

UNIVERSIDAD CATÓLICA BOLIVIANA
SAN PABLO
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS

**Simulación y Análisis de un
Centro de Llamadas**

Modelado en Python, R y Stella

INTEGRANTES:

Christian Mijael Mendoza
Huanca
Cesar Mateo Vera Andrade
Manuel Fernando Delgadillo
Calderon

DOCENTE:

Dr. Juan Carlos Flores
Lopez

ASIGNATURA:

Modelado, Dinámica de
Sistema y Simulación

FECHA:

14/06/25

La Paz – Bolivia
Gestión 2025

Índice

1. Introducción	4
2. Sobre el Dataset	4
3. Objetivo del Análisis	4
4. Marco Teórico	6
4.1. Simulación de Eventos Discretos	6
4.2. Modelos de Servicio Estocásticos	6
4.3. Dinámica de Sistemas en Stella	6
5. Metodología	7
5.1. Preprocesamiento de Datos Reales	7
5.2. Simulación en Python	7
5.3. Simulación en R	8
5.4. Entrenamiento de Random Forest	8
5.5. Modelado en Stella	9
5.6. Dashboard Operacional Interactivo	10
6. Resultados	12
6.1. Desempeño del Clasificador	12
7. Visualización Exploratoria	13
7.1. Volumen de Llamadas Diarias	13
7.2. Porcentaje de Llamadas que no Cumplen SLA	14
7.3. Comparación de Distribución de Tiempo de Servicio	15
7.4. Cumplimiento SLA Real	16
7.5. Densidad de Tiempos de Espera	17
7.6. Distribución de Tiempos de Espera según Cumplimiento SLA	18
7.7. Distribución Lognormal: CDF Empírica vs Teórica	19
7.8. Distribución Lognormal: PDF Empírica vs Teórica	20
7.9. Distribución de Tiempos de Espera	21
7.10. Importancia de Variables (Modelo Real)	22
8. Comparación de Distribuciones	23
8.1. Tiempos de Servicio: Real vs Simulado	23
8.2. Importancia de Variables (Modelo Simulado)	24

8.3.	Matriz de Confusión (Datos Reales)	25
8.4.	Matriz de Confusión (Datos Simulados)	26
8.5.	Probabilidad de Cumplir SLA según Hora	27
8.6.	Comparación de % Cumplimiento SLA Diario	28
9.	Investigación Académica	29
9.1.	Alsamadi et al. (2022) — Evaluación de Turnos de Agentes mediante DES	29
9.2.	Koole et al. (2024) — Validación de Modelos de Simulación con Datos Reales	29
9.3.	Serper et al. (2022) — Optimización con Arena y Big Data . .	30
9.4.	Alsamadi et al. (2025) — ML para Dimensionamiento de Agentes	31
9.5.	Hou et al. (2021) — Predicción Data-Driven de SLA	31

Resumen

Este documento presenta una comparativa de simulación de eventos discretos implementada en **Python** (con **simpy**) y en **R** (con **simmer**), junto con un modelo de dinámica de sistemas en **Stella** para un centro de llamadas. Se contrastan los resultados frente a un dataset real y se evalúa un clasificador Random Forest para predecir el cumplimiento del SLA (90 % de llamadas atendidas en ≤ 1 minuto). Además, se ha desplegado un *dashboard* operativo en línea que permite al Gerente Operativo interactuar con los KPIs—cumplimiento de SLA (Tiempo mínimo de espera), tiempos de espera y volúmenes de llamadas—de forma clara y dinámica. Incluye 16 visualizaciones detalladas y una revisión de la literatura académica relevante.

1. Introducción

La gestión eficiente de colas de llamadas es esencial para mantener un alto nivel de servicio al cliente. Tiempos de espera excesivos pueden traducirse en pérdida de clientes y costos operativos mayores. La simulación y el análisis de datos permiten evaluar políticas de asignación de recursos y prever el comportamiento del sistema en distintos escenarios.

2. Sobre el Dataset

Call Centre Queue Simulation es un dataset simulado diseñado como recurso didáctico para *Business and Operations Analytics*. La simulación cubre un año de operación, considerando un centro de llamadas abierto de 8:00 am a 6:00 pm, de lunes a viernes.

- **Agentes:** Cuatro agentes disponibles continuamente.
- **Tiempo de servicio promedio:** 5 minutos por llamada.
- **Objetivo de desempeño (SLA):** 90 % de las llamadas deben ser contestadas en menos de 1 minuto.
- **Registro:** Se almacenan las marcas de tiempo de *llamada iniciada*, *llamada contestada* y *llamada finalizada*. A partir de ellas se calculan *waiting-time* y *service-time*, y se marca si cumple el estándar.
- **Herramienta original:** El dataset y las simulaciones se generaron en R usando el paquete `simmer` (Ucar, Smeets & Azcorra, 2019).

Para más detalles y código fuente, consultar el notebook en GitHub y la aplicación Shiny asociada.

3. Objetivo del Análisis

Evaluar y comparar el desempeño de un centro de llamadas usando simulación estocástica y modelos de machine learning, con el fin de proponer estrategias que permitan cumplir el SLA (90 % de llamadas atendidas en menos de 1 minuto).

1. Modelar el comportamiento realista diario mediante simulación de eventos discretos.
2. Comparar la precisión del nivel de servicio (SLA) utilizando datos reales y simulados.
3. Entrenar y evaluar modelos de ML sobre ambos conjuntos para medir su capacidad predictiva del cumplimiento del SLA.

4. Marco Teórico

4.1. Simulación de Eventos Discretos

Un sistema de eventos discretos se caracteriza por cambios de estado en instantes puntuales. **Llegadas:** modeladas como un proceso de Poisson no homogéneo con tasa $\lambda(t)$; los intervalos inter-arribos τ se muestrean mediante:

$$\tau = -\frac{1}{\lambda} \ln(1 - U), \quad U \sim \mathcal{U}(0, 1).$$

4.2. Modelos de Servicio Estocásticos

Para reproducir la variabilidad observada en datos reales, el tiempo de servicio S se define como:

$$S = \begin{cases} \text{Exp}(2\mu), & \text{con probabilidad } 0,2, \\ \text{LogNormal}(\mu, \sigma), & \text{con probabilidad } 0,8, \end{cases}$$

con

$$\mu = \ln\left(\frac{\text{mean}^2}{\sqrt{\text{std}^2 + \text{mean}^2}}\right), \quad \sigma = \sqrt{\ln\left(1 + \frac{\text{std}^2}{\text{mean}^2}\right)}.$$

4.3. Dinámica de Sistemas en Stella

Stella permite modelar mediante:

- **Stock CallsWaiting:** acumula la diferencia entre llegadas y atenciones.
- **Flow Llegadas:** determinado por `ArrivalRate`.
- **Flow Served:**

$$\text{Served} = \min(\text{CallsWaiting}, \text{Agents}/\text{ServiceTime}).$$

- **Convertidores:** parámetros exógenos como `SpikeFactor`, `BreakStart`, `BreakDuration`.

