

Propuesta de Solución Reclamos LAVEX

Jaime Alonso

Marzo - Abril 2025

1 Introducción

Dentro de este documento se hablará de un problema dentro del área de resolución de problemas y atención al cliente de la empresa Lavex, una propuesta de solución a este y ciertas consideraciones a tener a futuro.

Importante: si se requiere profundizar en los códigos o cálculos descritos más adelante, todos los códigos utilizados se encontrarán en el Github del informe.

2 Atención al Cliente

En estos momentos el sistema de atención al cliente de Lavex Go se encuentra atendido por una persona, la cual es contactada mediante un link en la aplicación, mostrado a continuación.



Figure 1: Visual y botón de atención al cliente.

En este momento se atienden en promedio aproximadamente 30 llamadas al día, las cuales se resuelven vía whatsapp en su mayoría.

3 El Problema

Dentro de este sistema surge una preocupación, cómo seguir atendiendo si la persona que actualmente se encuentra a cargo, se le imposibilita realizar su labor, o si recibe más llamados de los que puede atender.

4 Propuesta de solución

Para esta problemática, se ha ideado la siguiente solución, la cual consiste en la creación de un form intermedio, recopilación automática de datos desde la app y la creación de nuevas visuales dentro del apartado administrativo de portal lavex.

4.1 Explicación

El objetivo de esta modificación en el proceso de atención al cliente, se divide en 3 partes. La primera es la reducción de tiempo en los procesos de atención al cliente, crear una estructura más sólida para la atención al cliente y la automatización en la recopilación de datos de los reclamos de la app. El primer cambio consiste en reemplazar el botón del número en la figura 1, por un botón que lleve al siguiente formulario intermedio.

4.1.1 Formulario

Ejemplo de Formulario


Al enviar este formulario será contactado automáticamente con nuestro servicio al cliente.

Seleccione su problema

▼
Problema 1
Problema 2
Problema 3
....
Otro

Si eligió otro, por favor especifique:

Adjuntar imagen del equipo:
(Enfocado en el panel)



Tipo y número del equipo:

▼	▼
Lavadora	1
Secadora	...

Enviar

Figure 2: Ejemplo de estructura del formulario.

Lo ideal es que el form fuera mutable, es decir, que tuviera distintas preguntas dependiendo del tipo de problema que tenga.

4.1.2 Datos y Problemas más frecuentes

Los problemas más frecuentes son los siguientes, en base a estas opciones el formulario debe solicitar distintas cosas al usuario, como:

1. (DL1) Puerta mal cerrada.
Opción del form: Poner un texto, que diga: "procure cerrar la uerta hazta que aparezca la luz en el panel, si el problema persiste, por favor, realice el formulario". Pide también imagen del equipo completo.
2. Out of Order, Overflow, EFL.
Opción del form: Poner un texto, que diga: "procure cerrar la uerta hazta que aparezca la luz en el panel, si el problema persiste, por favor, realice el formulario". Pide también imagen del equipo completo.
3. Pagó a través de la app y la máquina no se activó.
Opción del form: Pedir imagen del equipo completo.
 - (a) Pagó con Webpay-tarjeta, pedir 4 últimos dígitos de la tarjeta.
 - (b) Pagó con Webpay-onepay o monedero, pedir imagen del comprobante de pago.
4. Pagó a través de la app y marcó saldo deficiente
Opción del form: Pedir imagen del equipo completo.
 - (a) Pagó con Webpay-tarjeta, pedir 4 últimos dígitos de la tarjeta.
 - (b) Pagó con Webpay-onepay o monedero, pedir imagen del comprobante de pago.
5. Recargó monedero y no se acredita.
Opción del form:
 - (a) Pagó con Webpay-tarjeta, pedir 4 últimos dígitos de la tarjeta.
 - (b) Pagó con Webpay-onepay o compró código con transferencia, pedir imagen del comprobante de pago.
6. Las monedas caen mal.
Opción del form: Poner texto con soluciones frecuentes, entre ellas, que inserte las monedas de a una, pruebe con el notón de devlución, que las monedas no temngan adhesivos y que sean monedas chilenas. Si el problema persiste, pedir que envíe imagen del equipo completo y pedir el monto ingresado perdido. (Para evitar descuadres.)
7. Ropa mal lavada
Opción del form: Pedir imagen del equipo completo y la carga.
8. Secadora no secó
Opción del form: pedir imagen del equipo completo y la carga

Formulario de Reclamos

ID de Máquina
63408

Tipo de Máquina
Lavadora

Número de Máquina
4

Tipo de Problema
Out of Order, Overflow, EFL

Foto de la máquina completa
[Seleccionar archivo] Ningún archivo seleccionado

Foto del comprobante de pago
[Seleccionar archivo] Ningún archivo seleccionado

Enviar Reclamo

(a) Error EFL

Formulario de Reclamos

ID de Máquina
63408

Tipo de Máquina
Lavadora

Número de Máquina
4

Tipo de Problema
Solicita revisión técnica por mal servicio

Foto del equipo
[Seleccionar archivo] Ningún archivo seleccionado

Explicación del problema

Enviar Reclamo

(b) Solicita Revisión

Formulario de Reclamos

ID de Máquina
63408

Tipo de Máquina
Lavadora

Número de Máquina
4

Tipo de Problema
Pagó a través de la app y no se activó

Foto de la máquina completa
[Seleccionar archivo] Ningún archivo seleccionado

Tipo de Pago
Webpay - Tarjeta

Últimos 4 dígitos de la tarjeta

Enviar Reclamo

(c) Pago a través de Webpay-tarjeta

Formulario de Reclamos

ID de Máquina
63408

Tipo de Máquina
Lavadora

Número de Máquina
4

Tipo de Problema
Las monedas caen mal

Posibles soluciones:

