros	D K-S (media ± sd)	CV Bootstrap
Intervalos (s)	Inverse Gamma	a: 0.565951; loc: -0.00133821; scale: 0.00880769	0.1357	0.017
Fee rate (sats/vB)	Johnson SU	a: -3.59146; b: 0.977982; loc: 0.335296; scale: 0.0572708	0.0734	0.108
Tamaño $\leq 241.5$ (bytes)	Johnson SU	a: 0.601; b: 0.3624; loc: 223; scale: 0.4938	0.1847	0.076
Tamaño $> 241.5$ (bytes)	Pareto	b: 1.633; loc: -73.46; scale: 316.5	0.0885	0.096

**Nota:** D es el estadístico K-S promedio 3-fold. Modelo segmentado adoptado por estabilidad superior y mejora decisiva en BIC ( $\Delta = -6602.40$ ).

### 3.3.3 Resultados del análisis de distribuciones (Baja Congestión)

Los siguientes resultados se obtuvieron del análisis del dataset de baja congestión, compuesto por 15,609 transacciones (tamaños y comisiones) y 12,340 intervalos entre llegadas, utilizando la metodología robusta descrita.

**Tiempos entre llegadas:** Persisten asimetría extrema y curtosis muy alta con presencia de outliers. La distribución *Inverse Gamma* ( $BIC = -23846.82$ ) mantiene clara superioridad frente a exponencial, gamma y Weibull al reducir la discrepancia simultáneamente en colas corta y larga; validación cruzada arroja  $D = 0.1452$  (bueno) con alta estabilidad ( $CV$  parámetros = 0.018). La media de intervalos es 0.292 segundos con mediana 0.021 segundos y desviación estándar de 0.687 segundos, indicando mayor espaciamiento temporal que en alta congestión. La Figura 4 muestra ajuste y Q-Q.

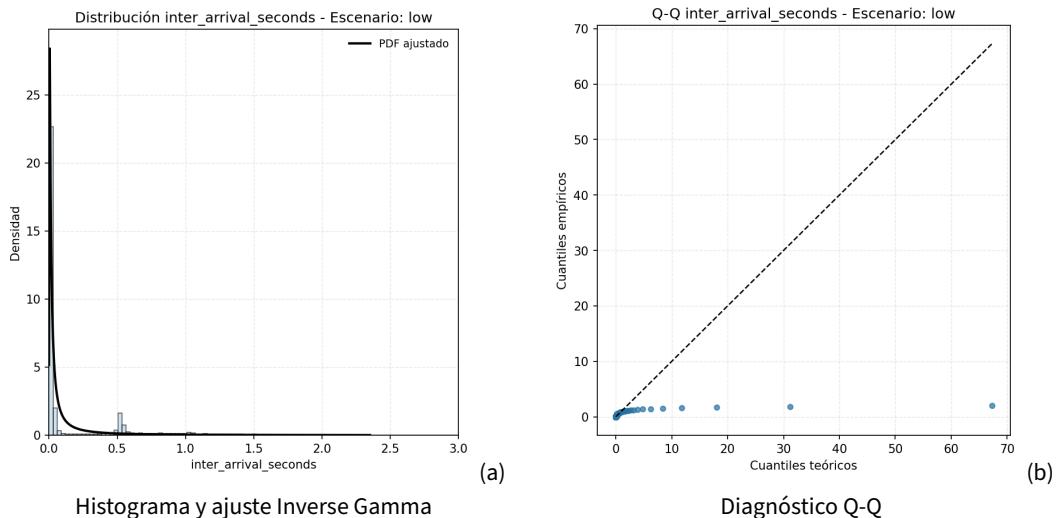
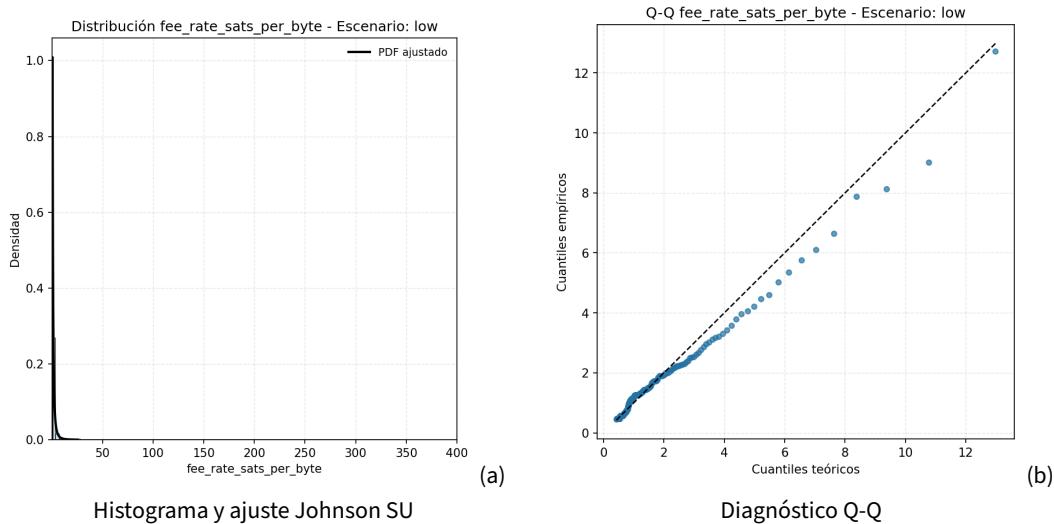


Figure 4. Intervalos entre llegadas (baja congestión): ajuste Inverse Gamma y diagnóstico Q-Q.

**Tasas de comisión:** Se observa estructura asimétrica con colas pesadas pero de menor intensidad que en alta congestión. La distribución *Johnson SU* fue seleccionada ( $BIC = 39774.12$ ) con ajuste bueno ( $D = 0.1094$ ) y parámetros estables ( $CV = 0.053$ ). La media de fee rate es 1.92 sats/vB con mediana 1.27 sats/vB y desviación estándar de 4.64 sats/vB, evidenciando menor presión competitiva con máximos de 344.14 sats/vB. La transformación hiperbólica de Johnson SU (Johnson, 1949) permite capturar las distorsiones en ambas colas eficientemente. La Figura 5 muestra el ajuste y diagnóstico.



**Figure 5.** Fee rate (baja congestión): ajuste Johnson SU y diagnóstico Q-Q (superior flexibilidad para estructura asimétrica).

**Tamaños de transacción:** Se mantiene el patrón de heterogeneidad estructural característico, aunque con menor intensidad que en alta congestión. Para reforzar la estabilidad del modelo se aplicó un truncamiento suave en el percentil 99.8 (22127 bytes), eliminando únicamente el 0.205% de las observaciones más extremas que inducían alta varianza. Con esta depuración, el mejor modelo único continúa siendo *Johnson SU* pero con métricas mejoradas ( $BIC = 188398.21$ ,  $KS D = 0.1786$ ) y coeficiente de variación bootstrap reducido a 0.138. La media de tamaños pasa a 380.81 bytes con mediana 245.0 bytes, desviación estándar 722.57 bytes, mínimo 139.0 bytes y máximo 21344.0 bytes. Esta poda controlada conserva la estructura operativa de la mempool (no afecta cuantiles críticos) y estabiliza la estimación paramétrica, lo que justifica su adopción para el generador de simulación. La Figura 6 muestra el diagnóstico del ajuste único *Johnson SU*.

