Documentación del Proyecto – Grupo 3

Universidad Nacional de Lanús.   
Ciclo Lectivo.

Licenciatura en Sistemas – Ingeniería de Software III

Gestión de Calidad, Estimación y Auditoría del Sistema de Pago del Transporte Público

**Equipo Docente:** Lic. Nicolás Pérez, Lic. Solange Lescano

**Integrantes:**   
 - Ruiz Pereira,Roberto Andres (GitHub: rpruiz33)  
 - Gomez, Johann Oriel (GitHub: johanngomez)   
 - Fernández Breda, Sol (GitHub: sol-fernandez-breda)  
 - Mansilla Anahí Maitén (GitHub: Anahimm)   
 - Perafan, Valentina Manuela (GitHub: ValenPerafan)

**Repositorio base del trabajo:** <https://github.com/rpruiz33/Ing-SW-3-tp->

# Índice

1. Contexto del Proyecto 3

2. Estructura del Repositorio 3

3. Requisitos Previos e Instalación 5

4. Cronograma de entregas 5

5. Aplicación de Modelos y Estándares de Calidad 5

6. Estimación de Recursos y Esfuerzo 5

6.2. Estimación por Puntos de Función 6

6.3 Asignación de Recursos y Cronograma Interno 6

7. Métricas y Mejora Continua 8

7.2. Métricas de Performance por Períodos 8

8. Análisis de Variables Clave (MIC-MAC) 8

9. Gestión de Riesgos y Contingencia 9

10. Auditoría de Sistemas y Peritajes Informáticos 11

11. Metodología CRISP-DM en Transporte 11

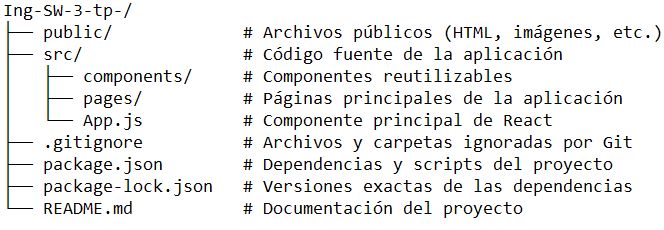
12. Cierre del Proyecto y Lecciones Aprendidas 11

# 1. Contexto del Proyecto

Este trabajo práctico se basa en un caso simulado del Gobierno Nacional que busca implementar un sistema unificado de pago para el transporte público.   
Como equipo consultor, nuestra tarea es aplicar conocimientos de calidad, auditoría, métricas y estimación para asesorar la implementación del sistema, con foco en el usuario final, la trazabilidad de los datos y la transparencia del desarrollo.

# 2. Estructura del Repositorio

La estructura general del repositorio base utilizado para el desarrollo es:



Dentro de la carpeta src/ se encuentran los archivos principales de la aplicación:

* components/: Contiene componentes visuales reutilizables como menús, formularios y elementos de navegación. Estos elementos permiten mantener una estructura modular y coherente en toda la interfaz del sistema. Por ejemplo:
  + *Footer.jsx:* Este componente representa el pie de página de la aplicación. Incluye información adicional como redes sociales y derechos de autor. Su presencia en todas las páginas asegura una experiencia de usuario coherente y profesional.
  + *Historial.jsx:* Este componente recupera los movimientos almacenados en localStorage (recargas y pagos de boletos) y los presenta en una tabla detallada. Además, genera un resumen agrupado por número de línea, mostrando cuántos pasajeros usaron cada una y el total recaudado. La información se presenta de forma clara y permite al usuario revisar su historial y analizar sus viajes y gastos de forma ordenada.
  + *Home.jsx y Home2.jsx:* Ambos componentes están destinados a la página de inicio de la aplicación. Home.jsx siendo la versión principal, mientras que Home2.jsx es una variante en desarrollo. Da la bienvenida al usuario y proporcionar acceso rápido a las funcionalidades principales del sistema.
  + *Login.jsx:* Este componente gestiona el proceso de autenticación de usuarios. Incluye formularios para ingresar credenciales y lógica para validar el acceso, asegurando que solo usuarios autorizados puedan acceder a las funcionalidades protegidas de la aplicación.
  + *Navbar.jsx:* Representa la barra de navegación superior de la aplicación. Proporciona enlaces a las diferentes secciones del sistema, como inicio, recarga, historial, etc. Facilita la navegación y mejora la usabilidad al permitir un acceso rápido a las distintas funcionalidades.
  + *PagarBoleto.jsx:* Este componente permite a los usuarios simular el pago de un boleto de transporte, seleccionando una línea y una distancia recorrida. Se encarga de calcular el monto correspondiente y registrar la operación en el historial del usuario. El usuario debe ingresar el número de la línea de transporte (por ejemplo, “60”) y seleccionar una de las tres distancias predefinidas (1 km = $400, 6 km = $800, 10 km = $1200). Al enviar el formulario, se simula el pago del pasaje y se muestra un mensaje de confirmación. La operación también se almacena en el localStorage para su posterior consulta en el historial de movimientos.
  + *Recarga.jsx:* Facilita al usuario la recarga de saldo en su cuenta. Permite a los usuarios ingresar un monto, seleccionar un medio de pago (como MercadoPago, Cuenta DNI, etc.) y un medio de transporte (colectivo, subte o tren), y luego ejecutar una recarga. Al completar el formulario correctamente, el saldo del usuario se actualiza y se almacena el movimiento en el historial de recargas