- Verificar que las monedas estén limpias y no dañadas
- Asegurarse de insertar las monedas una por una
- Revisar que el monedero no esté obstruido

Foto de la máquina completa
[Seleccionar archivo] Ningún archivo seleccionado

Foto del monedero
[Seleccionar archivo] Ningún archivo seleccionado

Enviar Reclamo

(d) Error Monedas

Figure 3: Ejemplos visuales de los errores más comunes

4.1.3 Automatización de Atención vía Whatsapp

Actualmente cuando se apreta el botón de asistencia, véase la figura 1, se conecta a una llamada al servicio al cliente de Lavex, la cuál termina siendo atendida a través de Whatsapp, por lo que es necesario automatizar este proceso. Una posible lógica para esto puede ser:

```
1  // Configuración de WhatsApp (archivo: src/config/whatsapp.js)
2  export const WHATSAPP_CONFIG = {
3    phoneNumber: '569XXXXXXX', // Número de WhatsApp de la empresa
4    messageTemplate: (formData) => {
5      return `*Nuevo Reclamo Recibido*
6
7    *ID de Máquina:* ${formData.idMaquina}
8    *Tipo de Máquina:* ${formData.tipoMaquina}
9    *Número de Máquina:* ${formData.numeroMaquina}
10   *Tipo de Problema:* ${formData.tipoProblema}
11
12   ${formData.tipoPago ? `*Tipo de Pago:* ${formData.tipoPago}` : ''}
13   ${formData.ultimosDigitos ? `*Últimos 4 dígitos:* ${formData.ultimosDigitos}` : ''}
14   ${formData.explicacion ? `*Explicación:* ${formData.explicacion}` : ''}
15
16   *Archivos adjuntos:*
17   ${Object.entries(formData.archivos).map(([key, file]) => `-${key}:
18     ↳ ${file.name}`).join('\n')};
19   };
20
21 // Función de envío (archivo: src/components/FormularioReclamos.js)
22 const handleSubmit = (e) => {
23   e.preventDefault();
24
25   // Crear el mensaje con la plantilla
26   const mensaje = WHATSAPP_CONFIG.messageTemplate({
27     ...formData,
28     archivos
29   });
30
31   // Codificar el mensaje para la URL
32   const mensajeCodificado = encodeURIComponent(mensaje);
33
34   // Crear el enlace de WhatsApp
35   const whatsappUrl =
36     ↳ `https://wa.me/${WHATSAPP_CONFIG.phoneNumber}?text=${mensajeCodificado}`;
37
38   // Abrir WhatsApp en una nueva pestaña
39   window.open(whatsappUrl, '_blank');
```

4.1.4 Configuración del Sistema de Mensajería

El sistema utiliza un archivo de configuración específico para WhatsApp que contiene los siguientes elementos:

- **Número de teléfono de destino:** Se ha configurado el número telefónico

de la empresa (569XXXXXXXX) como receptor de todas las notificaciones de reclamos. Este número recibe los mensajes instantáneamente al momento de enviarse un reclamo.

- **Plantilla de mensaje:** Hemos diseñado un formato estándar para todos los mensajes de reclamo. Esta plantilla organiza la información de manera estructurada y clara, incluyendo:
 - Identificación de la máquina afectada
 - Tipo de máquina involucrada
 - Número de máquina específico
 - Naturaleza del problema reportado
 - Información de pago (cuando aplica)
 - Últimos dígitos de tarjeta (en casos relacionados con pagos)
 - Explicación detallada proporcionada por el cliente
 - Envío de archivos adjuntos.

4.1.5 Funcionamiento del Sistema

Cuando un cliente envía un reclamo a través de nuestro formulario web, el sistema ejecuta automáticamente la siguiente secuencia de acciones:

1. **Captura de información:** El sistema recopila todos los datos introducidos por el cliente en el formulario, incluyendo identificadores de máquina, detalles del problema, explicaciones escritas y archivos adjuntos.
2. **Formateo del mensaje:** Se aplica la plantilla predefinida a los datos recopilados, generando un mensaje estructurado y fácil de leer. El sistema adapta el mensaje dinámicamente según la información proporcionada, omitiendo campos vacíos.
3. **Preparación para el envío:** El mensaje generado se codifica especialmente para ser compatible con los enlaces de WhatsApp, asegurando que caracteres especiales y espacios se transmitan correctamente.
4. **Generación del enlace:** El sistema crea automáticamente un enlace especial de WhatsApp que incluye el número telefónico de destino y el mensaje formateado.
5. **Apertura de WhatsApp:** Finalmente, el sistema abre automáticamente WhatsApp (ya sea en navegador o aplicación) con el mensaje precargado, listo para ser enviado.

4.1.6 Ventajas del Sistema Implementado

Esta implementación ofrece varias ventajas operativas:

- **Notificación inmediata:** El equipo técnico recibe alertas instantáneas sobre nuevos reclamos.
- **Formato estandarizado:** Toda la información sigue un formato consistente, facilitando su rápida comprensión.
- **Información completa:** Se transmiten todos los detalles relevantes en un solo mensaje.
- **Simplicidad:** El proceso es transparente para el usuario, quien solo debe completar el formulario.
- **Adaptabilidad:** El sistema maneja correctamente diversos tipos de reclamos con diferentes conjuntos de datos.