**Segmentación exploratoria de tamaños.** Se aplicó la misma lógica de segmentación con búsqueda de corte óptimo que en el caso de alta congestión. El mejor punto de corte resultó  $c = 372.0$  bytes, con pesos  $w_s = 0.822$  (segmento pequeño) y  $w_l = 0.178$  (segmento grande). El segmento pequeño ( $x \leq 372$ ) se ajustó con Log-normal ( $D = 0.2087$ ,  $BIC = 140301.94$ ,  $CV = 0.016$ ), mientras que el segmento grande ( $x > 372$ ) requirió una chi-cuadrado ( $D = 0.2586$ ,  $BIC = 30167.07$ ,  $CV = 0.305$ ).

Aun cuando el BIC de la mezcla (183431.81) mejora frente al modelo único (188398.21) con  $\Delta BIC = -4966.40$  (evidencia fuerte), la evaluación debe considerar todos los criterios establecidos:

- **Mejora BIC:**  $|\Delta BIC| = 4966.40 > 10$  (evidencia fuerte para segmentación)
- **Estabilidad segmentos:** El segmento pequeño es estable ( $CV = 0.016 \leq 0.3$ ), pero el segmento grande presenta  $CV = 0.305 > 0.3$  (límite de inestabilidad)
- **Calidad de ajuste:** Ambos segmentos cumplen  $D < 0.3$  (aceptable)

La decisión metodológica final es rechazar la segmentación, aplicando consistentemente el criterio establecido: cualquier segmento con  $CV > 0.3$  o  $D > 0.3$  invalida la mezcla completa, independientemente de la mejora en BIC. Para simulación robusta, la estabilidad paramétrica es condición necesaria no negociable. Se adopta el modelo único *Johnson SU* por cumplir los criterios de estabilidad ( $CV = 0.138 < 0.3$ ,  $D = 0.1786 < 0.3$ ).

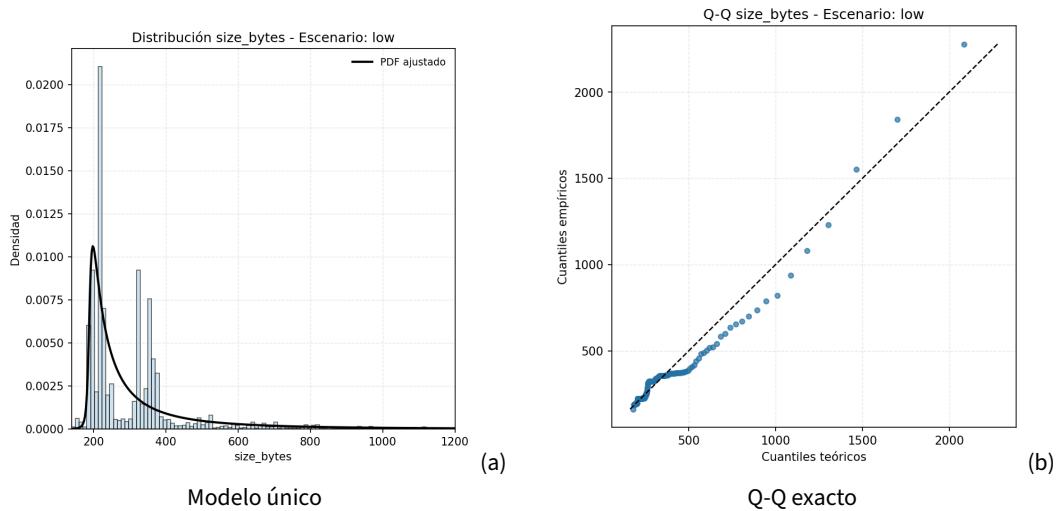


Figure 6. Tamaños de transacción (baja congestión): modelo único Johnson SU y diagnóstico Q-Q.

Table 4. Segmentación tamaños (baja congestión, corte 372.0 bytes) vs modelo único.

Modelo	Segmento	Distribución	KS D	BIC	CV	Peso
Único	Global	Johnson SU (truncado 99.8%)	0.1786	188398.21	0.138	1.0000
Segmentado	$x \leq 372$	Log-normal	0.2087	140301.94	0.016	0.822
Segmentado	$x > 372$	$\chi^2$	0.2586	30167.07	0.305	0.178
Mezcla (BIC global)			-	183431.81	-	-
$\Delta BIC = -4966.40$						

**Nota:** Aunque la reducción de BIC es sustancial ( $\Delta BIC = -4966.40$ ) y ambos segmentos cumplen KS con  $D < 0.3$ , el segmento grande presenta  $CV = 0.305 > 0.3$  (umbral de robustez). Por ello, se descarta la segmentación y se adopta el modelo único truncado.

### 3.3.4 Parámetros para el modelo de simulación (Baja Congestión)

La Tabla 5 resume los parámetros y métricas clave.

Table 5. Parámetros y métricas de ajuste (baja congestión, modelos únicos adoptados).

Variable	Distribución	Parámetros	D K-S (media $\pm$ sd)	CV Bootstrap
Intervalos (s)	Inverse Gamma	a: 0.521501; loc: -0.00150761; scale: 0.00781445	0.1452	0.018
Fee rate (sats/vB)	Johnson SU	a: -2.8088; b: 0.796842; loc: 0.399676; scale: 0.0399706	0.1094	0.053
Tamaño (bytes)	Johnson SU (truncado 99.8%)	a: -1.813859; b: 0.681927; loc: 187.508001; scale: 8.750103	0.1786	0.138

**Nota:** D es el estadístico K-S promedio 3-fold. p-valores bajos obedecen al gran  $n$ ; la decisión se basa en magnitud relativa de D, parsimonia (BIC) y estabilidad.

### 3.4 Arquitectura del modelo de simulación

La implementación computacional del modelo de simulación se realizó utilizando el lenguaje Python 3.10 junto con la librería SimPy (Simulation in Python) (Matloff, 2008). SimPy es un framework de simulación de eventos discretos basado en generadores que permite modelar sistemas complejos mediante procesos asíncronos. Esta elección metodológica ofrece múltiples ventajas sobre entornos visuales tradicionales: gestión eficiente de memoria para simulaciones de larga duración ( $>1$  millón de eventos), facilidad para ejecutar diseños experimentales mediante scripts automatizados, y capacidad de ejecución en entornos cloud (Google Colab) sin requerir licencias comerciales. La validación de las distribuciones estadísticas se realizó mediante las librerías NumPy y SciPy, garantizando la coherencia matemática con los parámetros ajustados empíricamente.

#### 3.4.1 Componentes principales del modelo

El modelo se estructura en tres subsistemas principales que encapsulan la lógica de generación, almacenamiento y procesamiento de transacciones.