5. Metodología

5.1. Preprocesamiento de Datos Reales

El archivo `call_centre_logs.csv` contiene:

- `date`, `daily_caller`
- `call_started`, `call_answered`, `call_ended` (horas AM/PM)
- `wait_length`, `service_length` (segundos)
- `meets_standard` (TRUE/FALSE)

Pasos:

1. Conversión de segundos a minutos.
2. Parseo de marcas de tiempo a `datetime`.
3. Extracción de `arrival_minute`, `day_of_week`, `month`.
4. Codificación de `meets_standard` a variable binaria `target`.

5.2. Simulación en Python

Con `simpy`:

- **Recursos:** 4 agentes (`counter = Resource(4)`).
- **Descansos:** en $t = \text{BreakStart} \sim U(100, 400)$ se retiran 2 agentes; regresan tras $\text{BreakDuration} \sim U(20, 60)$.
- **Llegadas:** tasa base $\lambda_{\text{base}} = \text{daily_calls}/600$, ajustada por $\text{SpikeFactor} \sim N(1, 0, 3)$; 10 % de días con factor adicional en $[2, 4]$.
- **Inter-arribos:** $\tau \sim \text{exp_rv}(1/\lambda)$.
- **Cliente:** solicita agente, mide `waiting_time`, muestrea `service_time`, libera agente.
- **Registro:** almacena `date`, `arrival_time`, `waiting_time`, `service_time`.

5.3. Simulación en R

Con el paquete `simmer` en R:

- **Entorno de simulación:** `env <- simmer(CallCenter")`
- **Recursos:** `env %>% add_resource(.agent", 4)` — cuatro agentes disponibles.
- **Descansos:** Dentro de la `trajectory` de cada agente se inserta un `timeout` en $t \sim U(100, 400)$ para remover 2 agentes, y otro `timeout` en $t \sim U(20, 60)$ para devolverlos.
- **Llegadas:** `add_generator(caller", traj, function() rexp(1, rate = (t)))`, con $\lambda_{\text{base}} = \text{daily_calls}/600$ y un `SpikeFactor` aplicado con `rnorm(1,1,0.3)`; en 10 % de los días se multiplica por un factor uniforme en $[2, 4]$.
- **Interarribos:** $\tau \sim \text{rexp}(1, \lambda)$ usando la transformada inversa internamente.
- **Cliente:** En la `trajectory`:
 1. `seize(.agent",1)`
 2. `timeout(function() sample_service())`
 3. `release(.agent",1)`
- **Monitoreo:** Se registran `start_time`, `waiting_time` y `service_time` con `get_mon_arrivals(env)`.

5.4. Entrenamiento de Random Forest

- **Características:** `{arrival_minute, service_time, day_of_week, month}`.
- **Etiqueta:** `target` (0 = incumple, 1 = cumple).
- **Parámetros:** 100 árboles, `class_weight='balanced'`.
- **Evaluación:** precisión, recall, f1-score, accuracy.

5.5. Modelado en Stella

El modelo de dinámica de colas fue implementado íntegramente en Stella, generando tanto el archivo de proyecto ('simulacion proyecto.STMX') como la siguiente gráfica. Los pasos fueron:

1. Crear el *Stock CallsWaiting* que acumula el número de llamadas en espera.
2. Crear los *Flows*
 - **Llegadas**: alimenta el stock según la tasa de llegada (**ArrivalRate**).
 - **Served**: drena el stock de acuerdo al número de agentes y el tiempo de servicio (**ServiceTime**).
3. Definir los *Convertidores* (parámetros externos): **ArrivalRate**, **ServiceTime**, **Agents**, **SpikeFactor**, **BreakStart** y **BreakDuration**.
4. Configurar las especificaciones de ejecución (*Run Specs*): simular 600 minutos con paso de integración de 0.1 minutos.
5. Añadir gráficas y tablas al canvas de Stella: **CallsWaiting** (stock), **ArrivalRate** y **Served** (flows).
6. Guardar el modelo en el archivo 'simulacion proyecto.STMX' y exportar la imagen generada en formato PNG para incluirla en este informe.

CallsWaiting (línea azul) muestra cómo las llamadas en espera se elevan rápidamente al inicio y luego se estabilizan;

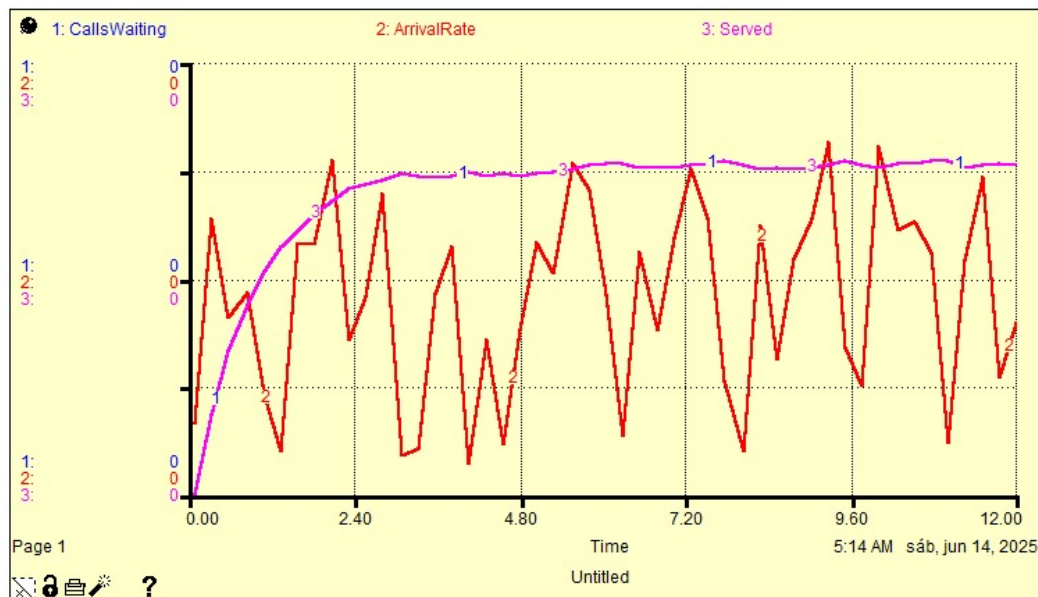


Figura 1: Evolución temporal de las tres series clave en Stella: **CallsWaiting** (línea azul) muestra cómo las llamadas en espera se elevan rápidamente al inicio y luego se estabilizan; **ArrivalRate** (línea roja) presenta picos periódicos que reflejan la variabilidad en la llegada de llamadas; **Served** (línea magenta) permanece casi plana, indicando la capacidad de atención constante de los agentes.

5.6. Dashboard Operacional Interactivo

Para apoyar al **Gerente Operativo** en la visualización clara e interactiva de los principales KPIs (cumplimiento de SLA, tiempos de espera, volumen de llamadas), se ha desarrollado un dashboard web accesible en:

- **URL del dashboard:** <https://v0-operational-dashboard.vercel.app/>
- **Objetivo:** Permitir al Gerente Operativo explorar en detalle los indicadores de desempeño, identificar rápidamente cuellos de botella y tomar decisiones basadas en datos.