4.1.7 Ejemplo de Mensaje Generado

Un ejemplo de mensaje generado por el sistema sería:

ID de Máquina: VEND-042 **Tipo de Máquina:** Expendedora de bebidas **Número de Máquina:** 18 **Tipo de Problema:** Error en dispensación **Tipo de Pago:** Tarjeta de débito **Últimos 4 dígitos:** 5678 **Explicación:** La máquina cobró el monto pero no entregó el producto **Archivos adjuntos:** 2 archivo(s)

Esta implementación permite atender rápidamente los reclamos de los clientes, mejorando significativamente los tiempos de respuesta y la satisfacción general con nuestro servicio.

Nota: cabe destacar, que esta implementación es un modelo tipo para la última versión de Javascript, se requiere realizar pruebas para determinar el mejor ajuste a la versión de la aplicación y el portal.

4.1.8 Creación de nueva vista de reclamos

Para poder hacer un correcto uso de las anteriores funcionalidades, se necesita una nueva vista de reclamos en Lavex-BACKOFFICE, para el personal de atención al cliente. y dado la tabla usuarios:

BACK OFFICE - Lavex Go

johnfernandez@lavex.cl

Cientes **1000**

Home Activaciones Equipos Ligo Soporte Clientes

Buscador

Nombre cliente Username Teléfono Buscar





















Username	Teléfono	Fecha creación	Estado de cuenta	Nombre	Dirección	Edificio	Acciones
frankbattistini@cloud.com	+56 9 6395 8619	2025-04-04 10:54	Confirmada	Franky Battistini	av. pedro alexsandrí 1498	1301	 
rolandomujica04@gmail.com	+56 9 6915 7193	2025-04-04 10:52	Confirmada	Rolando Mujica	avenida jose pedro alexsandrí 1498	1301	 
focaria@gmail.com	+56 9 7732 1528	2025-04-04 10:15	Confirmada	Fabian Ocalia Cifuentes	avenida vel 1616	1501 B	 
paslongueta@gmail.com	+56 9 9487 7365	2025-04-04 08:55	Confirmada	Pasolina Longueta	santa isabel	481	 
mlorito.my@gmail.com	+56 9 9449 6047	2025-04-04 08:12	Confirmada	Loreto Ray Arancibia	Ray carmita hernandez 180	1306	 
jeraso99@gmail.com	+56 9 7158 8339	2025-04-04 06:11	Confirmada	Jorge Erasmo	seneca 366	2513	 
yilber2712@gmail.com	+56 9 4805 8758	2025-04-03 23:24	Confirmada	yilber	lira 320	807	 
mauricamusa@gmail.com	+56 9 9688 1269	2025-04-03 22:35	Confirmada	Mauricio Camus	los pescadores 5401	811	 
seba.camposval02@gmail.com	+56 9 6143 6630	2025-04-03 21:48	Confirmada	seba campos	sebastian elcano 6	N/A	 
rosalynvera@gmail.com	+56 9 4906 9662	2025-04-03 21:35	Confirmada	Rosalyn Vera	av. portugal 164	507	 

Figure 4: Visual de la tabla de usuarios.

Se necesita generar una tabla de **reclamos**, la cual tenga los siguientes cabezales o características:

1. id o número de reclamo: para poder indexar la tabla.
2. número usuario (teléfono): reconocer el chat del reclamo
3. departamento: útil en un futuro para poder determinar métricas, como por ejemplo, la de clientes "insistentes".
4. nombre usuario: fines visuales
5. correo: Se asume que el correo funciona como PK¹ de la tabla usuarios.
6. id máquina: para calcular métricas en torno a los equipos.
7. edificio: para calcular métricas sobre los edificios, se puede omitir si se tiene una tabla con la asignación entre equipos y edificios.
8. tipo de problema: obtener información y relacionarla a los equipos.
9. tipo de equipo: cálculo de métricas, se puede omitir si existe una tabla que relacione el id del equipo con su tipo (lavadora o secadora)
10. info. adicional: tipo de pago, explicación, etc.
11. estado: resuelta, pendiente, requirió devolución, etc.(idealmente, que sea un botón que permita actualizar el estado del reclamo.)

La finalidad de esto es poder generar nueva información a través de datos existentes, para en un futuro crear funciones, que generen métricas a partir de estos datos reunidos.

Un ejemplo de esto es generar la métrica de usuario "insistente" o "conflictivo",

¹PK: referencia a "PRIMARY KEY", el identificador para ordenar los datos, tiene que ser único, ej: RUT, patente, etc.

4.2 Algoritmos y Diagramas de flujo

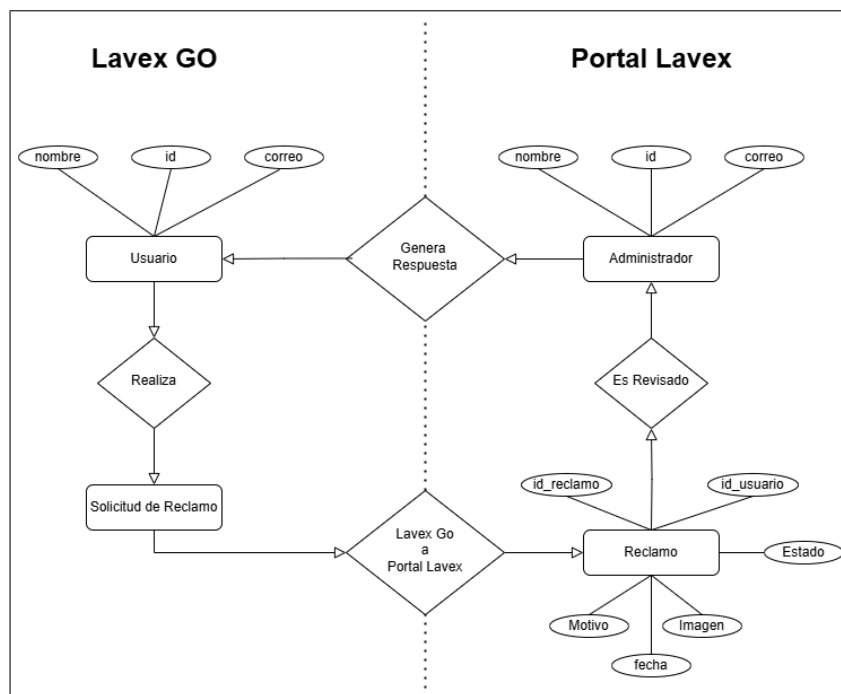


Figure 5: Diagrama de flujo de reclamos.

