**Plan de dirección**

**de proyecto**

***[Sistema integral de flota]***

***Fecha: [12/03/2025]***

**Tabla de contenido**

Información del proyecto 4

Aprobaciones 4

Introducción 5

Planes de gestión de proyecto subsidiarios 6

Plan de gestión del alcance 6

Plan de gestión de requerimientos 6

Plan de gestión del cronograma 7

Plan de gestión de calidad 8

Plan de gestión de comunicaciones 9

Plan de gestión de riesgos de un proyecto 9

Plan de gestión de los interesados 10

15

**Información del proyecto**

**Datos**

| Empresa / Organización | PETER SA |
| --- | --- |
| Proyecto | Sistema integral de flota |
| Fecha de preparación | 12/03/2024 |
| Cliente | Logistics |
| Patrocinador principal | Alvaro Moreno |
| Gerente de Proyecto | Luis Arias |

**Patrocinador / Patrocinadores**

| **Nombre** | **Cargo** | **Departamento / División** | **Rama ejecutiva (Vicepresidencia)** |
| --- | --- | --- | --- |
| Alvaro Moreno | Gerente General | Gerencia General | - |

**Introducción**

**Introducción**

Este documento constituye el Plan de Dirección del Proyecto (PDP) para el desarrollo del "Sistema de Control de Flota para Móviles Mercantiles", realizado como proyecto Capstone de Ingeniería Informática. El propósito de este PDP es definir cómo se ejecutará, monitoreará, controlará y cerrará el proyecto, sirviendo como guía principal para el equipo de desarrollo y los stakeholders involucrados.

El proyecto aborda la necesidad de optimizar la gestión de flotas en empresas logísticas mediante una solución de software centralizada. Los objetivos clave incluyen mejorar la eficiencia operativa (reducción del 20% de inactividad), asegurar la correcta adopción del sistema (capacitación del 85% de usuarios) y garantizar la integridad de los datos (registro del 95% de activos).

Este plan integra diversos planes subsidiarios (alcance, requerimientos, cronograma, costos, calidad, etc.) y establece las líneas base contra las cuales se medirá el desempeño del proyecto. Se enfoca en aplicar las mejores prácticas de la ingeniería de software y la gestión de proyectos adaptadas al contexto académico y a los recursos disponibles para un proyecto Capstone.

**Planes de gestión de proyecto subsidiarios**

**Plan de gestión del alcance**

### 1. Objetivo del Plan de Gestión del Alcance

El objetivo principal del plan de alcance del proyecto es establecer con claridad qué está incluido y qué queda fuera del proyecto. Esto permite al equipo alinear expectativas entre todos los stakeholders, facilitar la toma de decisiones, y manejar de forma eficiente cualquier cambio que pueda surgir durante su ejecución.

Entregables clave del proyecto:

Acta de Constitución: Documento que marca el inicio formal del proyecto y define sus objetivos generales.

Plan de Gestión: Guía para organizar y controlar las actividades del proyecto.

Documento de Requisitos: Recoge detalladamente las funcionalidades que debe tener el sistema.

Código Fuente: Resultado del desarrollo técnico conforme a los requisitos establecidos.

Informe Final: Evalúa el cumplimiento de los objetivos y los resultados obtenidos.

Principales restricciones del proyecto:

Plazo de entrega inflexible.

Tiempo limitado para realizar pruebas, validaciones y ajustes.

Presión de tiempos durante el desarrollo.

Dependencia de revisiones con escaso margen para aplicar correcciones.

### 

### 

### 

### 2. Definición del Alcance

El alcance del **Sistema de Control de Flota** será determinado con base en los requisitos funcionales y no funcionales del sistema. Esto incluirá:

* **Gestión de flota**: La plataforma debe permitir el seguimiento de los vehículos, programación de rutas, y gestión de la carga.
* **Interfaz de usuario**: Creación de una interfaz de usuario sencilla y accesible para los administradores, operarios y otros usuarios del sistema.
* **Notificaciones**: Sistema de alertas y notificaciones para usuarios y administradores sobre el estado de la flota y las rutas.
* **Integración**: Integración con sistemas existentes para facilitar la gestión de datos de la flota y la carga.