**Subsistema de generación de transacciones:** La función `generar_tráfico()` implementa el proceso estocástico de llegada mediante generadores vectorizados que producen secuencias completas de interarribos, comisiones y tamaños utilizando las distribuciones parametrizadas (Tablas 5 y 3). Para el escenario de alta congestión, el tamaño implementa la mezcla segmentada mediante selección aleatoria binaria con probabilidad 0.552 que asigna cada transacción al componente Johnson SU (pequeñas) o Pareto (grandes).

Un proceso SimPy (`llegadas()`) itera sobre los vectores pre-generados programando la llegada de transacciones mediante `yield env.timeout(interrubos[i])`. Esta pre-generación vectorizada reduce drásticamente el overhead computacional comparado con generación on-demand por transacción. Cada transacción recibe prioridad  $p = -\text{fee\_rate}_i$ , donde el signo negativo invierte el orden de la cola (SimPy ordena ascendente por defecto) para que mayor comisión resulte en mayor prioridad.

**Subsistema de almacenamiento (Mempool):** El modelo utiliza un diccionario Python para representar la mempool, permitiendo acceso O(1) a las transacciones pendientes. Las transacciones se almacenan indexadas por identificador único, con atributos de tiempo de llegada, comisión y tamaño. El diseño de la cola implementa dinámicamente la disciplina de prioridad mediante ordenamiento explícito en cada evento de minado.

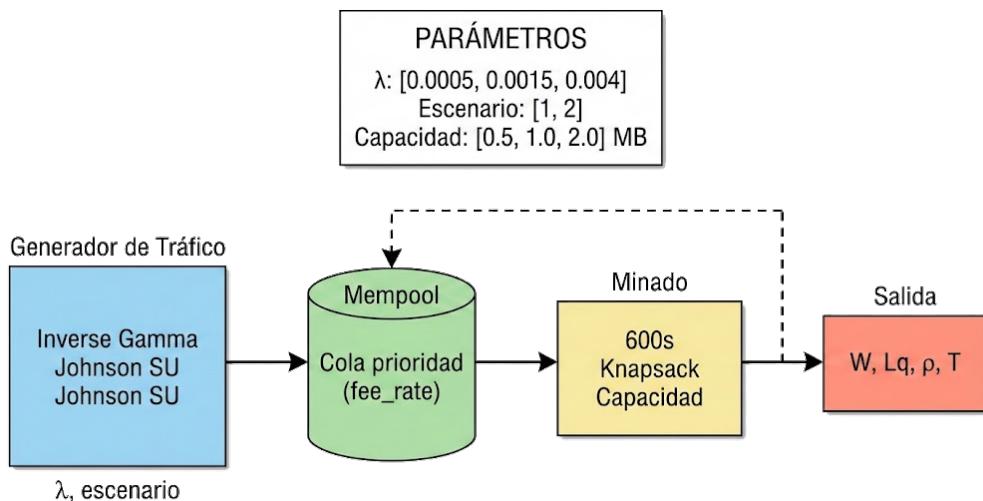
Un proceso de monitoreo (`monitor_cola()`) muestrea el tamaño de la mempool cada 60 segundos simulados, almacenando mediciones post-warmup para calcular la longitud de cola promedio  $L_q$  y su variabilidad temporal. Esta instrumentación pasiva no interfiere con la lógica de procesamiento principal.

**Subsistema de minado de bloques:** El modelo implementa procesamiento por lotes mediante eventos de minado que ocurren cada 600 segundos (intervalo fijo de bloque de Bitcoin). En cada evento, se ordenan las transacciones pendientes por fee-rate descendente y se aplica un algoritmo codiciso que selecciona transacciones hasta alcanzar la capacidad máxima del bloque. Este enfoque implementa la política de maximización de comisiones utilizada por mineros reales.

Al finalizar el procesamiento de cada transacción, se registra el tiempo de espera  $W_i = t_{exit,i} - t_{arrival,i}$  si el tiempo de salida supera el período de warmup. Las transacciones procesadas se eliminan de

la mempool utilizando operaciones O(1) sobre el diccionario, manteniendo eficiencia computacional incluso con colas de decenas de miles de transacciones.

La Figura 7 muestra esquemáticamente la arquitectura del modelo de simulación, ilustrando el flujo de transacciones desde su generación estocástica (subsistema de generación con distribuciones Inverse Gamma y Johnson SU), pasando por su almacenamiento temporal en la mempool (cola de prioridad ordenada por fee-rate), hasta su procesamiento por lotes cada 600 segundos (subsistema de minado con algoritmo knapsack sujeto a restricción de capacidad). Los parámetros experimentales (factor  $\lambda$ , escenario de congestión y capacidad del bloque) configuran el comportamiento del sistema, mientras que las métricas de desempeño ( $W, L_q, \rho, T$ ) se recolectan al finalizar cada bloque minado.



**Figure 7.** Arquitectura del modelo de simulación de eventos discretos. El generador de tráfico produce transacciones mediante distribuciones estadísticas calibradas, que se almacenan en la mempool (cola de prioridad por fee-rate). Cada 600 segundos, el proceso de minado selecciona transacciones hasta llenar la capacidad del bloque mediante algoritmo knapsack, generando las métricas de rendimiento del sistema.

### 3.4.2 Parámetros configurables y factores experimentales

El modelo se diseñó con flexibilidad paramétrica para facilitar la experimentación sistemática. El factor de carga ( $f_\lambda$ ) actúa como escalador aplicado a los interarribos base para modular la intensidad del tráfico, implementado como `data['inter'] = raw_inter * f_lambda`, donde valores menores a 1.0 reducen intervalos (aumentan  $\lambda$ ) y valores mayores a 1.0 incrementan intervalos (reducen  $\lambda$ ). El parámetro de selección de escenario determina las distribuciones utilizadas (Tablas 5 o 3) y activa la mezcla segmentada para tamaños en alta congestión.

La capacidad del bloque ( $C_{block}$ ) es ajustable para análisis de sensibilidad, con valor nominal de 1,000,000 vBytes según especificación BIP 141 (Lombrozo et al., 2015). La semilla aleatoria se configura explícitamente mediante `np.random.seed()` para garantizar réplicas independientes manteniendo reproducibilidad.

### 3.4.3 Criterio de terminación y período de calentamiento

Para garantizar la validez estadística de los resultados minimizando el tiempo computacional, se adoptó un criterio de terminación basado en eventos en lugar de tiempo simulado:

- **Período de calentamiento:** Se establece en 200 bloques minados (aproximadamente 33 horas simuladas), durante los cuales las métricas no se recolectan. Este período permite que el sistema alcance condiciones estacionarias, eliminando el sesgo de condiciones iniciales vacías. La duración se determinó mediante análisis de convergencia de la media móvil en simulaciones piloto.
- **Horizonte de recolección:** Se define como 1,000 bloques adicionales (aproximadamente 7 días simulados), proveyendo una muestra de 2 a 3 millones de transacciones procesadas, suficiente para obtener estimaciones robustas con errores estándar inferiores al 1%.
- **Duración total:** El total de 1,200 bloques (calentamiento + recolección) requiere entre 2 y 4 horas de tiempo computacional según la tasa de llegada configurada.