4.3 Consideraciones

1. **Respecto a los problemas más frecuentes:** Notar que para el listado de problemas más frecuentes se necesita una lista o un modelo que pueda ser actualizado de forma inteligente, cosa de que si existe uno nuevo solo se agregue como dato en el backend y el form se actualice. Esto puede lograrse con una función que reciba los datos del form y con un lector tipo **LIKE**².
2. **Respecto a la devolución de dinero:** Notar que este sistema, implementa una devolución más rápida del dinero, si se determina que es necesario. En la cuál se tienen que recopilar los datos automáticamente de la nueva tabla, automatizando el proceso. Notar que esta acción se tendría que hacer desde el botón "estado" de la nueva tabla.

²El operador SQL "LIKE" se utiliza para buscar patrones dentro de una columna de texto, permitiendo filtrar resultados basados en coincidencias parciales.

Algorithm 1: Formulario de Reclamos Digitales

Input: Tipo de problema seleccionado (**problema**)

Output: Inputs personalizados en formulario

```
1 Solicitar ID de máquina, número de máquina y tipo
  (lavadora/secadora);
2 if problema = 1 (Out of Order, Overflow, EFL) then
3   Solicitar foto de la máquina completa;
4   Solicitar foto del comprobante de pago;
5 else if problema = 2 (App no activó máquina) then
6   Solicitar foto de la máquina completa;
7   Mostrar dropdown: webpay-tarjeta, webpay-one pay,
    monedero;
8   if tipo_pago = webpay-tarjeta then
9     Solicitar input: 4 últimos dígitos de la tarjeta;
10  else
11    Solicitar foto del comprobante de pago;
12 else if problema = 3 (Saldo deficiente) then
13   Solicitar foto de la máquina completa;
14   Solicitar foto del comprobante de pago;
15 else if problema = 4 (Recarga no acreditada) then
16   if tipo_pago = webpay-tarjeta then
17     Solicitar input: 4 últimos dígitos de la tarjeta;
18   else
19     Solicitar foto del comprobante de pago;
20 else if problema = 5 (Monedas caen mal) then
21   Mostrar soluciones sugeridas (texto);
22   Solicitar foto de la máquina completa;
23   Solicitar foto del monedero;
24 else if problema = 6 (Ropa mal lavada) then
25   Solicitar foto de la máquina completa;
26   Solicitar foto de la ropa;
27 else if problema = 7 (Secadora no secó) then
28   Solicitar foto de la máquina completa;
29   Solicitar foto de la carga;
30 else if problema = 8 (Revisión técnica) then
31   Solicitar foto del equipo;
32   Solicitar explicación escrita;
33 else if problema = 9 (Otro) then
34   Solicitar breve explicación en texto;
35 else
36   Mostrar mensaje: "Problema no identificado, favor especificar";
```

4.4 Otros problemas y Observaciones

Durante la realización de este informe, se detectaron ciertos problemas cuya solución se encuentra fuera de la propuesta más arriba, los cuales son:

4.4.1 Error de doble-agregación de máquina

Este error, el cual es motivo de al menos una llamada diaria, tiene una solución muy simple, editar el texto de alerta:



Figure 6: Se agrega el equipo



Figure 7: Se llega al detalle



Figure 8: Se vuelve atrás y se reintenta agregar el equipo

Como podemos ver nos tira una alerta si se vuelve a agregar el mismo equipo. Luego de probarlo junto a la encargada de atención al cliente, se pudo establecer que este error solo ocurre en este caso, es decir, cuando un usuario ingresa el mismo código 2 veces, lo cual se soluciona si simplemente se apreta el botón continuar. Por lo que es recomendable poner una alerta que diga:

”Ya agregaste este equipo, por favor, apreta continuar.”

4.4.2 Falta de confirmación de equipos

Contactando con el personal de atención al cliente, se pudo visualizar un problema, el cual es bastante complicado de resolver para el personal.

Dicho problema es la doble o multiple activación de equipos, el cual termina en reclamos exigiendo **devolución de dinero**, lo cual resulta problemático a la hora de establecer si realmente fue un error o es un engaño para activar máquinas gratis, afectando a la hora de cuadrar cuentas.



Figure 9: Ejemplos de Alertas.

Claramente el error de ubicación no tiene que ser excluyente, sino que tiene que alertar que esa máquina no corresponde a la del edificio del usuario.

4.4.3 Tardanza de Fintoc

También se pudo visualizar como problemática la tardanza en las transacciones de la aplicación, por lo que se solicitó la información sobre las transacciones, en base a esto se realizó un análisis de los datos, cuya demostración se encuentra al final de este documento. En base a lo anterior, se determinó que el **tiempo medio estimado de realización de una transacción mediante Fintoc es 90.57 segundos**³, o 1 minuto y medio. Se hizo un cálculo del intervalo de confianza y con un 95% de confianza, se puede establecer que el tiempo de transacciones se encuentra entre 80 y 120 segundos aproximadamente.