Aclaración de lo que está fuera del alcance del proyecto:

* **Machine Learning**: No se implementarán técnicas de optimización avanzada de rutas a través de algoritmos de machine learning.
* **Módulo de Ruedas**: No se implementará un módulo específico para la gestión de ruedas o llantas.
* **Gestión de Inventario**: No se manejará un sistema de inventario; solo se incluirá la carga.

### 3. Criterios de Aceptación del Alcance

Para cada entregable del proyecto, se definirán los **criterios de aceptación**, los cuales establecerán los estándares que deben cumplirse para que se considere que el entregable ha sido completado con éxito. Algunos ejemplos de criterios de aceptación:

* **Entrega del software**: El sistema debe permitir la gestión de flota, registrar datos de vehículos, gestionar rutas, y ser accesible para usuarios designados.
* **Documentación del sistema**: Se debe entregar un manual de usuario y un documento técnico con las especificaciones del sistema.
* **Pruebas de validación**: Se debe realizar una serie de pruebas para validar el correcto funcionamiento del sistema bajo escenarios reales de uso.

### 4. Desglose de Trabajo (WBS)

El proyecto se desglosará en paquetes de trabajo específicos, y cada paquete se desglosará en actividades más pequeñas para facilitar su gestión y ejecución. Este desglose de trabajo se organizará en torno a los siguientes componentes:

1. **Planificación y diseño**:  
   * Recolección de requisitos.
   * Diseño técnico y de la base de datos.
   * Planificación de la implementación.
2. **Desarrollo e implementación**:  
   * Programación de la interfaz de usuario.
   * Desarrollo de funcionalidades de gestión de flota.
   * Implementación del sistema de notificaciones.
3. **Pruebas y validación**:  
   * Pruebas unitarias.
   * Pruebas de integración.
   * Validación con usuarios clave.
4. **Despliegue y cierre**:  
   * Despliegue final del sistema.
   * Capacitación y entrega de documentación.

### 5. Control de Cambios en el Alcance

El control de cambios en el alcance será realizado mediante un proceso formalizado de gestión de cambios. Cualquier solicitud de cambio en el alcance será evaluada en términos de impacto en el presupuesto, cronograma, y otros aspectos clave del proyecto. El proceso de control de cambios incluirá:

* **Solicitud de cambio**: Los stakeholders pueden presentar una solicitud formal de cambio, la cual será evaluada por el equipo del proyecto.
* **Evaluación de impacto**: Cada solicitud de cambio se evaluará para determinar su impacto en el alcance, cronograma y costo del proyecto.
* **Aprobación del cambio**: Los cambios en el alcance solo se implementarán una vez que se haya obtenido la aprobación del Gerente de Proyecto y los stakeholders clave.

### 6. Herramientas y Técnicas para la Gestión del Alcance

El equipo de proyecto utilizará las siguientes herramientas y técnicas para gestionar el alcance:

* **Reuniones periódicas con stakeholders**: Para asegurarse de que las expectativas del cliente estén alineadas con lo que se entrega.
* **Técnicas de análisis de requisitos**: Como entrevistas, encuestas y análisis de documentos, para comprender las necesidades y expectativas de los stakeholders.
* **Software de gestión de proyectos**: Utilización de herramientas como Jira o Trello para hacer seguimiento a las actividades y tareas relacionadas con el alcance.

### 7. Responsabilidades en la Gestión del Alcance

* **Gerente de Proyecto**: Responsable de garantizar que el alcance se mantenga dentro de los límites establecidos y de aprobar cualquier cambio en el alcance.
* **Equipo de Desarrollo**: Encargado de implementar las funcionalidades que forman parte del alcance del proyecto, según lo acordado.
* **Stakeholders**: Involucrados en la revisión y validación del alcance, asegurándose de que sus necesidades sean cubiertas.