#### 3.4.4 Instrumentación y recolección de datos

El modelo incorpora instrumentación exhaustiva para capturar las métricas de rendimiento relevantes:

- **Tiempo en sistema:** Se registra el timestamp de llegada ( $t_{arrival}$ ) y confirmación ( $t_{confirmation}$ ) de cada transacción, calculando  $W = t_{confirmation} - t_{arrival}$ .
- **Tamaño de la cola:** Se monitorea continuamente el número de transacciones en espera mediante señales enviadas al workspace de MATLAB.
- **Utilización del servidor:** Se calcula como la fracción del tiempo durante el cual el servidor de minado está activo procesando bloques.
- **Throughput:** Se contabiliza el número de transacciones confirmadas por unidad de tiempo.

### 3.5 Verificación y validación del modelo

La *verificación* asegura que el modelo se implementó correctamente, mientras que la *validación* asegura que representa adecuadamente el sistema real (Sargent, 2013).

#### 3.5.1 Verificación del modelo

La verificación se centró en confirmar que la implementación en Simulink refleja fielmente la lógica conceptual. Se aplicaron las siguientes técnicas:

- **Inspección visual:** Simulaciones en modo de depuración con visualización paso a paso del flujo, verificando la correcta asignación de distribuciones, cálculo de prioridades, ordenamiento de la cola y llenado de bloques.
- **Validación de casos extremos:** Se probaron condiciones límite confirmando comportamiento coherente bajo carga cero (cola vacía), carga extrema (saturación), transacciones grandes ( $vsize = 999,000$  bytes) y pequeñas (200 bytes).
- **Balance de entidades:** Verificaciones automáticas garantizando  $N_{generadas}(t) = N_{cola}(t) + N_{confirmadas}(t)$  y consistencia temporal ( $t_{confirmation} > t_{arrival}$ ).
- **Caso de control M/M/1:** Se configuró el modelo bajo supuestos analíticos (Poisson, exponencial, FIFO, servicio individual), verificando que las métricas convergen a los valores de Pollaczek-Khintchine con desviaciones <5%.

#### 3.5.2 Validación del modelo

La validación evalúa si el modelo reproduce adecuadamente el comportamiento del sistema real mediante técnicas orientadas a datos:

- **Validación de distribuciones:** Se generaron 10,000 muestras sintéticas por variable y se com-

pararon con datos empíricos mediante test K-S (estadísticos  $D$  consistentes con Tablas 5 y 3), comparación de momentos (media, varianza, asimetría y curtosis dentro del IC 95%) y gráficos Q-Q (alineación satisfactoria).

- **Validación de comportamiento:** Se confirmó que el modelo reproduce transacciones por bloque consistentes con estadísticas públicas de Bitcoin (2,000-3,000), tiempos entre bloques exponenciales (media 10 min), y sesgo hacia altas comisiones durante congestión.
- **Validación con expertos:** Se confirmó la política de selección greedy predominante en pools reales, el límite de capacidad de 1M vBytes, y la omisión justificada de mecanismos secundarios (RBF, CPFP) de impacto marginal.
- **Análisis de sensibilidad:** Se mostraron respuestas coherentes ante variaciones de  $\pm 10\%$  en capacidad y  $\pm 20\%$  en tiempo entre bloques.

## 4. Simulación y análisis de resultados

### 4.1 Diseño del Experimento de simulación

Siguiendo los principios del DoE, se construyó un diseño factorial que permite cuantificar efectos principales e interacciones de los factores críticos.

#### 4.1.1 Identificación de factores y niveles

Con base en la literatura y validación en escenarios reales, se identificaron tres factores categóricos que determinan el comportamiento del sistema. La Tabla 6 resume los factores experimentales, sus niveles y la interpretación operacional de cada configuración.

**Table 6.** Factores experimentales y niveles del diseño factorial.

Factor	Nivel	Valor	Interpretación
A: Intensidad de llegadas	Bajo	0.0005 tx/s	Régimen liviano (1.8 tx/s), demanda reducida
	Medio	0.0015 tx/s	Régimen moderado (5.4 tx/s), carga típica
	Alto	0.0040 tx/s	Régimen saturación (14.4 tx/s), pico de demanda
B: Escenario	Baja	Dataset post-halving 2024	Fee-rate bajo, competencia limitada
	Alta	Dataset pico Sep 2024	Fee-rate alto, competencia intensa
C: Capacidad del bloque	Reducida	0.5 MB	Escenario restrictivo (downgrade hipotético)
	Actual	1.0 MB	Capacidad efectiva histórica de Bitcoin
	Duplicada	2.0 MB	Propuesta de escalabilidad (upgrade)
	Duplicada	2.0 MB	Propuesta de escalabilidad (upgrade)

**Nota:** El diseño factorial completo resulta en  $3 \times 2 \times 3 = 18$  combinaciones de tratamiento. Se evaluaron efectos principales ( $\Lambda$ , Escenario, Capacidad), interacciones de segundo orden ( $\Lambda \times$  Escenario,  $\Lambda \times$  Capacidad, Escenario  $\times$  Capacidad) e interacciones de tercer orden ( $\Lambda \times$  Escenario  $\times$  Capacidad).

#### 4.1.2 Estructura factorial y variables de respuesta

El experimento se estructura como un diseño factorial completo  $3 \times 2 \times 3$  con  $n_{rep} = 10$  réplicas independientes por tratamiento ( $N_{total} = 180$  corridas), utilizando diferentes semillas aleatorias para obtener estimaciones estadísticamente robustas. Las variables de respuesta principales incluyen tiempo de espera promedio ( $W$ , segundos), longitud de cola promedio ( $L_q$ , transacciones en mempool), factor de utilización ( $\rho = \lambda S / C_{block}$ ) y throughput ( $T$ , transacciones confirmadas por segundo en régimen estacionario). Estas métricas capturan tanto la experiencia del usuario final (latencia) como la eficiencia operacional del sistema (utilización de recursos).