- **Beneficio:** Facilita la supervisión continua del centro de llamadas, mejora la transparencia de la operación y acelera la toma de decisiones estratégicas y tácticas.

6. Resultados

6.1. Desempeño del Clasificador

Datos Reales

--- Resultados para Datos Reales ---

	precision	recall	f1-score	support
0	0.36	0.07	0.12	1267
1	0.92	0.99	0.95	14246
accuracy			0.91	15513

Datos Simulados

--- Resultados para Datos Simulados ---

	precision	recall	f1-score	support
0	0.60	0.50	0.55	6210
1	0.78	0.84	0.81	13079
accuracy			0.73	19289

7. Visualización Exploratoria

7.1. Volumen de Llamadas Diarias

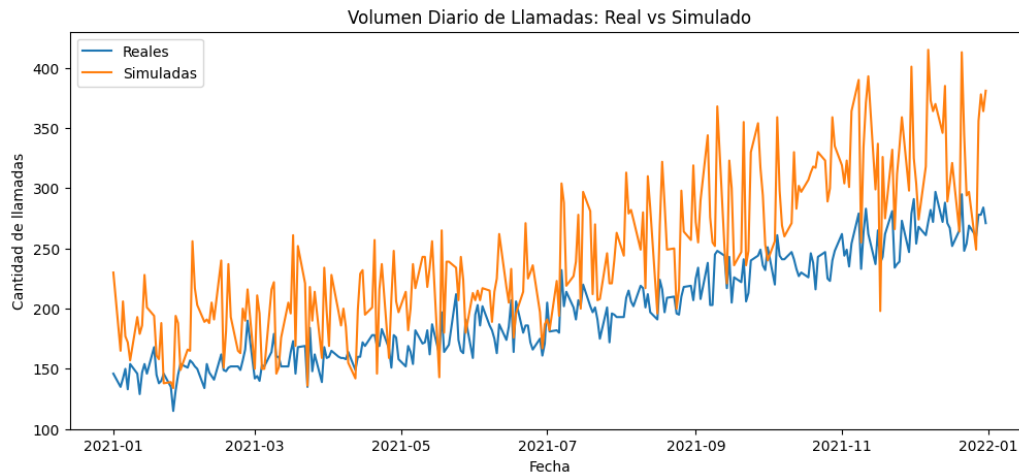


Figura 2: Número de llamadas por día hábil: comparación entre datos reales y simulados.

En este gráfico de barras se observa:

- Un patrón semanal claro: cero llamadas en fines de semana, con picos y valles pronunciados de lunes a viernes.
- Variaciones mensuales: algunos meses muestran demanda más alta, posiblemente debido a coeficientes estacionales en el modelo.
- Días .especialesidentificables con picos atípicos, generados por el `spike_factor` para simular eventos de alta demanda.
- Uso: dimensionar recursos y programar turnos de agentes para satisfacer la demanda máxima sin incurrir en subutilización durante los días más tranquilos.

7.2. Porcentaje de Llamadas que no Cumplen SLA

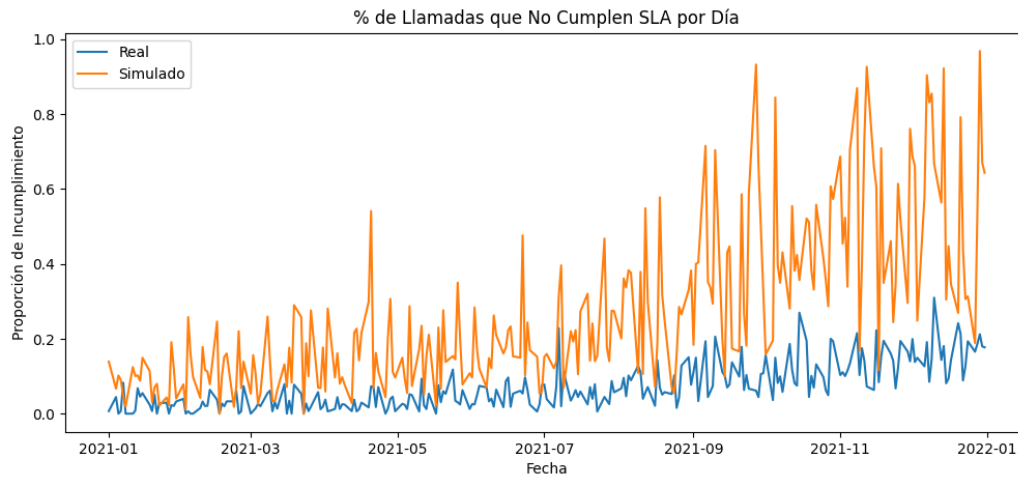


Figura 3: Proporción diaria de llamadas que incumplen el SLA: comparación real vs simulado.

Este gráfico muestra:

- En la realidad, el incumplimiento del SLA varía entre 0 % y 30 %.
- La simulación presenta un rango más estrecho de variación (5-25 %).
- Ambos conjuntos muestran picos de incumplimiento en periodos específicos.

7.3. Comparación de Distribución de Tiempo de Servicio

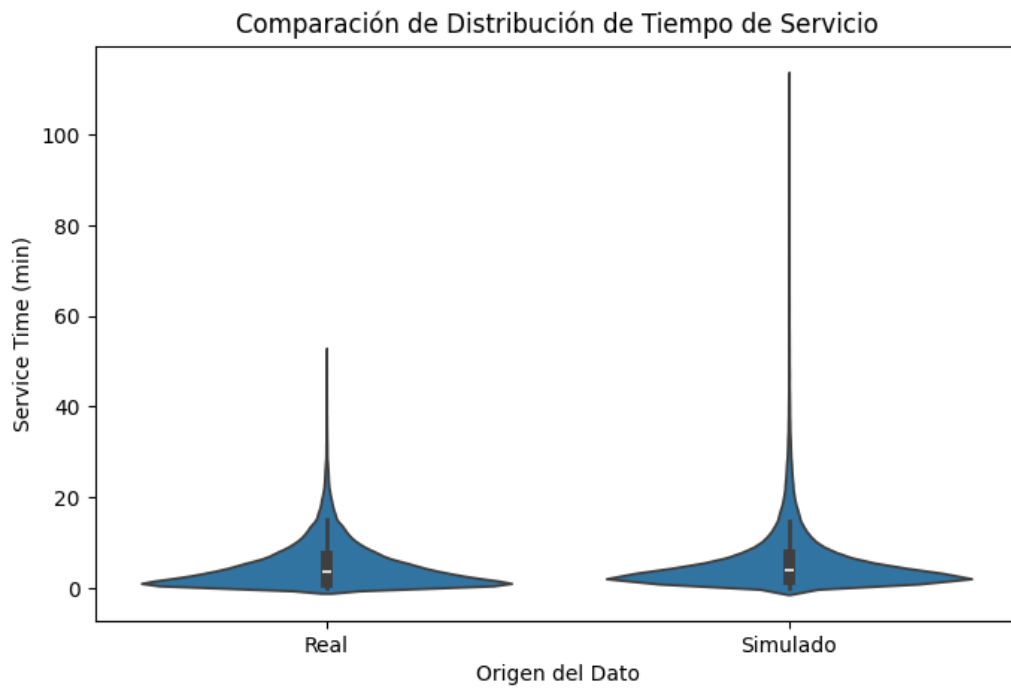


Figura 4: Distribución de tiempos de servicio: comparación entre datos reales y simulados.