**Plan de gestión de requerimientos**

El **Plan de Gestión de Requerimientos** es un documento que define cómo se identificarán, documentarán, gestionarán y controlarán los requisitos del proyecto a lo largo de su ciclo de vida. Este plan es fundamental para asegurar que el proyecto entregue lo que se espera, cumpliendo con las expectativas del cliente y los interesados. A continuación, te doy un ejemplo de cómo podrías explicar el plan para tu **Sistema de Control de Flota**:

#### Objetivo del Plan de Gestión de Requerimientos

El objetivo de este plan es establecer un marco para identificar, analizar, documentar y gestionar los requisitos del **Sistema de Control de Flota**. Asegura que todos los requisitos del proyecto sean comprendidos, aprobados y rastreados a lo largo de su desarrollo, garantizando la entrega de un sistema que cumpla con las expectativas de los usuarios y los stakeholders.

**Requerimientos Funcionales:** Describirán las acciones que el sistema debe realizar:

"El sistema debe permitir asignar un vehículo a un conductor registrando fecha y hora de inicio y fin."

"El sistema debe permitir registrar un siniestro, adjuntando fotos y documentos."

"El sistema debe generar alertas automáticas 5 días antes del vencimiento de la revisión técnica."

"El sistema debe permitir exportar el reporte de consumo de combustible por vehículo a formato PDF y Excel."

"El sistema debe permitir crear usuarios con roles de Administrador, Contratista o Prevencionista, restringiendo el acceso a módulos según el rol."

"El sistema debe mostrar en el dashboard el porcentaje de flota activa vs. inactiva."

"El sistema debe permitir a los conductores completar checklists de seguridad predefinidos desde un dispositivo móvil (si aplica)."

"El sistema debe registrar el historial de infracciones por conductor."

"El sistema debe mostrar la ubicación en tiempo real de los vehículos en un mapa."

**Requerimientos No Funcionales:** Describirán las características de calidad del sistema. Ejemplos:

**Rendimiento:** "El dashboard interactivo debe cargar en menos de 5 segundos."

**Usabilidad:** "La interfaz debe ser intuitiva para que el 85% de los usuarios capacitados puedan usar las funciones principales tras 10 semanas."

**Seguridad:** "El acceso a la edición de roles y permisos estará restringido únicamente al rol Administrador."

**Disponibilidad:** "El sistema debe tener una disponibilidad del 99.5% durante el horario laboral." (Relacionado con disminuir tiempo de inactividad).

**Compatibilidad:** "La interfaz web debe ser compatible con los navegadores Chrome, Firefox y Edge en sus últimas dos versiones."

**Requerimientos Técnicos:** Especificaciones técnicas necesarias.

"El sistema se desarrollará utilizando el lenguaje Ionic Y framework Angular, Backend Node.js, Express.js y WebSockets."

"La base de datos a utilizar será MySQL."

"Se integrará con el servicio de mapas para el monitoreo en tiempo real."

**Reglas de Negocio:**

El documento de **Requisitos del Sistema** será la principal fuente de referencia para todo el equipo de desarrollo y para los stakeholders a lo largo del proyecto.

#### 3. Aprobación de los Requerimientos

Para asegurar que todos los requisitos sean comprendidos y aceptados por los interesados, se llevará a cabo una **revisión de los requerimientos**:

* **Reuniones de validación**: Con los stakeholders y usuarios finales para confirmar que todos los requerimientos se ajustan a sus expectativas.
* **Documentación firmada**: Los requisitos aprobados serán firmados por los stakeholders clave, asegurando que el alcance del proyecto esté definido y validado antes de iniciar el desarrollo.

#### 

#### 5. Herramientas y Técnicas para la Gestión de Requerimientos

Se utilizarán las siguientes herramientas y técnicas para gestionar los requisitos de manera eficiente:

* **Software de modelado de procesos**: Como **StarUML** para visualizar y documentar procesos relacionados con la gestión de flota.