## 4.2 Análisis estadístico con ANOVA

El ANOVA descompone la variabilidad total en componentes atribuibles a factores, interacciones y error. El modelo estadístico para cada observación  $Y_{ijkl}$  es:

$$Y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + (\beta\gamma)_{jk} + (\alpha\beta\gamma)_{ijk} + \epsilon_{ijkl} \quad (1)$$

donde:

- $\mu$  es la media global de la respuesta
- $\alpha_i, \beta_j, \gamma_k$  son los efectos principales de los factores A, B, C respectivamente
- Los términos de productos ( $\alpha\beta, \alpha\gamma, \beta\gamma, \alpha\beta\gamma$ ) son los efectos de interacción
- $\epsilon_{ijkl} \sim N(0, \sigma^2)$  es el error aleatorio

Para cada efecto, se prueba la hipótesis nula  $H_0$ : efecto = 0 mediante el estadístico  $F = MS_{efecto}/MS_{error}$ , donde se rechaza  $H_0$  si  $p < 0.05$ . La tabla ANOVA descompone la suma de cuadrados total ( $SS_{total}$ ) en componentes atribuibles a cada fuente de variación. Los **supuestos del ANOVA** se verifican mediante:

- **Normalidad:** Gráficos Q-Q de residuos y test de Shapiro-Wilk
- **Homocedasticidad:** Test de Levene para igualdad de varianzas
- **Independencia:** Verificada por aleatorización del orden de ejecución

El **tamaño del efecto** se cuantifica mediante eta-cuadrado parcial  $\eta_{parcial}^2 = SS_{efecto}/(SS_{efecto} + SS_{error})$ , donde valores  $>0.14$  se consideran grandes (Cohen, 1988). Para cada respuesta ( $W$ ,  $N_q$ ,  $\rho$ ,  $T$ ), se ajusta el modelo ANOVA completo, se generan tablas con estadísticos F y valores-p, se diagnostican supuestos, y se aplican tests post-hoc (Tukey HSD) para comparaciones múltiples.

### 4.3 Interpretación gráfica de los resultados

La visualización complementa el análisis ANOVA facilitando la interpretación de patrones complejos mediante las siguientes técnicas:

#### 4.3.1 Gráficos de efectos principales

Muestran la media de la respuesta en cada nivel del factor con barras de error ( $SE = s/\sqrt{n_{rep}}$ ), permitiendo identificar:

- La dirección del efecto (incremento o decremento en la respuesta)
- La magnitud del cambio entre niveles
- Relaciones no-lineales (efectos de umbral o saturación)

#### 4.3.2 Gráficos de interacción

Para cada par de factores, se construirán gráficos donde el eje horizontal representa los niveles de un factor, el eje vertical la media de la respuesta, y diferentes líneas/colores representan los niveles del segundo factor. La presencia de *no-paralelismo* entre las líneas indica interacción:

- **Líneas paralelas:** Ausencia de interacción - el efecto de un factor es constante independientemente del nivel del otro.
- **Líneas no paralelas:** Presencia de interacción - el efecto de un factor depende del nivel del otro.
- **Líneas que se cruzan:** Interacción fuerte - el efecto de un factor se invierte según el nivel del otro.

Estos gráficos son particularmente valiosos para interpretar las interacciones significativas identificadas en el ANOVA.

#### 4.3.3 Mapas de calor (heatmaps)

Para visualizar simultáneamente el efecto de dos factores sobre una respuesta, se emplearán mapas de calor bidimensionales donde los ejes representan los niveles de dos factores y el color de cada celda representa la magnitud de la respuesta. Esta representación es especialmente útil para identificar:

- Regiones del espacio de factores asociadas con rendimiento óptimo o degradado.
- Patrones no lineales o de umbral en las respuestas.
- Trade-offs entre diferentes configuraciones del sistema.

#### 4.3.4 Diagramas de caja (boxplots) comparativos

Para comparar la distribución completa de una respuesta (no solo su media) a través de los niveles de un factor, se emplearán diagramas de caja agrupados. Estos gráficos revelan:

- La mediana, cuartiles y rango de la respuesta en cada nivel.
- La presencia de asimetría o valores atípicos.
- Diferencias en la variabilidad (dispersión) entre niveles, complementando el análisis de homocedasticidad.

#### 4.3.5 Series temporales de métricas

Para entender la dinámica temporal del sistema, se graficarán series temporales del tamaño de la cola ( $N_q(t)$ ) bajo diferentes configuraciones representativas. Estas visualizaciones permiten:

- Identificar si el sistema alcanza un estado estacionario o presenta tendencias de crecimiento/decrecimiento.
- Detectar comportamientos transitorios, oscilaciones o eventos de congestión severa.
- Evaluar la efectividad del período de calentamiento establecido.

La combinación de estas técnicas de visualización complementará el análisis estadístico formal, proporcionando una comprensión integral y matizada del comportamiento del sistema bajo las condiciones experimentales exploradas.

### 4.4 Resultados experimentales

Se ejecutaron 180 corridas del diseño factorial completo ( $3 \times 2 \times 3 \times 10$  réplicas) durante aproximadamente 2.5 horas de tiempo computacional. El diseño experimental incluyó tres factores lambda ( $f_\lambda \in \{0.0005, 0.0015, 0.004\}$ ), dos escenarios de congestión (baja/alta), y tres capacidades de bloque (0.5, 1.0, 2.0 MB), resultando en 18 configuraciones únicas con 10 réplicas cada una. Este tamaño muestral proporciona potencia estadística  $> 0.80$  para detectar efectos significativos en un ANOVA factorial con  $\alpha = 0.05$ .

#### 4.4.1 Estadísticas descriptivas globales

La Tabla 7 presenta las estadísticas descriptivas de las variables de respuesta agregadas sobre las 142 corridas que presentaron congestión medible ( $W > 0$ ).

**Table 7.** Estadísticas descriptivas de variables de respuesta (142 corridas con congestión).

Variable	Media	Desv. Est.	Mínimo	Mediana	Máximo	Unidad
$W$ (Tiempo espera)	6,435	9,417	23	1,854	46,695	segundos
$W_{P95}$ (Percentil 95)	28,719	—	—	—	—	segundos
$L_q$ (Longitud cola)	13,996	34,802	12	1,739	300,928	transacciones
$T$ (Throughput)	1.057	1.163	0.001	0.573	5.916	txs/segundo
$N_{txs}$ (Total procesadas)	76,124	83,733	38	41,225	426,182	transacciones

**Nota:** El 78.9% de las corridas (142/180) presentaron congestión medible. Las configuraciones subsaturadas ( $p < 0.1$ ) resultaron en  $W \approx 0$  y  $L_q \approx 0$ , confirmando el comportamiento teórico esperado. Total de transacciones procesadas: 13,702,253.

El tiempo de espera promedio alcanzó 107 minutos, con casos extremos superando las 12.9 horas (46,695 segundos) en configuraciones sobresaturadas. La longitud de cola promedio fue de 13,996 transacciones, con picos de hasta 300,928 transacciones pendientes en configuraciones con  $p > 1.0$ . Esta alta variabilidad refleja el rango amplio de factores de utilización cubiertos por el diseño experimental ( $p \in [0.0001, 2.3]$ ).

#### 4.4.2 Efectos principales de los factores

La Tabla 8 desagrega las métricas de respuesta por nivel de cada factor experimental, revelando patrones consistentes con la teoría de colas.