Estas páginas corresponden directamente a funcionalidades estimadas por puntos de función en la sección 6.2 del presente informe.

* App.js: Es el componente principal que gestiona las rutas y la navegación general del sistema. Se apoya en librerías como react-router-dom para vincular URLs con vistas específicas, como /login, /recarga, /historial.

Por último, se destaca que cada componente implementado tiene correspondencia directa con tareas detalladas en el cronograma interno (ver sección 6.3). Por ejemplo:

* Recarga.jsx se asocia con las tareas 22 y 23.
* Login.jsx se corresponde con la tarea 22.
* Navbar.jsx responde a la tarea 21.
* Historial.jsx está vinculado con la funcionalidad de consulta de viajes.

Este vínculo entre código fuente, estimaciones y cronograma permite asegurar la trazabilidad entre planificación y ejecución.

# 3. Requisitos Previos e Instalación

- Node.js (v14+) – npm

*Instalación:*

1. git clone <https://github.com/rpruiz33/Ing-SW-3-tp->
2. cd Ing-SW-3-tp-
3. npm install
4. npm start

# 4. Cronograma de entregas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N° | Hito | Fecha | Entregable asociado | Responsable |
| 1 | Entrega 1 | 19/05/2025 | Versión intermedia del TP | Equipo 3 |
| 2 | Entrega 2 (Final) | 16/06/2025 | Documento final completo | Equipo 3 |

# 5. Aplicación de Modelos y Estándares de Calidad

Se eligió aplicar el modelo CMMI Nivel 2. Se analiza cómo se aplicaría en:

* Desarrollo: definición de procesos repetibles, gestión de requisitos.
* Mantenimiento: control de versiones, gestión de incidencias.
* Atención al usuario: documentación accesible, trazabilidad de reclamos.

Autoevaluación:

* Desarrollo: Media
* Mantenimiento: Baja
* Atención al usuario: Media

# 6. Estimación de Recursos y Esfuerzo

Se utilizaron puntos de función para estimar funcionalidades como:

* Registro de usuario
* Pago con QR
* Carga de saldo automática
* Validación en transporte
* Visualización de historial

Se utilizó una estructura WBS (Work Breakdown Structure) para dividir el proyecto en módulos y submódulos organizados jerárquicamente. Esta estructura fue esencial para organizar el trabajo de desarrollo, asignar tareas y estimar tiempos de ejecución.

El proyecto se dividió en:

* Módulo Servicio de Pago: boceto, base de datos, integración con billeteras.
* Módulo Cliente: ABM, asociaciones con medios de pago y transportes.
* Módulo Servicio Público: asociación medio-pago, demanda y reportes.
* Desarrollo web y validación: implementación visual, revisión del cliente, despliegue.

Cada tarea del WBS está asociada a una funcionalidad del sistema, reflejada directamente en la estructura de carpetas y componentes del repositorio.

6.2. Estimación por Puntos de Función

Se aplicó la técnica de Puntos de Función para estimar el esfuerzo de desarrollo de las principales funcionalidades:

|  |  |  |  |
| --- | --- | --- | --- |
| **Funcionalidad** | **Tipo** | **Complejidad** | **Puntos de Función** |
| Registro de usuario | Entrada | Media | 4 |
| Carga de saldo con billetera | Entrada | Alta | 6 |
| Lectura de QR en transporte | Salida | Alta | 7 |
| Consulta de historial de viajes | Consulta | Baja | 3 |
| Asociación cliente-medio pago | Interna | Media | 4 |

Total estimado: 24 puntos. Esto se traduce en un esfuerzo aproximado de 3 meses-persona. Esta estimación fue la base para la planificación del cronograma y distribución del trabajo.