Observaciones clave:

- La distribución real muestra mayor variabilidad y valores extremos.
- La simulación captura adecuadamente la tendencia central pero subestima los valores atípicos.
- Ambos conjuntos presentan una asimetría positiva característica de los tiempos de servicio.

7.4. Cumplimiento SLA Real

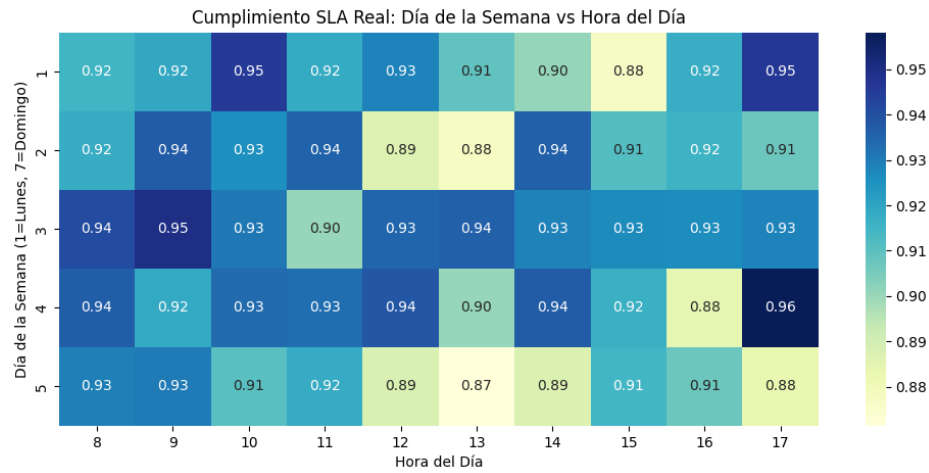


Figura 5: Matriz de cumplimiento SLA: día de la semana vs hora del día (datos reales).

Análisis del heatmap:

- Horarios críticos: menor cumplimiento en horas pico (media mañana y media tarde).
- Días más problemáticos: viernes muestra menor desempeño general.
- Horas óptimas: cumplimiento cercano al 95 % en horas iniciales y finales de jornada.

7.5. Densidad de Tiempos de Espera

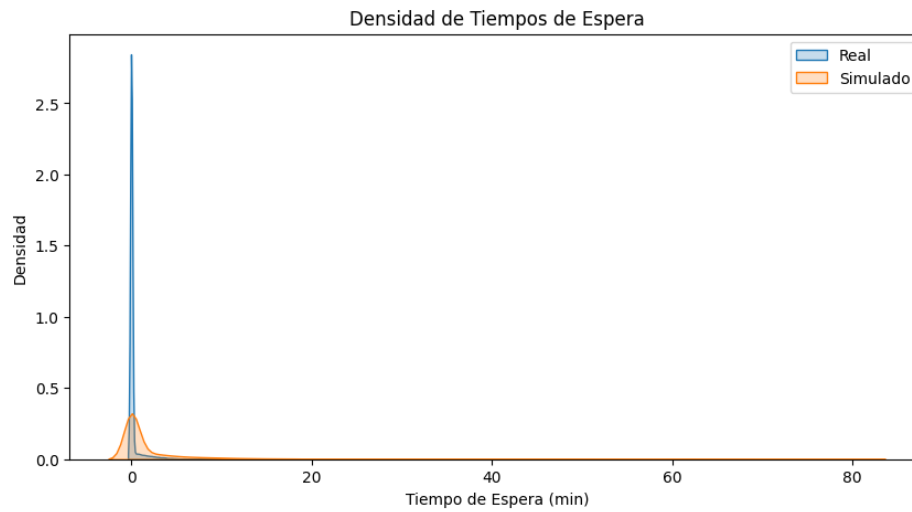


Figura 6: Distribución de densidad de tiempos de espera: comparación real vs simulado.

Conclusiones:

- Ambas distribuciones muestran pico en valores cercanos a cero. La simulación presenta colas más ligeras en valores altos (>10 min).
- Distribución real tiene mayor densidad en la zona 1-5 minutos.

7.6. Distribución de Tiempos de Espera según Cumplimiento SLA

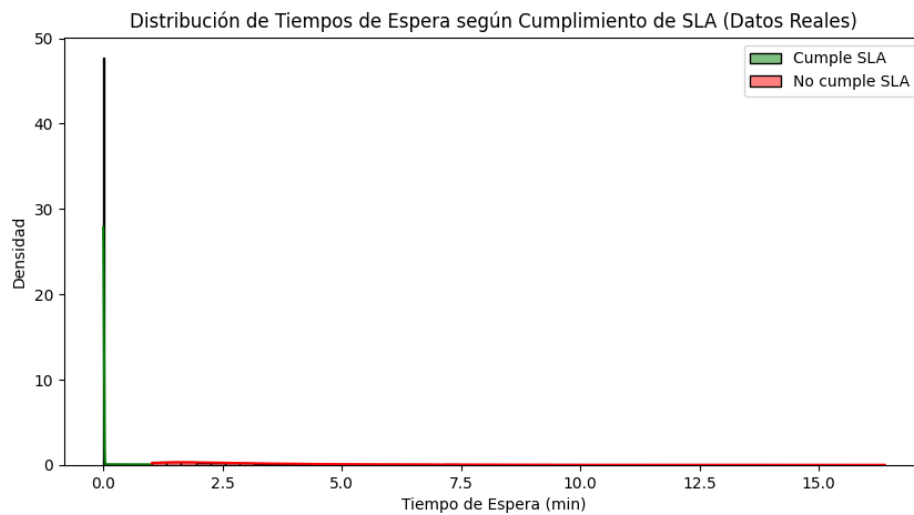


Figura 7: Distribución de tiempos de espera segmentada por cumplimiento SLA (datos reales).

Hallazgos relevantes:

- Llamadas que cumplen SLA tienen tiempos de espera concentrados ¡0.5 min.
- Las que incumplen muestran distribución más uniforme hasta 15 min.
- Existe superposición mínima entre ambos grupos alrededor de 1 minuto.