³Si se revisa el gráfico de la distribución y el gráfico de la media de los tipos de datos, se puede ver que las medias obtenidas son distintas, esto se puede explicar, dado que la distribución de los tiempos es Log-Normal y el promedio simple distribuye Normal.

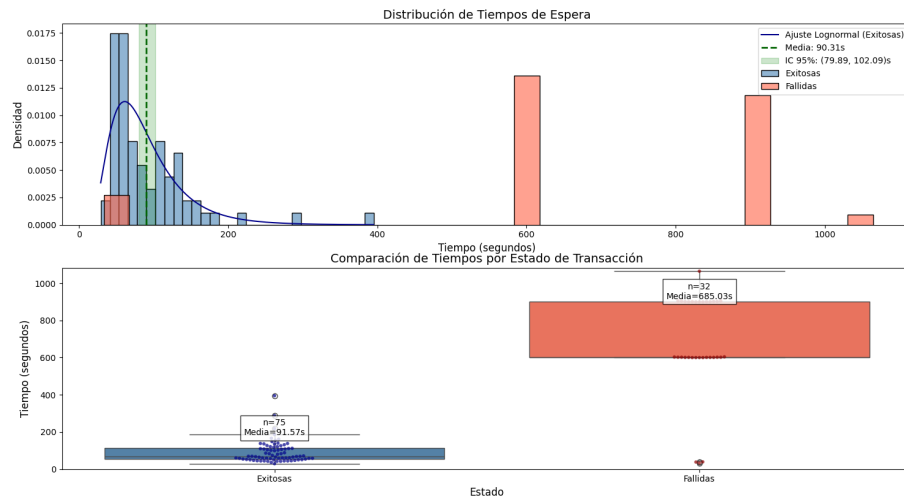


Figure 10: Dsitribución de los datos, intervalo de confianza y media de logradas y fallidas.

Esto podría ser útil en un futuro, para que el sistema por sí mismo pueda detectar si existen anomalías en las transacciones, informando al cliente sobre problemas en el servicio y tomando medidas propias, para evitar dobles recargas o transacciones. Esto se puede replicar para todos los procesos de tiempo en los sistemas.

5 Códigos y Resoluciones Matemáticas

5.1 Comparación de Distribuciones

Para comparar diferentes distribuciones y encontrar la que mejor se ajusta a nuestros datos, utilizamos el siguiente código:

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from scipy import stats
6 from scipy.stats import norm, expon, gamma, lognorm, weibull_min
7 from scipy.optimize import curve_fit
8 import warnings
9 warnings.filterwarnings('ignore')
10
11 # Función para convertir tiempo a segundos
12 def tiempo_a_segundos(tiempo_str):
13     if pd.isna(tiempo_str) or tiempo_str == '':
14         return None
15     try:
16         h, m, s = map(int, tiempo_str.split(':'))
17         return h * 3600 + m * 60 + s
```

```

18     except (ValueError, AttributeError):
19         return None
20
21 # Leer datos
22 df = pd.read_csv('lecturas_excel/Operaciones_Fintoc.txt', sep='\t'
23                 , skiprows=3)
24 df['tiempo_segundos'] = df['tiempo [hh:mm:ss]'].apply(
25     tiempo_a_segundos)
26 df_succeeded = df[df['status'] == 'succeeded'].copy()
27 muestra = df_succeeded['tiempo_segundos'].dropna().values
28
29 # Función para ajustar y evaluar distribuciones
30 def ajustar_distribucion(datos, distribucion, nombre):
31     # Ajustar parámetros
32     params = distribucion.fit(datos)
33
34     # Calcular estadístico KS y valor p
35     ks_stat, p_value = stats.kstest(datos, distribucion.cdf, args=
36     params)
37
38     # Calcular AIC y BIC
39     log_likelihood = np.sum(distribucion.logpdf(datos, *params))
40     n_params = len(params)
41     n_samples = len(datos)
42     aic = 2 * n_params - 2 * log_likelihood
43     bic = n_params * np.log(n_samples) - 2 * log_likelihood
44
45     return {
46         'nombre': nombre,
47         'params': params,
48         'ks_stat': ks_stat,
49         'p_value': p_value,
50         'aic': aic,
51         'bic': bic,
52         'log_likelihood': log_likelihood
53     }
54
55 # Lista de distribuciones a probar
56 distribuciones = [
57     (stats.expon, 'Exponencial'),
58     (stats.gamma, 'Gamma'),
59     (stats.lognorm, 'Lognormal'),
60     (stats.weibull_min, 'Weibull'),
61     (stats.norm, 'Normal')
62 ]
63
64 # Ajustar todas las distribuciones
65 resultados = []
66 for dist, nombre in distribuciones:
67     try:
68         resultado = ajustar_distribucion(muestra, dist, nombre)
69         resultados.append(resultado)
70     except Exception as e:
71         print(f"Error ajustando {nombre}: {str(e)}")
72
73 # Crear DataFrame con resultados
74 df_resultados = pd.DataFrame(resultados)

```