6.3 Asignación de Recursos y Cronograma Interno

A partir del WBS y los puntos de función, se definió un cronograma detallado por tareas y responsables. Cada integrante asumió tareas técnicas en módulos específicos.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Horas Estimadas** | **Días Estimados** | **Tarea WBS** | **Nro Actividad** | **Actividad** | **Fecha Inicio** | **Predecesora** | **Responsable** |
| **5** | 1 | 1.1.1 | 0 | - Diseñar boceto en Figma | 10-4-2025 | - | **Todos** |
| **4** | 1 | 1.1.2 | 1 | -Diseñar entidad en modelo relacional (DER) | 11-4-2025 | 0 | **Ruiz Pereira** |
| **6** | 2 | 1.1.2 | 2 | - Crear base de datos | 12-4-2025 | 1 | **Anahi Mansilla** |
| **3** | 1 | 1.1.2 | 3 | -Insertar datos en Base de Datos | 14-4-2025 | 2 | **Ruiz Pereira** |
| **4** | 1 | 1.1.3 | 4 | - Analizar APIs de billeteras digitales | 12-4-2025 | - | **Johann Gomez** |
| **10** | 2 | 1.1.3 | 5 | - Implementar integración con billeteras | 14-4-2025 | 4 | **Johann Gomez** |
| **3** | 1 | 1.1.4 | 6 | - Pruebas de validación transacciones | 21-4-2025 | 5 | **Anahi Mansilla** |
| **6** | 2 | 1.2.1 | 7 | - ABM de clientes(back end) | 21-4-2025 | - | **Johann Gomez** |
| **3** | 1 | 1.2.2 | 8 | - Relacionar clientes con medios de pago | 22-4-2025 | 7 | **Johann Gomez** |
| **3** | 1 | 1.2.3 | 9 | - Registrar líneas de transporte | 23-4-2025 | 2 | **Johann Gomez** |
| **3** | 1 | 1.2.4 | 10 | - ABM para líneas de transporte | 24-4-2025 | - | **Valentina Perafan** |
| **4** | 1 | - | 11 | - Reunion de equipo | 26-4-2025 | -- | **todos** |
| **3** | 1 | - | 12 | -Reunion con el cliente | 7-5-2025 | - | **todos** |
| **4** | 1 | 1.3.1 | 13 | - Asociar viaje con medio de pago | 7-5-2025 | 2 | **Johann Gomez** |
| **3** | 1 | 1.3.1 | 14 | - Test de asociación viaje - pago | 8-5-2025 | 13 | **Anahi Mansilla** |
| **3** | 1 | 1.3.1 | 15 | - Testeo de asociación cliente-pago | 9-5-2024 | 7 | **Anahi Mansilla** |
| **4** | 1 | 1.3.2 | 16 | - Cálculo de demanda y frecuencia | 10-5-2025 | - | **Sol Fernandez Breda** |
| **4** | 1 | 1.3.2 | 17 | - Programar cálculo de demanda | 11-5-2025 | 16 | **Sol Fernandez Breda** |
| **3** | 1 | 1.3.3 | 18 | - Definir estructura de reportes | 12-5-2025 | - | **Sol Fernandez Breda** |
| **4** | 1 | 1.3.3 | 19 | - Implementar generación de reportes | 13-5-2025 | 18 | **Sol Fernandez Breda** |
| **4** | 1 | 2.1.1 | 20 | - Desarrollo de pantalla de inicio | 22-5-2025 | - | **Ruiz Pereira** |
| **2** | 1 | 2.1.1 | 21 | - Desarrollo de navbar y footer | 23-5-2025 | - | **Ruiz Pereira** |
| **2** | 1 | 2.1.1 | 22 | - Desarrollo login | 24-5-2025 | - | **Ruiz Pereira** |
| **3** | 1 | 2.1.1 | 23 | - Desarrollo de pantalla recarga | 26-5-2025 | - | **Ruiz Pereira** |
| **2** | 1 | 2.1.1 | 24 | - Validar errores de acceso | 2-6-2025 | 22 | **Anahi Mansilla** |
| **2** | 1 | 2.1.2 | 25 | - Reunion y validacion con el cliente | 3-6-2025 | 1 a 24 | **Valentina Perafan** |
| **4** | 1 | 2.1.3 | 26 | - Compilación y despliegue | 4-6-2025 | 25 | **Ruiz Pereira** |
| **3** | 1 | 2.1.4 | 27 | - Documentar | 5-6-2025 | 26 | **Sol Fernandez Breda** |
| **18** | 3 | 2.1.4 | 28 | - Auditoría | 6-6-2025 | todas | **Valentina Perafan** |