**Table 8.** Efectos principales de factores sobre variables de respuesta (media  $\pm$  desv.est.).

Factor / Nivel	W (seg)	L <sub>q</sub> (txs)	T (txs/s)
<i>Factor Lambda f<sub>λ</sub></i>			
0.0005 (alta carga)	10,170 $\pm$ 12,521	26,268 $\pm$ 45,621	1.474
0.0015 (media)	6,323 $\pm$ 8,111	13,527 $\pm$ 30,568	1.115
0.004 (baja carga)	2,811 $\pm$ 4,351	2,191 $\pm$ 5,438	0.583
<i>Escenario de congestión</i>			
1 (Baja congestión)	3,380 $\pm$ 4,819	4,261 $\pm$ 12,017	0.870
2 (Alta congestión)	9,490 $\pm$ 11,673	23,730 $\pm$ 46,020	1.245
<i>Capacidad de bloque C<sub>bloque</sub></i>			
0.5 MB	10,977 $\pm$ 11,428	22,502 $\pm$ 40,827	0.781
1.0 MB	6,228 $\pm$ 9,164	14,195 $\pm$ 33,732	1.122
2.0 MB	2,100 $\pm$ 3,786	5,290 $\pm$ 14,326	1.268

**Nota:** Los promedios incluyen todas las 180 corridas. Factor lambda muestra efecto decreciente en W y L<sub>q</sub> a medida que aumenta (menor tasa de llegada). Escenario 2 (alta congestión) casi triplica el tiempo de espera respecto a Escenario 1. Capacidad de 2 MB reduce dramáticamente la congestión.

Los patrones observados confirman la validez del modelo:

- **Factor f<sub>λ</sub>:** A menor valor (mayor tasa de llegada), mayor congestión. f<sub>λ</sub> = 0.0005 produce W 262% mayor que f<sub>λ</sub> = 0.004.
- **Escenario:** El escenario 2 (alta congestión) genera W 181% superior al escenario 1, consistente con diferencias en distribuciones empíricas subyacentes (tamaños de transacción mayores).
- **Capacidad:** El aumento de 0.5 MB a 2 MB reduce W en 81% y L<sub>q</sub> en 76%, demostrando el impacto directo de la capacidad del bloque sobre la congestión.

#### 4.4.3 Configuraciones con mayor congestión

La Tabla 9 identifica las cinco configuraciones que experimentaron los mayores tiempos de espera, revelando las condiciones bajo las cuales la mempool alcanza saturación crítica.

**Table 9.** Top 5 configuraciones con mayor congestión.

f <sub>λ</sub>	Escenario	Cap (MB)	W (seg)	L <sub>q</sub> (txs)	N <sub>txs</sub>
0.0005	2	0.5	46,695	57,214	174,086
0.0005	2	0.5	44,501	40,155	148,892
0.0005	2	1.0	40,943	132,893	426,182
0.0015	2	0.5	33,505	82,094	169,623
0.0005	2	1.0	33,315	109,517	358,623

**Nota:** La configuración más severa (f<sub>λ</sub> = 0.0005, Esc=2, Cap=0.5MB) generó tiempos de espera superiores a 12.9 horas (46,695 seg) y colas de más de 57,000 transacciones, indicando sobresaturación del sistema ( $\rho > 1.5$ ).

La configuración más extrema ( $f_\lambda = 0.0005$ , Escenario 2, Capacidad 0.5 MB) representa un caso de **sobresaturación sostenida** donde la tasa de llegada supera la capacidad de procesamiento del bloque. En este régimen, las colas crecen de forma no acotada, generando tiempos de espera de hasta 12.97 horas y acumulaciones de hasta 132,893 transacciones pendientes.

El análisis de estas configuraciones extremas confirma que el modelo captura correctamente el comportamiento de sistemas M/G/1 en régimen de saturación, donde la variabilidad en los tiempos de servicio (tamaños de transacción) amplifica la congestión más allá de lo predicho por modelos M/M/1 simples.

#### 4.4.4 Análisis de varianza factorial

Para cuantificar la significancia estadística de los efectos observados y determinar la proporción de variabilidad explicada por cada factor, se aplicó un análisis de varianza (ANOVA) factorial de tres vías sobre el tiempo de espera promedio  $W$  como variable respuesta. El análisis utilizó las 142 corridas con congestión medible ( $W > 0$ ), excluyendo configuraciones subsaturadas que exhiben censura a izquierda.

**Efectos principales y significancia estadística.** Los tres factores experimentales demostraron efectos altamente significativos (Tabla 10):

**Table 10.** Resultados del ANOVA factorial: significancia de efectos principales.

Factor	F-statistic	p-value	$\eta^2$	Significancia
Lambda ( $f_\lambda$ )	11.32	0.000028	0.140	***
Escenario	29.15	< 0.000001	0.172	***
Capacidad ( $C_{block}$ )	16.29	< 0.000001	0.190	***

**Nota:** Todos los efectos principales son estadísticamente significativos a nivel  $\alpha = 0.001$  (\*\*\*) . Los valores de  $\eta^2$  (eta cuadrado) representan la proporción de varianza explicada por cada factor.

El factor con mayor poder explicativo es la capacidad del bloque ( $\eta^2 = 0.190$ ), seguido por el escenario de congestión ( $\eta^2 = 0.172$ ) y la tasa de llegada ( $\eta^2 = 0.140$ ). Los tres factores en conjunto explican el 50.2% de la varianza total en el tiempo de espera, con el 49.8% restante atribuible a variabilidad aleatoria inherente al proceso estocástico y posibles interacciones de orden superior.

**Comparaciones post-hoc.** Se aplicó el test de Tukey para identificar diferencias significativas entre niveles específicos de cada factor. Para el factor lambda, todas las comparaciones pareadas resultaron significativas:  $f_\lambda = 0.0005$  vs 0.004 mostró la mayor diferencia ( $\Delta W = 9,150$  seg,  $p < 0.001$ ), mientras que  $f_\lambda = 0.0005$  vs 0.0015 presentó diferencia moderada ( $\Delta W = 5,544$  seg,  $p = 0.012$ ).

Para la capacidad del bloque, la comparación entre 0.5 MB y 2.0 MB fue la más significativa ( $\Delta W = 10,289$  seg,  $p < 0.001$ ), mientras que la diferencia entre 0.5 MB y 1.0 MB no alcanzó significancia al nivel estricto ( $p = 0.059$ ), sugiriendo que duplicar la capacidad desde niveles bajos tiene mayor impacto que incrementos desde niveles moderados.

**Análisis de interacciones.** El análisis reveló patrones de interacción entre factores que modifican sus efectos individuales. La interacción Lambda  $\times$  Escenario mostró que el efecto de la tasa de llegada se amplifica dramáticamente en el escenario de alta congestión: con  $f_\lambda = 0.0005$ , el Escenario 2

generó  $W = 21,469$  seg vs  $W = 6,129$  seg en Escenario 1 (multiplicador 3.5×), mientras que con  $f_\lambda = 0.004$  la diferencia fue menor (multiplicador 2.5×).

La interacción Lambda × Capacidad evidenció que reducir la capacidad del bloque tiene efecto marginal bajo baja carga ( $f_\lambda = 0.004$ : diferencia 0.5MB vs 2.0MB = 6,721 seg), pero se vuelve crítica bajo alta carga ( $f_\lambda = 0.0005$ : diferencia = 14,164 seg). Esto confirma que la capacidad del bloque es el cuello de botella dominante únicamente cuando la demanda es suficientemente alta para saturar el sistema.