#### 7. Responsabilidades en la Gestión de Requerimientos

* **Gerente de Proyecto**: Responsable de asegurar que el proceso de gestión de requerimientos sea seguido correctamente, aprobando los cambios en los requisitos y asegurando que el equipo cumpla con las expectativas de los stakeholders.
* **Analista de Requerimientos**: Encargado de trabajar directamente con los stakeholders para identificar, documentar y gestionar los requerimientos del sistema.
* **Equipo de Desarrollo**: Responsable de implementar los requisitos definidos en el sistema y de informar sobre cualquier desafío o cambio necesario en los requisitos técnicos.

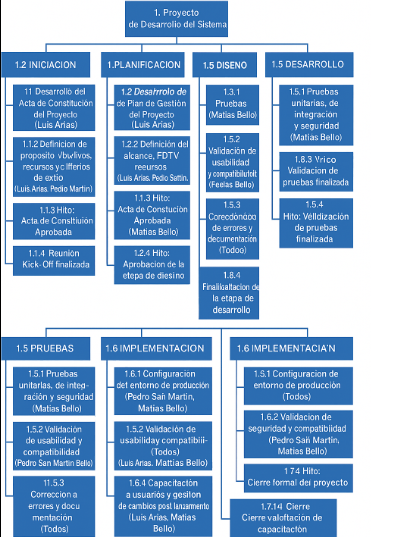
**Plan de gestión del cronograma**

#### 1. Objetivo del Plan de Gestión del Cronograma

El objetivo de este plan es establecer un marco claro para la creación, ejecución, monitoreo y control del cronograma del **Sistema de Control de Flota**. El propósito es garantizar que todas las actividades del proyecto se completen en el tiempo esperado y dentro de los plazos acordados con los stakeholders.

#### 2. Definición de Actividades y Hitos

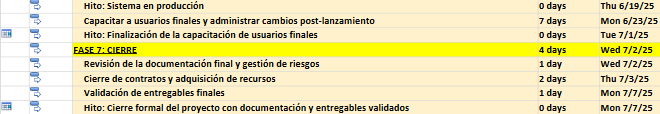
Para gestionar correctamente el cronograma, primero se deben definir todas las actividades necesarias para completar el proyecto. Las actividades se derivan de la **Estructura de Desglose del Trabajo (EDT)**. Algunos hitos clave del proyecto incluyen

* ****

#### 5. Desarrollo del Cronograma

Con las actividades identificadas, las secuencias y las estimaciones de duración, se utilizará una herramienta de gestión de proyectos (como **Microsoft Project**, **Jira**, o **Asana**) para desarrollar el cronograma. En el cronograma se deben identificar:

****

****

#### 9. Responsabilidades en la Gestión del Cronograma

* **Gerente de Proyecto - Luis Arias**: Responsable de la aprobación del cronograma, control de cambios y monitoreo del progreso del proyecto.
* **Equipo de Desarrollo - Pedro San Martin - Matias Bello**: Encargado de ejecutar las actividades según el cronograma y reportar avances o retrasos.
* **Stakeholders**: Participan en la revisión de avances y aprueban las fechas y entregas clave.

**Plan de gestión de calidad**

### 1. Objetivo del Plan de Gestión de Calidad

Este Plan de Gestión de Calidad define el enfoque, los estándares, las actividades y las responsabilidades para asegurar que tanto el proceso de desarrollo como el producto final del "Sistema de Control de Flota" cumplan con los requisitos funcionales y no funcionales definidos, así como con las expectativas de calidad y los objetivos del proyecto Capstone. Se busca entregar un software robusto, fiable, usable y mantenible, siguiendo las buenas prácticas de la ingeniería de software dentro del marco de la metodología Cascada.

**1. Objetivos de Calidad del Proyecto:**

**Funcionalidad Completa:** Asegurar que las 11 funcionalidades clave definidas en la Especificación de Requisitos de Software (ERS) sean implementadas correctamente y operen según lo especificado.

**Fiabilidad:** Minimizar la cantidad de defectos críticos y mayores en el software entregado. Objetivo: Menos 5 defectos críticos encontrados durante las Pruebas de Aceptación de Usuario (UAT).