# 7. Métricas y Mejora Continua

Se definen:

* Métrica de software: % de disponibilidad mensual
* Métrica de proceso: Tiempo promedio de resolución de bugs
* Métrica de usuario: Satisfacción del usuario (% positivos)

Acciones:

* Automatización de pruebas
* Refinamiento de tickets
* Encuestas mensuales

7.2. Métricas de Performance por Períodos

Se aplicó el método EV (Earned Value) para medir el avance del proyecto en 5 períodos claves. Para cada semana de control, se calcularon:

* PV (Planned Value): valor planeado según cronograma.
* EV (Earned Value): valor realmente ganado.
* AC (Actual Cost): costo real acumulado.
* TCPI (To-Complete Performance Index): índice de rendimiento para completar.

También se calcularon métricas derivadas:

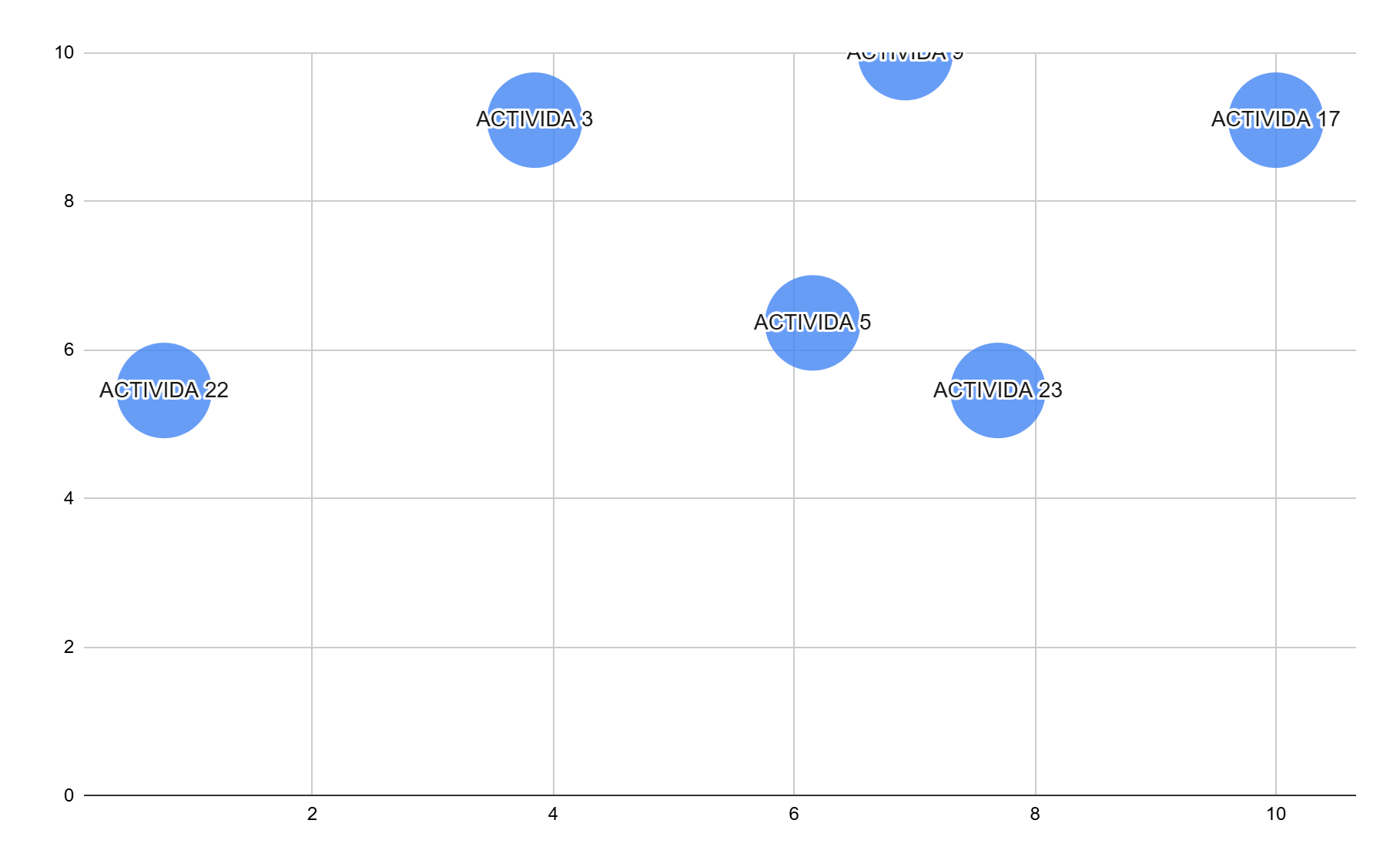
|  |  |  |
| --- | --- | --- |
| Métrica | Significado | Ejemplo Estimaciones P1 |
| SV | Variación de cronograma | -7 |
| CV | Variación de costos | -2 |
| SPI | Índice de desempeño en plazo | 0.78 |
| CPI | Índice de desempeño en costo | 0,98 |
| TCPI | índice de desempeño proyectado | 1,02 |