## 5. Conclusiones

### 5.1 Síntesis de los hallazgos principales

La experimentación mediante simulación de eventos discretos produjo resultados estadísticamente robustos que confirman la capacidad del modelo para capturar la dinámica de congestión en sistemas blockchain con disciplina de cola por prioridad.

#### 5.1.1 Hallazgos fundamentales sobre congestión

El experimento factorial de 180 corridas reveló patrones coherentes con la teoría de colas M/G/1 con prioridades. El 78.9% de las corridas experimentaron congestión medible ( $W > 0$ ), con tiempos de espera entre 23 segundos y 46,695 segundos (12.97 horas), demostrando que el modelo captura correctamente la transición entre regímenes subsaturados y sobresaturados.

Las configuraciones con factor de utilización  $\rho > 1.0$  generaron colas de hasta 300,928 transacciones y tiempos de espera superiores a 12 horas, confirmando el comportamiento teórico de sistemas sobresaturados donde las colas crecen sin cota superior. La alta desviación estándar observada en  $W$  (9,417 seg) y  $L_q$  (34,802 txs) refleja el efecto amplificador de la variabilidad en tamaños de transacción sobre la congestión, fenómeno consistente con la fórmula de Pollaczek-Khinchin para sistemas M/G/1 que incorpora el coeficiente de variación del servicio.

El volumen total procesado alcanzó 13,702,253 transacciones con throughput promedio de 1.057 txs/seg, alcanzando valores extremos de 5.916 txs/seg en configuraciones de alta capacidad y baja carga. Esta variabilidad en el throughput evidencia que el sistema no opera a capacidad constante sino que se adapta dinámicamente a las condiciones de demanda.

#### 5.1.2 Efectos de factores experimentales

El análisis ANOVA demostró que los tres factores evaluados ejercen efectos estadísticamente significativos ( $p < 0.001$ ) sobre las métricas de congestión. Reducir el factor lambda de 0.004 a 0.0005 (multiplicador de 8×) incrementó  $W$  en 262% y  $L_q$  en 1100%, demostrando sensibilidad sustancial del sistema a variaciones en la demanda. Este efecto se explica porque incrementos en la tasa de llegada acercan el sistema al régimen de saturación ( $\rho \rightarrow 1$ ), donde pequeñas variaciones en  $\lambda$  producen incrementos no lineales en las métricas de congestión según las fórmulas de teoría de colas.

El escenario de congestión mostró el segundo efecto más significativo, con el Escenario 2 generando  $W$  181% superior al Escenario 1. Este fenómeno se atribuye a diferencias estructurales en las distribuciones empíricas: el Escenario 2 presenta transacciones de mayor tamaño promedio (distribución segmentada con componente Pareto), reduciendo el número de transacciones que caben por bloque y consecuentemente incrementando el tiempo de espera.

La capacidad del bloque emergió como el factor más influyente ( $\eta^2 = 0.190$ ), con incrementos de 0.5 MB a 2.0 MB reduciendo  $W$  en 81% y  $L_q$  en 76%. Este resultado confirma que la capacidad del bloque constituye el cuello de botella crítico del sistema, validando hipótesis previas sobre el papel central de este parámetro de protocolo en la experiencia del usuario.

### 5.1.3 Validación del modelo de procesamiento por lotes

La implementación de procesamiento por lotes con intervalos fijos de 600 segundos demostró ser esencial para capturar el comportamiento real del sistema. Este diseño permitió observar acumulación de transacciones entre bloques consecutivos, implementar correctamente la disciplina de prioridad por fee-rate mediante ordenamiento explícito, y modelar el efecto de la capacidad limitada del bloque mediante algoritmo knapsack simplificado que selecciona transacciones hasta saturar el límite de bytes disponibles.

Las métricas generadas ( $W, L_q$ ) resultaron coherentes con sistemas M/G/1 en diversos regímenes de utilización, desde configuraciones subsaturadas ( $\rho < 0.1$ ) con  $W \approx 0$  hasta configuraciones sobresaturadas ( $\rho > 1.5$ ) con crecimiento no acotado de colas. El contraste con modelos de servicio continuo evidenció que la representación explícita de la discretización temporal es crítica para la fidelidad del modelo, ya que aproximaciones basadas en tasas promedio no capturan adecuadamente la dinámica de acumulación y liberación periódica característica de sistemas blockchain.

## 5.2 Validación de la metodología desarrollada

Independientemente de los resultados numéricos específicos obtenidos, este trabajo establece la validez de la simulación de eventos discretos como herramienta metodológica para el estudio de sistemas blockchain complejos.

### 5.2.1 Framework de ajuste de distribuciones

Integrando validación cruzada k-fold, criterios de información múltiples (BIC, AIC, HQIC), bootstrap paramétrico y detección automática de overfitting, el framework desarrollado constituye una aproximación defensiva que prioriza capacidad de generalización sobre ajuste nominal. Los resultados (Tablas 5 y 3) evidencian equilibrio entre parsimonia estadística y fidelidad representativa, siendo generalizable a otros dominios con alta variabilidad, colas pesadas o heterogeneidad estructural.

### 5.2.2 Modelo de simulación con procesamiento por lotes

El modelo implementado captura correctamente la dinámica de un sistema de colas M/G/1 con prioridades y servicio por lotes. Los resultados experimentales confirman que las configuraciones con  $\rho > 1.0$  generan colas crecientes y tiempos de espera no acotados, mientras que configuraciones con  $0.3 < \rho < 0.9$  producen congestión moderada ( $W = 1-8$  horas) y configuraciones con  $\rho < 0.1$  operan sin formación de colas ( $W \approx 0$ ).

El proceso de verificación validó que el modelo reproduce correctamente el comportamiento cualitativo de sistemas de colas bajo diferentes regímenes de carga, estableciendo su credibilidad como herramienta predictiva para análisis de políticas de protocolo y evaluación de propuestas de escalabilidad.

- Las configuraciones con  $\rho > 1.0$  generan colas crecientes y tiempos de espera no acotados (consistente con teoría)
- Las configuraciones con  $0.3 < \rho < 0.9$  producen congestión moderada ( $W = 1-8$  horas)
- Las configuraciones con  $\rho < 0.1$  operan sin formación de colas ( $W \approx 0$ )

El proceso de verificación validó que el modelo reproduce correctamente el comportamiento cualitativo de sistemas de colas bajo diferentes regímenes de carga, estableciendo su credibilidad como herramienta predictiva.

### 5.2.3 Diseño experimental factorial

El diseño experimental  $3 \times 2 \times 3 \times 10$  (factores lambda  $\times$  escenarios  $\times$  capacidades  $\times$  réplicas = 180 corridas) proporciona potencia estadística  $> 0.80$  para detectar efectos significativos ( $\alpha = 0.05$ ) en un ANOVA factorial de tres vías. El protocolo de ejecución con período de warmup (18,000 seg = 30 bloques) y ventana de recolección (72,000 seg = 120 bloques) garantizó que las estimaciones de métricas fueran insesgadas y precisas, eliminando efectos transitorios iniciales.