```

72 df_resultados = df_resultados.sort_values('aic')
73
74 # Imprimir resultados en formato de tabla
75 print("\nComparación de Ajustes de Distribuciones:")
76 print("=====")
77 print("\n{:<15} {:<15} {:<15} {:<15} {:<15} {:<15}".format(
78     "Distribución", "AIC", "BIC", "KS Stat", "Valor p", "Ajuste"))
79 print("-" * 90)
80
81 for _, row in df_resultados.iterrows():
82     # Determinar calidad del ajuste
83     if row['p_value'] > 0.05:
84         ajuste = "Bueno"
85     elif row['p_value'] > 0.01:
86         ajuste = "Regular"
87     else:
88         ajuste = "Malo"
89
90     print("{:<15} {:<15.2f} {:<15.2f} {:<15.4f} {:<15.4f} {:<15}".
91         format(
92             row['nombre'],
93             row['aic'],
94             row['bic'],
95             row['ks_stat'],
96             row['p_value'],
97             ajuste
98         ))
99
100 # Imprimir recomendación detallada
101 print("\nRecomendación:")
102 print("=====")
103 mejor_dist = df_resultados.iloc[0]
104 print(f"\nLa mejor distribución es: {mejor_dist['nombre']}")
105 print(f"\nRazones:")
106 print(f"1. Tiene el menor AIC ({mejor_dist['aic']:.2f})")
107 print(f"2. Tiene el menor BIC ({mejor_dist['bic']:.2f})")
108 print(f"3. Estadístico KS: {mejor_dist['ks_stat']:.4f}")
109 print(f"4. Valor p: {mejor_dist['p_value']:.4f}")
110
111 if mejor_dist['p_value'] > 0.05:
112     print("\nInterpretación:")
113     print("La distribución se ajusta bien a los datos (valor p > 0.05)")
114     print("Podemos usar esta distribución para modelar los tiempos de espera")
115 else:
116     print("\nInterpretación:")
117     print("Aunque es la mejor de las opciones probadas, el valor p sugiere que")
118     print("ninguna de las distribuciones probadas se ajusta perfectamente a los datos")
119     print("Esto podría indicar que los tiempos de espera siguen un patrón más complejo")
120
121 # Crear gráfico comparativo
122 plt.figure(figsize=(15, 10))

```