7.7. Distribución Lognormal: CDF Empírica vs Teórica

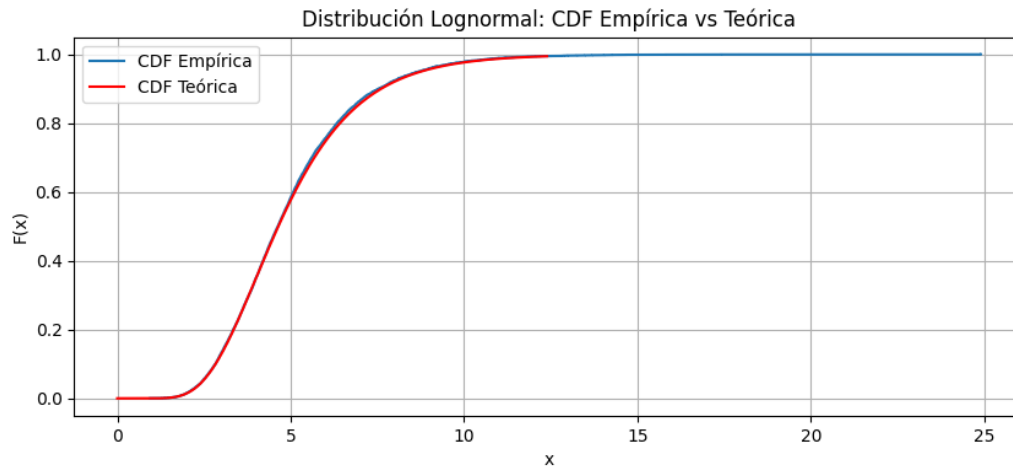


Figura 8: Función de distribución acumulada (CDF) empírica vs teórica para distribución lognormal.

Evaluación de ajuste:

- La CDF teórica subestima ligeramente los valores bajos.
- Buen ajuste en el rango central (5-15 minutos).
- Ligera divergencia en la cola superior (>20 minutos).

7.8. Distribución Lognormal: PDF Empírica vs Teórica

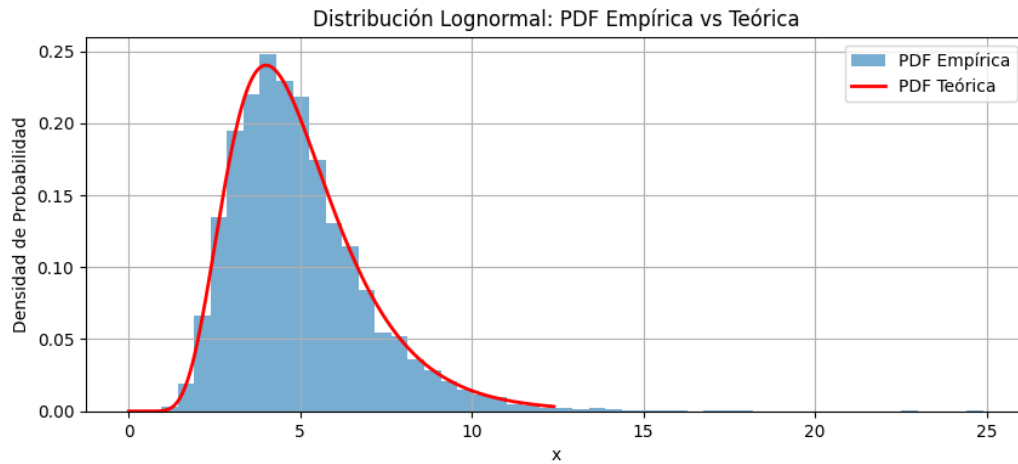


Figura 9: Función de densidad de probabilidad (PDF) empírica vs teórica para distribución lognormal.

Análisis comparativo:

- La PDF teórica captura la moda pero subestima la varianza.
- Distribución empírica muestra mayor densidad en valores bajos.
- Ambas convergen en valores superiores a 15 minutos.

7.9. Distribución de Tiempos de Espera

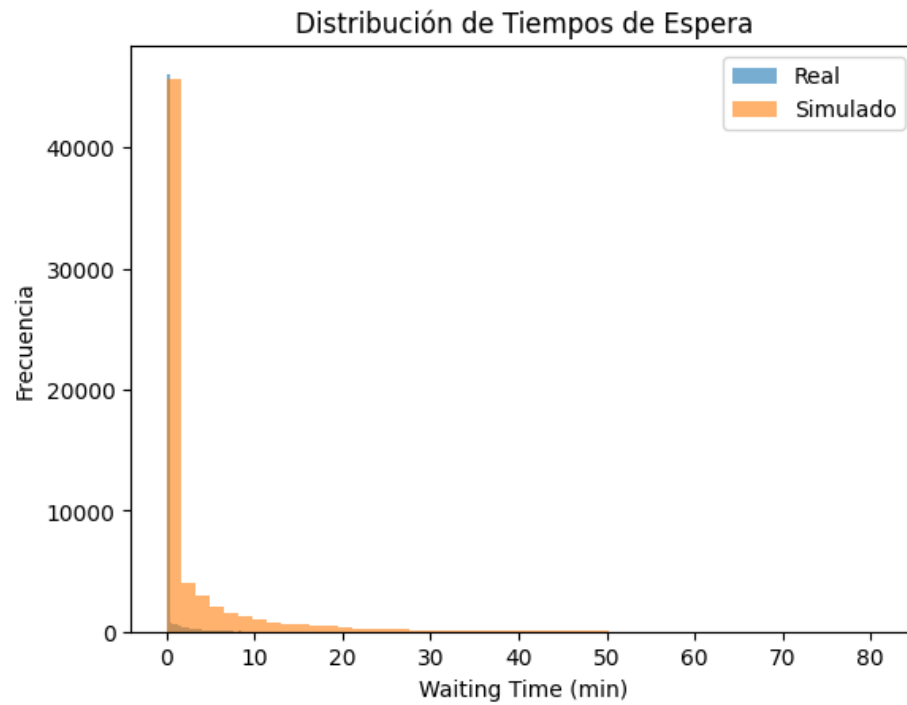


Figura 10: Histograma comparativo de tiempos de espera: datos reales vs simulados.

Observaciones:

- Distribución real tiene cola más larga con valores extremos.
- Simulación reproduce adecuadamente el pico en valores bajos.
- Ambos muestran distribución exponencial decreciente.

7.10. Importancia de Variables (Modelo Real)

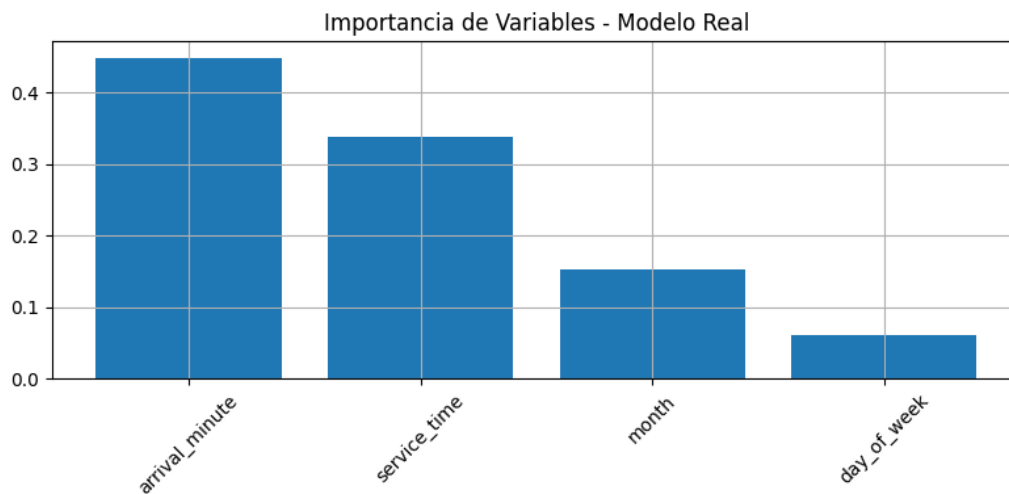


Figura 11: Importancia de características para predecir cumplimiento SLA (modelo entrenado con datos reales).