# 8. Análisis de Variables Clave (MIC-MAC)

Se utilizó la técnica MIC-MAC para clasificar actividades críticas según su influencia y dependencia. Esta herramienta fue clave para priorizar tareas y detectar posibles cuellos de botella.

Las actividades con alta influencia y dependencia se clasificaron como variables clave (por ejemplo: integración con billeteras y validación de transacciones). Se desarrollaron antes que otras menos críticas para mitigar riesgos.

A continuación, se presenta un mapa de posicionamiento:



# 9. Gestión de Riesgos y Contingencia

Se identificaron riesgos clave en fases críticas del proyecto:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Nombre del Riesgo** | **Módulo WBS** | **Prob** | **Gravedad** | **Impacto** | **Tipo** | **Umbral** |
| **1** | Herramienta deprecada, sin soporte | 2,1 | 3 | 5 | 15 | Estrategico | Cuando falten 3 meses para que la libreria deje de estar disponible ejecuto el plan de acción |
| **2** | Problemas de integracion con otros  sistemas | 1.1 | 4 | 5 | 20 | Externo | Si luego de 2 semanas de trabajo técnico no se logra una integración estable o  aparecen más de 3 fallas críticas en la  interoperabilidad, |
| **3** | Falta de pruebas | 1.3 | 3 | 5 | 15 | Interno | Si al realizar la primera integración del sistema se detectan más de 5 fallas no previstas, se debe suspender el pase  a producción hasta realizar  pruebas de regresión completas. |
| **4** | Cisne negro : caida de la red | 2..1 | 5 | 5 | 25 | Externo | Si la caída de red supera las 2 horas sin recuperación, se activa el plan de  contingencia de trabajo offline |
| **5** | No cumplir con normativas legales o regulatorias. | 1.1 | 3 | 4 | 12 | Interno | Detectar al menos 1 hallazgo de  incumplimiento en auditorías internas o externas obliga a detener el despliegue hasta su resolución. |

Se definieron planes de contingencia y se hizo seguimiento durante los sprints de implementación.

*RIESGO DE EJEMPLO:*

Caída de la red por desastres naturales o por falta de infraestructura adecuada para situaciones que la requieran. Este es un riesgo que puede ser externo por desastres naturales como puede ser una catástrofe climática natural o por ataques o interferencias que hacen la denegación de los servicios de conectividad en donde el sistema depende de la conectividad y por desastres naturales o por falta y o deficiencia de la infraestructura, como también se puede producir por falta de suministro eléctrico, se produce una caída en la red

*PLANES DE CONTINGENCIA:*

1) Tomamos como parámetros dos horas sin conexión a la red, entonces aplicamos el plan de contingencia de tener un back up de los datos y así poder trabajar en forma offline, por ejemplo el equipo de desarrollo tendrá que trabajar sin soporte online y se verá reducido a tareas quizás un poco mas triviales en tanto y en cuanto no se puedan hacer cambios en el repositorio remoto, sin poder obviamente recuperar o generar nuevos cambios, de esa misma metodología trabajaran los demás actores de la organización se deberán enfocarse en tareas que puedan realizarse localmente, como procesamiento de documentos, revisiones internas o preparación de entregables, esperando una solución de la parte técnica sobre la infraestructura