```

123 # Histograma de datos
124 plt.subplot(2, 1, 1)
125 sns.histplot(muestra, bins=30, stat='density', alpha=0.6, label='
    Datos')
126
127 # Graficar las distribuciones ajustadas
128 x = np.linspace(min(muestra), max(muestra), 1000)
129 for _, row in df_resultados.iterrows():
130     dist = next(d for d, n in distribuciones if n == row['nombre'])
131     y = dist.pdf(x, *row['params'])
132     plt.plot(x, y, label=f"{row['nombre']} (AIC: {row['aic']:.2f})"
133              )
134 plt.title('Comparación de Ajustes de Distribuciones', fontsize=14)
135 plt.xlabel('Tiempo (segundos)', fontsize=12)
136 plt.ylabel('Densidad', fontsize=12)
137 plt.legend()
138
139 # Gráfico Q-Q para la mejor distribución
140 plt.subplot(2, 1, 2)
141 mejor_dist = next(d for d, n in distribuciones if n ==
142                 df_resultados.iloc[0]['nombre'])
143 params = df_resultados.iloc[0]['params']
144 stats.probplot(muestra, dist=mejor_dist, sparams=params, plot=plt)
145 plt.title(f'Gráfico Q-Q para {df_resultados.iloc[0]["nombre"]}',
146          fontsize=14)
147 plt.tight_layout()
148 plt.show()

```

Listing 1: Código para comparación de distribuciones

5.2 Explicación de la Comparación de Distribuciones

Este código extiende nuestro análisis anterior comparando varias distribuciones estadísticas para determinar cuál se ajusta mejor a nuestros datos:

1. **Distribuciones evaluadas:** El código compara las distribuciones exponencial, gamma, lognormal, Weibull y normal.
2. **Criterios de selección:** Para cada distribución, calculamos:
 - AIC (Criterio de Información de Akaike)
 - BIC (Criterio de Información Bayesiano)
 - Estadístico Kolmogorov-Smirnov y su valor p asociado
3. **Evaluación visual:** Se generan gráficos comparativos:
 - Histograma de datos con curvas de densidad de probabilidad superpuestas
 - Gráfico Q-Q para evaluar la bondad de ajuste de la mejor distribución

Estos análisis nos permiten confirmar si la distribución log-normal es realmente la mejor opción para modelar nuestros datos o si alguna otra distribución podría ser más adecuada.

5.3 Estimador de Máxima Verosimilitud para una Distribución Log-Normal

5.3.1

Una variable aleatoria X sigue una distribución log-normal con parámetros μ y σ^2 si la variable aleatoria $Y = \ln(X)$ sigue una distribución normal con media μ y varianza σ^2 .

5.3.2 Función de Densidad de Probabilidad

La función de densidad de probabilidad (FDP) de una distribución log-normal es:

$$f(x; \mu, \sigma^2) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln(x) - \mu)^2}{2\sigma^2}\right), \quad x > 0 \quad (1)$$

5.3.3 Método de Máxima Verosimilitud

Supongamos que tenemos una muestra de observaciones independientes e idénticamente distribuidas X_1, X_2, \dots, X_n de una distribución log-normal. Queremos encontrar los estimadores de máxima verosimilitud para μ y σ^2 .

Función de Verosimilitud

La función de verosimilitud es:

$$L(\mu, \sigma^2) = \prod_{i=1}^n f(x_i; \mu, \sigma^2) \quad (2)$$

$$= \prod_{i=1}^n \frac{1}{x_i\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln(x_i) - \mu)^2}{2\sigma^2}\right) \quad (3)$$

$$= \frac{1}{(\sigma\sqrt{2\pi})^n} \prod_{i=1}^n \frac{1}{x_i} \exp\left(-\frac{(\ln(x_i) - \mu)^2}{2\sigma^2}\right) \quad (4)$$

5.3.4 Log-Verosimilitud

Tomamos el logaritmo natural de la función de verosimilitud para simplificar:

$$\ln L(\mu, \sigma^2) = \ln \left[\frac{1}{(\sigma\sqrt{2\pi})^n} \prod_{i=1}^n \frac{1}{x_i} \exp\left(-\frac{(\ln(x_i) - \mu)^2}{2\sigma^2}\right) \right] \quad (5)$$

$$= -n \ln(\sigma) - \frac{n}{2} \ln(2\pi) - \sum_{i=1}^n \ln(x_i) - \frac{1}{2\sigma^2} \sum_{i=1}^n (\ln(x_i) - \mu)^2 \quad (6)$$

Derivadas Parciales

Para encontrar los estimadores de máxima verosimilitud, derivamos la log-verosimilitud con respecto a μ y σ^2 e igualamos a cero:

$$\frac{\partial \ln L}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^n (\ln(x_i) - \mu) = 0 \quad (7)$$

$$\frac{\partial \ln L}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{i=1}^n (\ln(x_i) - \mu)^2 = 0 \quad (8)$$

Resolución del Sistema

De la primera ecuación:

$$\sum_{i=1}^n (\ln(x_i) - \mu) = 0 \quad (9)$$

$$\sum_{i=1}^n \ln(x_i) - n\mu = 0 \quad (10)$$

$$\mu = \frac{1}{n} \sum_{i=1}^n \ln(x_i) \quad (11)$$

De la segunda ecuación:

$$-\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \sum_{i=1}^n (\ln(x_i) - \mu)^2 = 0 \quad (12)$$

$$\frac{n}{2\sigma^2} = \frac{1}{2(\sigma^2)^2} \sum_{i=1}^n (\ln(x_i) - \mu)^2 \quad (13)$$

$$\frac{n}{2} = \frac{1}{2\sigma^2} \sum_{i=1}^n (\ln(x_i) - \mu)^2 \quad (14)$$

$$n\sigma^2 = \sum_{i=1}^n (\ln(x_i) - \mu)^2 \quad (15)$$

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (\ln(x_i) - \mu)^2 \quad (16)$$

5.3.5 Estimadores de Máxima Verosimilitud

Por lo tanto, los estimadores de máxima verosimilitud para los parámetros de una distribución log-normal son:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \ln(X_i) \quad (17)$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (\ln(X_i) - \hat{\mu})^2 \quad (18)$$

Donde $\hat{\mu}$ es la media muestral de los logaritmos de las observaciones, y $\hat{\sigma}^2$ es la varianza muestral de los logaritmos.

5.3.6 Propiedades de los Estimadores

Los estimadores $\hat{\mu}$ y $\hat{\sigma}^2$ son:

- Consistentes
- Asintóticamente normales
- Asintóticamente eficientes

Además, podemos calcular los estimadores de máxima verosimilitud para la media m y la varianza v de la distribución log-normal original:

$$\hat{m} = \exp\left(\hat{\mu} + \frac{\hat{\sigma}^2}{2}\right) \quad (19)$$

$$\hat{v} = \exp(2\hat{\mu} + \hat{\sigma}^2)(\exp(\hat{\sigma}^2) - 1) \quad (20)$$

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from scipy import stats
6 from scipy.stats import norm
7 import warnings
8 warnings.filterwarnings('ignore')
9
10 # Función para convertir tiempo a segundos
11 def tiempo_a_segundos(tiempo_str):
12     if pd.isna(tiempo_str) or tiempo_str == '':
13         return None
14     try:
15         h, m, s = map(int, tiempo_str.split(':'))
16         return h * 3600 + m * 60 + s
17     except (ValueError, AttributeError):
18         return None
19
20 # Función para calcular intervalo de confianza de la media en
21 # distribución lognormal
22 def intervalo_media_lognormal(muestra, alpha=0.05):
23     log_muestra = np.log(muestra)
24     n = len(log_muestra)
25
26     mu_hat = np.mean(log_muestra)
27     sigma_hat = np.std(log_muestra, ddof=0) # EMV
28
29     # Estimación de la media de X
30     media_hat = np.exp(mu_hat + 0.5 * sigma_hat**2)
31
32     # Margen de error con delta method
33     z = norm.ppf(1 - alpha / 2)
34     se_log_mean = np.sqrt((sigma_hat**2) / n + (sigma_hat**4) / (2
35     * (n - 1)))
36     margen = z * se_log_mean
```

```

36     # Intervalo de confianza para E[X]
37     inferior = np.exp(mu_hat + 0.5 * sigma_hat**2 - margen)
38     superior = np.exp(mu_hat + 0.5 * sigma_hat**2 + margen)
39
40     return media_hat, (inferior, superior)
41
42 # Leer datos
43 df = pd.read_csv('lecturas_excels/Operaciones_Fintoc.txt', sep='\t',
44                 , skiprows=3)
45 df['tiempo_segundos'] = df['tiempo [hh:mm:ss]'].apply(
46     tiempo_a_segundos)
47
48 # Separar datos por estado
49 df_succeeded = df[df['status'] == 'succeeded'].copy()
50 df_failed = df[df['status'] == 'failed'].copy()
51 df_waiting = df[df['status'] == 'waiting'].copy()
52
53 # Obtener la muestra de tiempos exitosos
54 muestra_succeeded = df_succeeded['tiempo_segundos'].dropna().values
55
56 # Calcular estadísticas descriptivas
57 media, intervalo_media = intervalo_media_lognormal(
58     muestra_succeeded)
59 mu_hat = np.mean(np.log(muestra_succeeded))
60 sigma_hat = np.std(np.log(muestra_succeeded), ddof=0)
61
62 print("\nAnálisis de Tiempos de Espera (Distribución Lognormal)")
63 print("=====")
64 print(f"\nEstadísticas Descriptivas:")
65 print(f"Media estimada: {media:.2f} segundos")
66 print(f"Intervalo de confianza al 95%: ({intervalo_media[0]:.2f}, {
67     intervalo_media[1]:.2f}) segundos")
68 print(f"Parámetro mu (log-normal): {mu_hat:.4f}")
69 print(f"Parámetro sigma (log-normal): {sigma_hat:.4f}")
70
71 # Crear gráficos
72 plt.figure(figsize=(15, 10))
73
74 # 1. Histograma con ajuste lognormal
75 plt.subplot(2, 1, 1)
76
77 # Graficar histograma para transacciones exitosas
78 sns.histplot(muestra_succeeded, bins=30, stat='density', alpha=0.6,
79             color='steelblue', label='Exitosas')
80
81 # Graficar histograma para transacciones fallidas
82 muestra_failed = df_failed['tiempo_segundos'].dropna().values
83 sns.histplot(muestra_failed, bins=30, stat='density', alpha=0.6,
84             color='tomato', label='Fallidas')
85
86 # Graficar PDF lognormal para transacciones exitosas
87 x = np.linspace(min(muestra_succeeded), max(muestra_succeeded),
88                 1000)
89 pdf_lognorm = stats.lognorm.pdf(x, s=sigma_hat, scale=np.exp(mu_hat))
90
91 plt.plot(x, pdf_lognorm, 'darkblue', label='Ajuste Lognormal (
92     Exitosas)')