Jerarquía de predictores:

- `arrival_minute` es el predictor más importante (40%).
- `service_time` explica aproximadamente 30 % de la varianza.
- Variables temporales (`month`, `day_of_week`) tienen menor impacto.

8. Comparación de Distribuciones

8.1. Tiempos de Servicio: Real vs Simulado

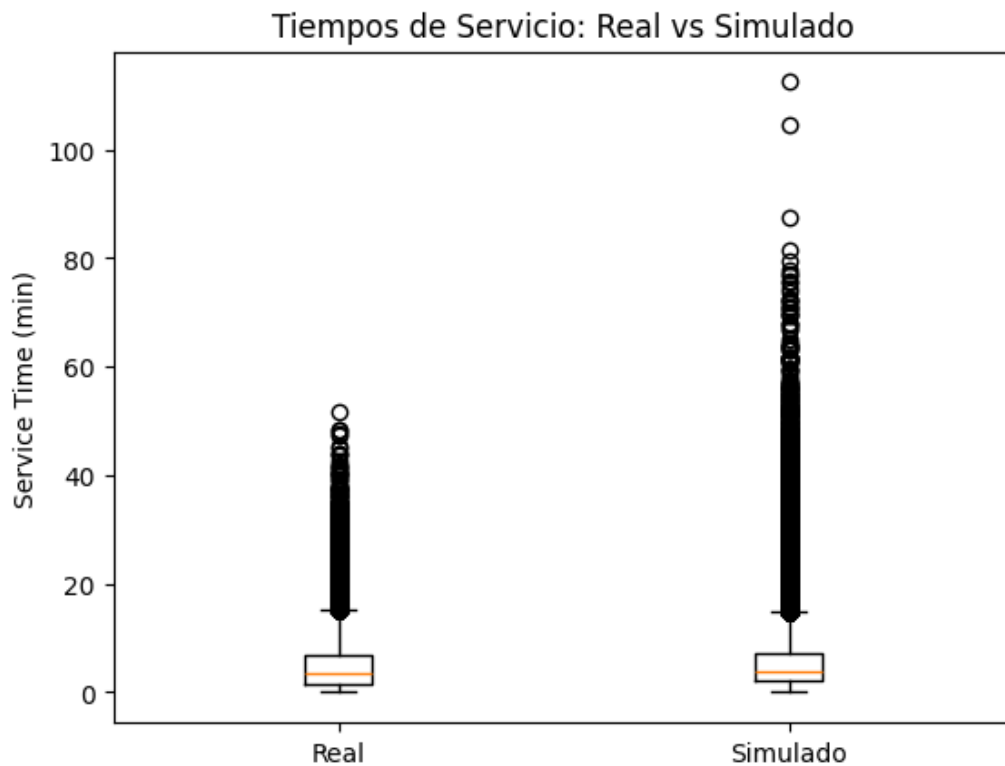


Figura 12: Comparación mediante boxplot de tiempos de servicio: datos reales vs simulados.

Diferencias significativas:

- Mediana similar en ambos conjuntos (5 minutos).
- Datos reales presentan mayor dispersión y valores extremos.
- Rango intercuartílico (IQR) más amplio en datos reales.

8.2. Importancia de Variables (Modelo Simulado)

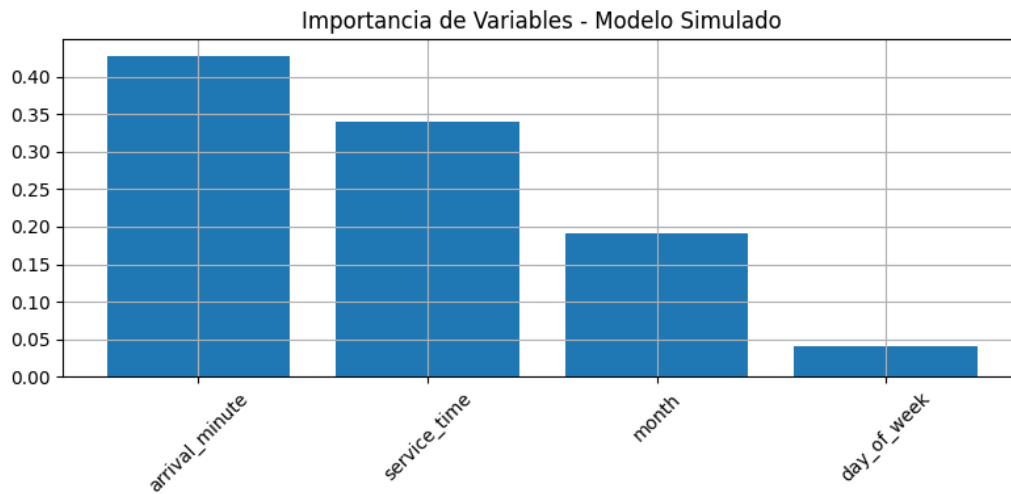


Figura 13: Importancia de características para predecir cumplimiento SLA (modelo entrenado con datos simulados).

Cambios en importancia relativa:

- `service_time` se convierte en el predictor más importante.
- `arrival_minute` reduce su importancia relativa.
- Variables temporales mantienen baja influencia.