2) Tener contemplado que la infraestructura de redes puede fallar entonces diseñar un esquema de red redundante, con respaldo físico y lógico (routers dobles, enlaces múltiples, topología tolerante a fallas). Contratar servicios de respaldo con otro proveedor de Internet, preferentemente con tecnología diferente. Implementar UPS y generadores eléctricos para mantener los dispositivos de red activos durante cortes de energía. Monitorear el estado de la red mediante herramientas de detección de caídas o degradación del servicio. Capacitar al personal técnico para actuar rápidamente ante fallos físicos o lógicos. Documentar y realizar simulacros periódicos para verificar su eficacia.

3) Dado que una caída de red afecta directamente al servicio de transporte, se propone desarrollar un modo offline inteligente que permita continuar operando temporalmente sin conectividad. Este modo permitiría validar pagos con datos precargados (como saldo o historial reciente) y almacenar localmente las transacciones hasta que la red se restablezca y se puedan sincronizar. Se establecerán reglas de uso controlado (por ejemplo, límite de viajes sin conexión) y mensajes claros dentro de la app para informar a los usuarios. El personal en campo deberá contar con procedimientos para verificar manualmente datos básicos y actuar ante fraudes o errores. Esta estrategia garantiza la continuidad del servicio al usuario y minimiza el impacto operativo durante caídas de red.

# 10. Auditoría de Sistemas y Peritajes Informáticos

Escenario: Reporte de cobros duplicados. Auditoría interna:

* Se solicita log de transacciones, base de datos y control de acceso.
* Herramientas: EnCase, Write Blocker

Informe técnico:

* Hallazgo: error en la validación de recargas.
* Recomendación: revisión del módulo de validaciones antes del deploy.

# 11. Metodología CRISP-DM en Transporte

*Caso: Predecir zonas de alta demanda*

1. Comprensión del negocio: Optimizar rutas
2. Comprensión de los datos: Origen/destino por día
3. Preparación: Limpieza y georreferenciación
4. Modelado: Clustering de zonas
5. Evaluación: Comparación con datos reales
6. Despliegue: Dashboard para planificadores urbanos

# 12. Cierre del Proyecto y Lecciones Aprendidas

*Checklist de Cierre:*

Al finalizar el proyecto, se verificaron los siguientes puntos como parte del cierre formal:

* Documentación técnica y de gestión completa.
* Revisión del código fuente y su correcta estructuración.
* Validación funcional del sistema (casos de uso clave).
* Implementación de pruebas básicas y control de calidad.
* Registro y seguimiento de tareas en base al cronograma WBS.
* Evaluación de métricas de avance y desempeño por período.
* Elaboración de informe final de auditoría y contingencia.

*Lecciones Aprendidas:*

A lo largo del desarrollo del sistema de pago del transporte público, se adquirieron aprendizajes significativos, tanto técnicos como metodológicos:

1. La planificación mediante Puntos de Función y WBS fue fundamental para estimar de manera realista los recursos y tiempos requeridos. Esta técnica facilitó una asignación más precisa de tareas y permitió establecer prioridades claras desde el inicio del proyecto.
2. El seguimiento con métricas PV, EV, AC y los indicadores derivados (SPI y CPI) permitió anticipar desvíos y corregir el rumbo del proyecto en tiempo real. Estas herramientas, aprendidas en clase, se integraron naturalmente a nuestra práctica y demostraron ser esenciales para la gestión efectiva del avance.
3. El uso del análisis MIC-MAC fue clave para identificar variables y actividades críticas, permitiendo priorizar tareas con alto impacto y dependencia. Esto evitó cuellos de botella en etapas sensibles como la integración de billeteras digitales y validación de transacciones.
4. La definición y monitoreo de riesgos, así como la preparación de planes de contingencia, contribuyeron a una mayor solidez del proyecto. Esta visión preventiva permitió adaptarse rápidamente ante imprevistos y responder con soluciones concretas.
5. El trabajo colaborativo, con distribución clara de roles y entregables por integrante, favoreció la eficiencia y la responsabilidad compartida, reflejando un enfoque profesional en la dinámica de equipo.