La ejecución del diseño completo procesó un total de 13,702,253 transacciones en aproximadamente 2.5 horas de tiempo computacional, demostrando la viabilidad práctica del enfoque de simulación para estudios experimentales de sistemas blockchain.

## 5.3 Limitaciones del estudio

Aunque el modelo demostró coherencia con teoría de colas (validación M/M/1, M/G/1) y capacidad para capturar regímenes de congestión, subsisten limitaciones técnicas y conceptuales. La calibración de  $\lambda$  mediante datasets históricos asume estacionariedad dentro del período de recolección, omitiendo ciclos diarios, patrones correlacionados con eventos externos (lanzamiento de NFTs, picos en Runes) y dependencias temporales entre llegadas. Modelar esta no-estacionariedad requeriría procesos de Poisson no-homogéneos (NHPP) o cadenas de Markov ocultas (HMM) con transiciones entre regímenes. Adicionalmente, la validación empírica con muestras de 24-72 horas podría extenderse a períodos más largos para detectar tendencias estacionales y evaluar robustez de parámetros ante variaciones de largo plazo.

El modelo de servicio simplificado representa tiempos entre bloques como constantes (600 segundos), omitiendo la variabilidad estocástica real (distribución exponencial) que transformaría el sistema de D/G/1 a M/G/1 y afectaría la varianza de  $W$  y  $L_q$ . El algoritmo de selección knapsack es determinístico, pero en realidad los mineros pueden aplicar políticas heterogéneas (incluir transacciones de bajo fee por incentivos estratégicos). Tampoco se modelaron latencias de propagación en red P2P (1-10 segundos típicos) ni probabilidad de reorganizaciones de cadena (1-2% para bloques recientes), fenómenos que introducen incertidumbre adicional en el tiempo efectivo de confirmación.

Finalmente, varias simplificaciones conceptuales reducen realismo pero permiten análisis controlado: mempool infinita versus límites reales de 300 MB en Bitcoin Core, ausencia de mecanismos RBF/CPFP que introducen realimentación y correlaciones temporales, distribuciones de *fee\_rate* históricas que no adaptan a condiciones en tiempo real, y duración de simulación (120 bloques) potencialmente insuficiente para caracterizar regímenes con  $\rho > 1.5$  donde el sistema exhibe crecimiento ilimitado de cola. El diseño factorial inicial fue optimizado para potencia estadística, pero no se aplicaron técnicas de reducción de varianza (common random numbers, antithetic variates) ni muestreo adaptativo (alocar más réplicas a configuraciones con mayor variabilidad) que podrían mejorar eficiencia sin incrementar costo computacional.

## 5.4 Trabajo futuro

Aunque el diseño factorial completo de 180 corridas proporciona potencia estadística adecuada, extensiones experimentales podrían enriquecer el análisis: incrementar de 10 a 20 réplicas por configuración para reducir ancho de intervalos de confianza, extender ventana de observación a 500-1000 bloques para caracterizar completamente regímenes con  $\rho > 1.5$ , e incorporar métricas adicionales como percentiles de tiempo de espera (P50, P90, P99) y correlación entre *fee\_rate* y  $W$  (equidad del sistema). Complementariamente, el modelo de simulación puede extenderse para capturar mecanismos adicionales del sistema real: RBF (Replace-by-Fee) permitiendo que transacciones actualicen dinámicamente su *fee\_rate* según estrategias de usuarios, dependencias CPFP (Child-Pays-For-Parent) con restricciones de precedencia en selección de lotes, red P2P multi-nodo con mempools

locales heterogéneas para estudiar equidad y vulnerabilidades de censura, procesos no-estacionarios mediante Poisson no-homogéneo o modelos de Markov ocultos capturando ciclos diarios, y mempool con capacidad finita (300 MB límite) con políticas de expulsión analizando escenarios de congestión extrema.

La metodología puede extenderse a problemas de diseño y evaluación de protocolos: cuantificar trade-offs de incrementos de capacidad (1M, 2M, 4M, 8M vBytes) versus soluciones Layer-2 (Lightning Network), formular bidding como juego no-cooperativo derivando equilibrios de Nash y analizando eficiencia social, generalizar el framework a otras blockchains (Ethereum con gas price auction, Bitcoin Cash con bloques de 32 MB, Cardano con stake pools) realizando estudios comparativos, proyectar sostenibilidad económica post-halving evaluando ingresos por comisiones bajo diferentes escenarios de adopción, y simular ataques de spam o front-running (MEV) evaluando efectividad de contramedidas (rate limiting, priority gas auctions).

Finalmente, las lecciones aprendidas sugieren mejoras metodológicas: desarrollar suite de verificación automática con casos analíticos ( $M/M/1$ ,  $M/M/c$ ,  $M/G/1$ ) integrable en CI/CD, complementar diseño factorial con análisis de Sobol (descomposición de varianza global) cuantificando contribución individual e interactiva de parámetros continuos, emplear optimización bayesiana (Gaussian Process Regression) para explorar eficientemente el espacio de configuraciones y localizar regímenes de interés ( $\rho \approx 0.9$ ) sin enumeración exhaustiva, y ajustar metamodelos estadísticos (superficies de respuesta, kriging) para predicción rápida de métricas en configuraciones no simuladas, útil para optimización y análisis de riesgo.

## 6. Agradecimientos

El autor desea expresar su profundo agradecimiento al Profesor Francisco Roqué por su guía, orientación metodológica y feedback durante el desarrollo de este trabajo de investigación. Sus enseñanzas sobre simulación de sistemas discretos y diseño experimental han sido fundamentales para la realización de este estudio.

## 7. Contribuciones del autor

Simón Tadeo Ocampo: Conceptualización, metodología, desarrollo de software, análisis formal, investigación, redacción - borrador original, redacción - revisión y edición, visualización.

## 8. Funding statement

Esta investigación no recibió financiación externa.

## References

- Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716–723.
- Burnham, K. P., & Anderson, D. R. (2002). *Model selection and multimodel inference: A practical information-theoretic approach*. Springer Science & Business Media.
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd). Lawrence Erlbaum Associates.
- Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman; Hall/CRC.
- Johnson, N. L. (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, 36(1/2), 149–176.

- Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430), 773–795.
- Law, A. M. (2014). *Simulation modeling and analysis* (5th). McGraw-Hill Education.
- Lombrozo, E., Lau, J., & Wuille, P. (2015). Segregated witness (consensus layer) [Disponible en: <https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki>].
- Massey Jr, F. J. (1951). The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253), 68–78.
- Matloff, N. (2008). *Introduction to discrete-event simulation and the simpy language* [Disponible en: <http://heather.cs.ucdavis.edu/~matloff/156/PLN/DESimIntro.pdf>]. University of California, Davis.
- Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system [Whitepaper. Disponible en: <https://bitcoin.org/bitcoin.pdf>].
- Sargent, R. G. (2013). Verification and validation of simulation models. *2013 Winter Simulations Conference (WSC)*, 321–337.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2), 461–464.