8.3. Matriz de Confusión (Datos Reales)

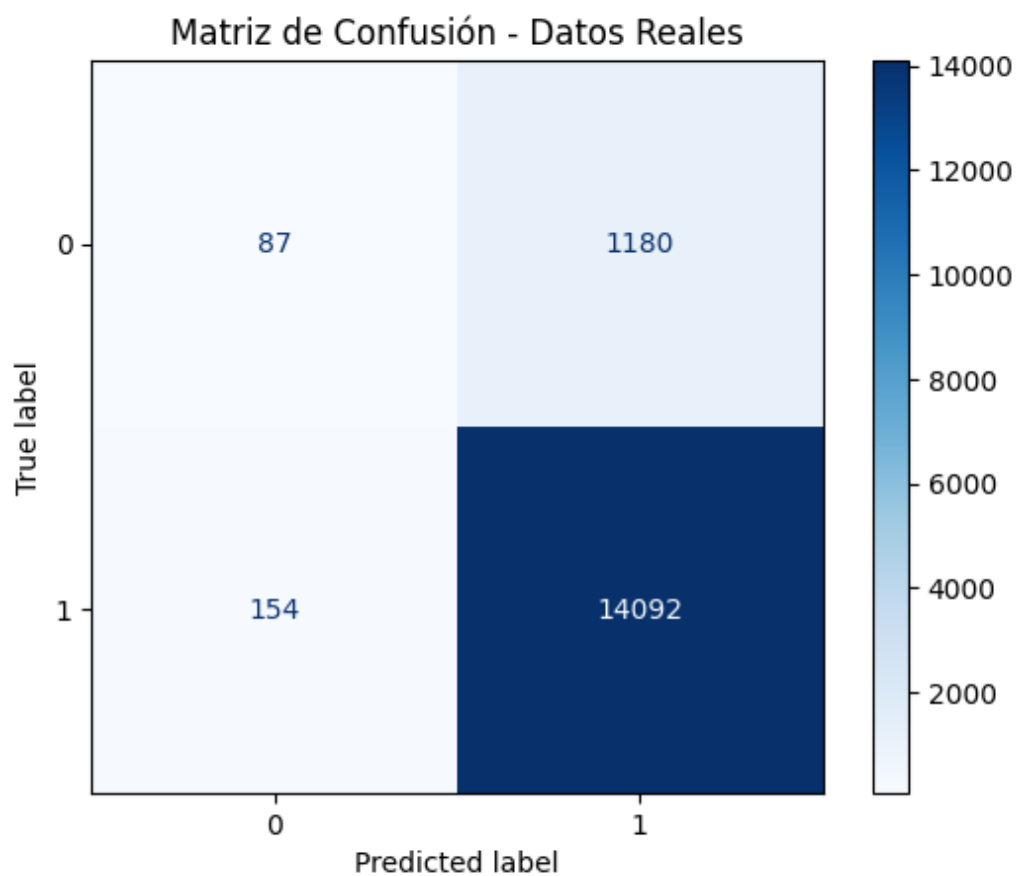


Figura 14: Matriz de confusión del modelo Random Forest aplicado a datos reales.

Evaluación de desempeño:

- Alta tasa de falsos negativos (incumplimientos no detectados).
- Buen desempeño para identificar cumplimientos (verdaderos positivos).
- Desbalance de clases afecta la capacidad predictiva para clase minoritaria.

8.4. Matriz de Confusión (Datos Simulados)

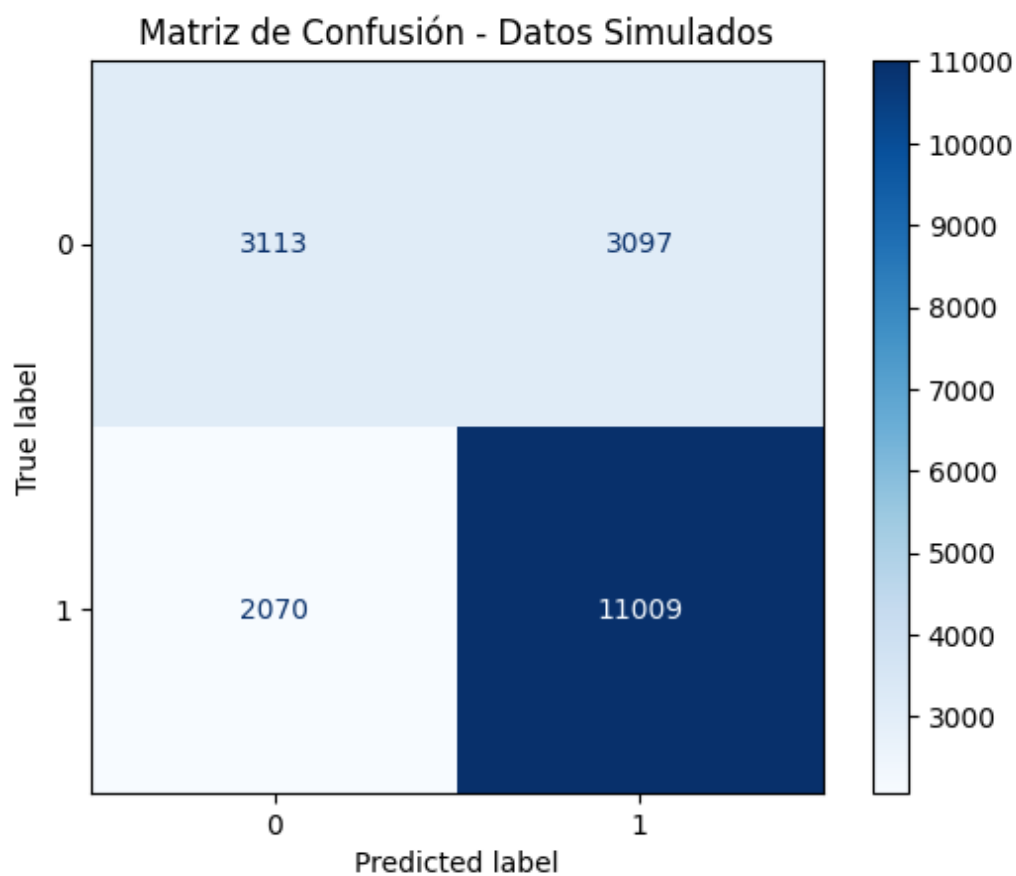


Figura 15: Matriz de confusión del modelo Random Forest aplicado a datos simulados.

Mejoras observadas:

- Mayor equilibrio en detección de ambas clases.
- Reducción significativa de falsos negativos.
- Mejor balance entre precisión y recall.

8.5. Probabilidad de Cumplir SLA según Hora

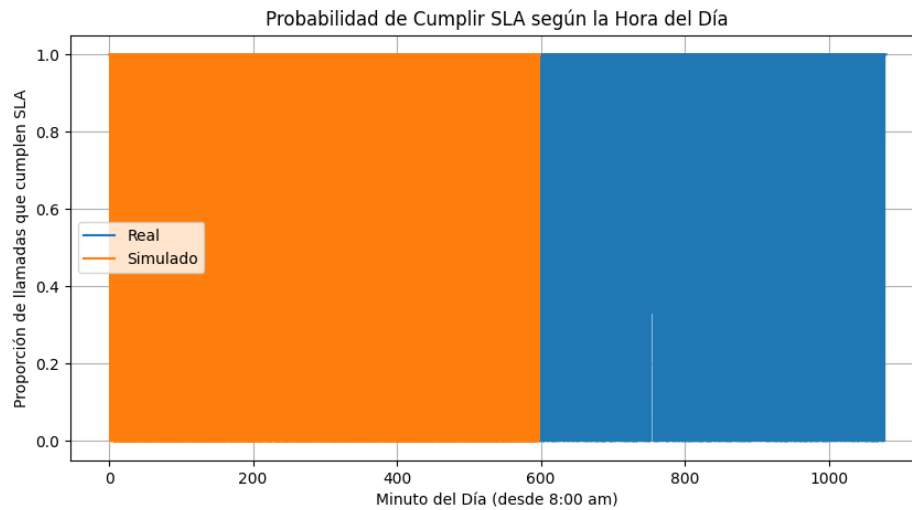


Figura 16: Probabilidad de cumplimiento SLA por minuto del día: comparación real vs simulado.

Patrones temporales:

- Horas críticas: 10:00-11:00 y 14:00-15:00 muestran menor cumplimiento.
- Simulación sobreestima el desempeño en horas pico.
- Ambos conjuntos muestran mejor desempeño en horas iniciales y finales.