```

```

86
87 # Agregar línea vertical para la media
88 plt.axvline(x=media, color='darkgreen', linestyle='--',
89             linewidth=2, label=f'Media: {media:.2f}s')
90
91 # Sombrear el intervalo de confianza
92 plt.axvspan(intervalo_media[0], intervalo_media[1], alpha=0.2,
93             color='green',
94             label=f'IC 95%: ({intervalo_media[0]:.2f}, {
95                 intervalo_media[1]:.2f})s')
96
97 plt.title('Distribución de Tiempos de Espera', fontsize=14)
98 plt.xlabel('Tiempo (segundos)', fontsize=12)
99 plt.ylabel('Densidad', fontsize=12)
100 plt.legend()
101
102 # 2. Boxplot comparativo
103 plt.subplot(2, 1, 2)
104
105 # Crear boxplot comparativo solo para succeeded y failed
106 df_filtered = df[df['status'].isin(['succeeded', 'failed'])]
107 sns.boxplot(x='status', y='tiempo_segundos', data=df_filtered,
108             hue='status', palette={'succeeded': 'steelblue', 'failed': 'tomato'},
109             legend=False)
110
111 # Superponer swarmplot para ver distribución de puntos
112 sns.swarmplot(x='status', y='tiempo_segundos', data=df_filtered,
113              hue='status', palette={'succeeded': 'darkblue', 'failed': 'darkred'},
114              alpha=0.7, size=4, legend=False)
115
116 # Personalizar boxplot
117 plt.title('Comparación de Tiempos por Estado de Transacción',
118          fontsize=14)
119 plt.xlabel('Estado', fontsize=12)
120 plt.ylabel('Tiempo (segundos)', fontsize=12)
121 plt.xticks([0, 1], ['Exitosas', 'Fallidas'])
122 plt.ylim(0, np.percentile(df_filtered['tiempo_segundos'].dropna(),
123                          95) * 1.2)
124
125 # Agregar estadísticas en el gráfico
126 plt.text(0, np.percentile(df_succeeded['tiempo_segundos'].dropna(),
127                          95),
128          f"n={len(df_succeeded)}\nMedia={df_succeeded['tiempo_segundos'].mean():.2f}s",
129          ha='center', va='bottom', fontsize=10, bbox=dict(facecolor='white', alpha=0.8))
130 plt.text(1, np.percentile(df_failed['tiempo_segundos'].dropna(),
131                          95),
132          f"n={len(df_failed)}\nMedia={df_failed['tiempo_segundos'].mean():.2f}s",
133          ha='center', va='bottom', fontsize=10, bbox=dict(facecolor='white', alpha=0.8))
134
135 plt.tight_layout()
136 plt.show()

```

```

131
132 # Prueba de bondad de ajuste
133 ks_stat, p_value = stats.kstest(muestra_succeeded, 'lognorm', args
    =(sigma_hat,), alternative='two-sided')
134 print(f"\nPrueba de bondad de ajuste (Kolmogorov-Smirnov):")
135 print(f"Estadístico KS: {ks_stat:.4f}")
136 print(f"Valor p: {p_value:.4f}")
137
138 if p_value > 0.05:
139     print("\nInterpretación:")
140     print("La distribución lognormal es un buen ajuste para los
    datos (valor p > 0.05)")
141 else:
142     print("\nInterpretación:")
143     print("La distribución lognormal no se ajusta perfectamente a
    los datos")
144     print("Esto podría indicar que los tiempos de espera siguen un
    patrón más complejo")

```

Listing 2: Código para análisis de distribución log-normal

5.4 Explicación del Código

El código implementa los siguientes componentes clave:

1. **Procesamiento de datos:** Convierte los tiempos en formato HH:MM:SS a segundos para su análisis.
2. **Estimadores de máxima verosimilitud:** Calcula $\hat{\mu}$ y $\hat{\sigma}^2$ mediante:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \ln(X_i) \quad (21)$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (\ln(X_i) - \hat{\mu})^2 \quad (22)$$

3. **Intervalo de confianza:** Utiliza el método delta para calcular el intervalo de confianza para la media de la distribución log-normal.
4. **Visualización:** Genera histogramas con la curva de densidad log-normal ajustada y box plots comparativos para evaluar visualmente el ajuste.
5. **Prueba de bondad de ajuste:** Aplica la prueba de Kolmogorov-Smirnov para verificar formalmente si los datos siguen una distribución log-normal.