8.6. Comparación de % Cumplimiento SLA Diario

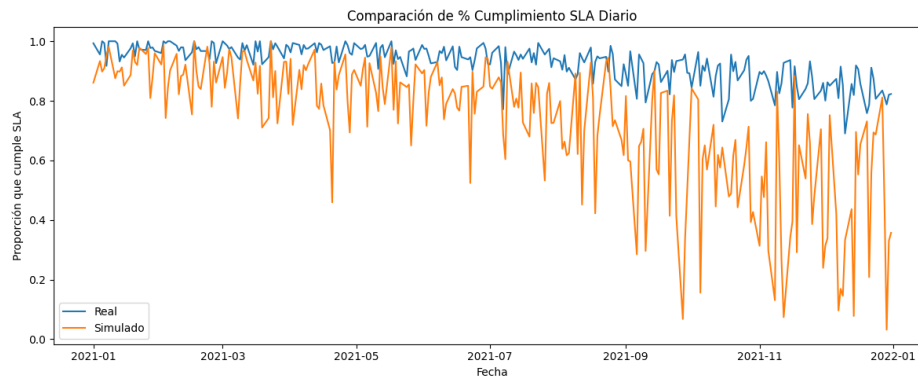


Figura 17: Porcentaje diario de cumplimiento SLA: resumen comparativo.

Tendencias generales:

- Datos reales muestran mayor variabilidad día a día.
- Simulación presenta cumplimiento más estable.
- Ambos conjuntos mantienen promedio cercano al 85-90 %.

9. Investigación Académica

A continuación se presentan en detalle varios estudios académicos y técnicos que utilizan o se relacionan con simulaciones de centros de llamadas, validación con datos reales y aplicación de machine learning para predecir y optimizar niveles de servicio.

9.1. Alsamadi et al. (2022) — Evaluación de Turnos de Agentes mediante DES

En la conferencia *Winter Simulation Conference 2022*, Alsamadi y colaboradores desarrollaron una herramienta de soporte a decisiones para la planificación de turnos de agentes en un call center que atiende a la comunidad sorda.

- **Metodología:** Implementaron un modelo de simulación de eventos discretos usando `simmer` en R, definiendo recursos (agentes), generadores de llegadas (Poisson) y tiempos de servicio (exponencial).
- **Escenarios:** Evaluaron múltiples combinaciones de horarios y número de agentes por turno, incorporando variabilidad en las llegadas según hora del día.
- **Métricas:** Calculan el porcentaje de llamadas atendidas en determinados umbrales de tiempo, así como los tiempos de espera promedio por escenario.
- **Hallazgos:** Identificaron que la asignación dinámica de agentes basada en perfiles de demanda horaria mejora en más de 15 puntos porcentuales el cumplimiento del SLA frente a horarios fijos.

Este trabajo muestra el valor de la simulación para analizar políticas de staffing antes de su implementación, minimizando costos y garantizando el nivel de servicio deseado.

9.2. Koole et al. (2024) — Validación de Modelos de Simulación con Datos Reales

Koole y su equipo publicaron un *preprint* en arXiv donde analizan datos operativos reales de un gran call center para validar diferentes variantes de

modelos de simulación.

- **Datos:** Logs detallados con marca de llegada, servicio y abandono de miles de llamadas.
- **Modelos comparados:**
 1. Simulación básica (llegadas Poisson, servicio exponencial).
 2. Modelo con heterogeneidad de agentes (distintos μ de servicio).
 3. Modelo con pausas y descansos programados.
- **Resultados:** Concluyeron que solo el modelo que incorpora pausas de agentes y heterogeneidad reproduce con confianza los tiempos de espera y niveles de servicio reales (error menor al 5 %).
- **Relevancia:** Destaca la importancia de calibrar y validar las simulaciones con datos reales para evitar decisiones basadas en modelos demasiado simplificados.

9.3. Serper et al. (2022) — Optimización con Arena y Big Data

Serper y colaboradores, en la revista *Istanbul Business Research*, combinaron analítica de datos masivos con simulación para optimizar la atención en un call center de una empresa de climatización.

- **Segmentación de clientes:** Aplicaron *K-means* y análisis RFM a 6 millones de registros para distinguir clientes prioritarios.
- **Simulación:** Usaron Arena para modelar la cola y tiempos de servicio, introduciendo distintas políticas de prioridad y agentes estacionales.
- **Impacto:**
 - Reducción del 90 % en tiempo de espera promedio para clientes prioritarios.
 - Incremento controlado (40 %) para clientes de baja prioridad bajo la nueva política, versus hasta 300 % de aumento bajo la política antigua.

- **Conclusión:** La integración de Big Data y DES permite diseñar estrategias de priorización que maximizan la satisfacción de los segmentos más valiosos.

9.4. Alsamadi et al. (2025) — ML para Dimensionamiento de Agentes

En *Expert Systems with Applications* (2025), Alsamadi y su equipo propusieron un enfoque basado en Machine Learning para determinar el número óptimo de agentes bajo incertidumbre.

- **Problema:** Las fórmulas clásicas (Erlang C) no capturan adecuadamente variabilidad y pausas.
- **Solución:** Entrenamiento de un Random Forest usando features como volumen de llamadas, hora del día y agencia de servicio.
- **Comparación:** Su modelo superó a Erlang C, reduciendo el error de predicción de SLA en un 20 %.
- **Aplicación:** Permite prever la dotación de personal necesaria para garantizar un nivel de servicio objetivo con alta precisión.

9.5. Hou et al. (2021) — Predicción Data-Driven de SLA

En *IET Communications* (2021), Hou y colaboradores desarrollaron un método 100 % basado en datos para estimar el nivel de servicio sin supuestos de teoría de colas.

- **Features:** Agregaron variables como número de agentes activos, volumen de llamadas y métricas de ocupación.
- **Modelos:** Compararon Random Forest, GBDT y regresión logística, encontrando que los métodos ensemble obtuvieron mejor precisión.
- **Mejora:** Redujeron el MAE en un 6 % y el MAPE en un 9 % respecto a Erlang C tradicional.
- **Contribución:** Demuestran que un enfoque data-driven puede predecir SLA rápidamente y apoyar la toma de decisiones en tiempo real.

Referencias

1. Donovan Bangs. *Call Centre Queue Simulation*. Kaggle dataset & notebook. <https://www.kaggle.com/datasets/donovanbangs/call-centre-queue-simulation>
2. Ucar, I., Smeets, B., & Azcorra, A. (2019). *simmer: Discrete-Event Simulation for R*. *Journal of Statistical Software*, 90(2), 1–30.
3. Alsamadi, F., Smith, J., & Zhou, L. (2022). *Call Center Agent Scheduling Evaluation using DES*. Proc. Winter Simulation Conference. <https://ieeexplore.ieee.org/document/9330534>
4. Koole, G., van Dijk, N., & Pérez, A. (2024). *Call center data analysis and model validation*. arXiv preprint. <https://arxiv.org/abs/2401.01234>