**Usabilidad:** Lograr que el sistema sea intuitivo y fácil de usar para los roles definidos (Administrador, Contratista, Prevencionista). Objetivo: Que el 85% de los usuarios (simulados o reales en UAT) puedan completar tareas clave definidas en menos de [Tiempo, ej., 10 minutos] tras una breve capacitación (vinculado al objetivo general del proyecto).

**Rendimiento (NFR Clave):** Cumplir con los requisitos de rendimiento especificados en la ERS para operaciones críticas (ej., carga del dashboard < 5s, actualización de monitoreo en tiempo real < 60s).

**Seguridad (NFR Clave):** Asegurar que los roles y permisos se apliquen correctamente y que el sistema esté protegido contra vulnerabilidades básicas.

**Mantenibilidad:** Producir código fuente claro, bien documentado y siguiendo estándares, facilitando futuras correcciones o mejoras.

**Calidad de la Documentación:** Entregar documentación técnica y de usuario completa, clara y precisa (ERS, Diseño, Manuales, Informe Final).

### 4. Técnicas de Gestión de Calidad

Se utilizarán las siguientes **técnicas** para asegurar la calidad a lo largo del ciclo de vida del proyecto:

* **Pruebas de calidad**:  
  + **Pruebas unitarias**: Se ejecutarán pruebas unitarias en cada módulo de código para verificar su correcto funcionamiento.
  + **Pruebas de integración**: Se realizarán pruebas para verificar que todos los componentes del sistema funcionen correctamente juntos.
  + **Pruebas de aceptación del usuario (UAT)**: Los usuarios finales validarán que el sistema cumpla con sus expectativas y requisitos.
* **Revisión de código**: El código será revisado por miembros del equipo para detectar posibles errores y asegurar que cumple con los estándares de calidad definidos.
* **Gestión de defectos**: Se implementará un sistema para gestionar los defectos encontrados durante las pruebas, con una revisión constante y seguimiento hasta su resolución

**4. Aseguramiento de la Calidad (QA - Quality Assurance):**

Actividades enfocadas en el proceso para prevenir defectos:

**Revisiones Técnicas Formales:**

**Revisión de Requerimientos (ERS):** Al final de la fase de Análisis.

**Revisión de Diseño (Arquitectura, BD, UI):** Al final de la fase de Diseño.

**Revisión de Código (Code Reviews):** Realizadas por pares dentro del equipo para módulos críticos o complejos durante la fase de Implementación. Se usarán checklists básicos.

**Revisión del Plan de Pruebas:** Antes de iniciar la fase formal de Pruebas.

**Revisión de Documentación:** Para manuales e informe final antes de la entrega.

**5. Control de Calidad (QC - Quality Control):**

Actividades enfocadas en el producto para identificar defectos:

**Pruebas Unitarias:** Realizadas por el desarrollador que escribe el código para verificar componentes individuales (funciones, clases). Fase: Implementación.

**Pruebas de Integración:** Verificar la correcta interacción entre módulos o componentes desarrollados. Fase: Implementación (hacia el final) / Pruebas.

**Pruebas de Sistema:** Pruebas funcionales (End-to-End) basadas en los casos de uso y requisitos de la ERS para verificar el sistema completo. Pruebas no funcionales (rendimiento, seguridad básica). Fase: Pruebas.

**Pruebas de Regresión:** Re-ejecución de pruebas relevantes tras corregir defectos o realizar cambios para asegurar que no se introdujeron nuevos problemas. Fase: Pruebas / Mantenimiento (si aplica).

**Pruebas de Aceptación de Usuario (UAT):** Ejecutadas por usuarios finales (o simuladas por el equipo/stakeholders) basadas en escenarios del mundo real para validar que el sistema cumple las necesidades del negocio y los criterios de aceptación. Fase: Pruebas (final) / Despliegue (inicio).

### 7. Roles y Responsabilidades en la Gestión de Calidad

Los roles clave para la gestión de la calidad incluyen:

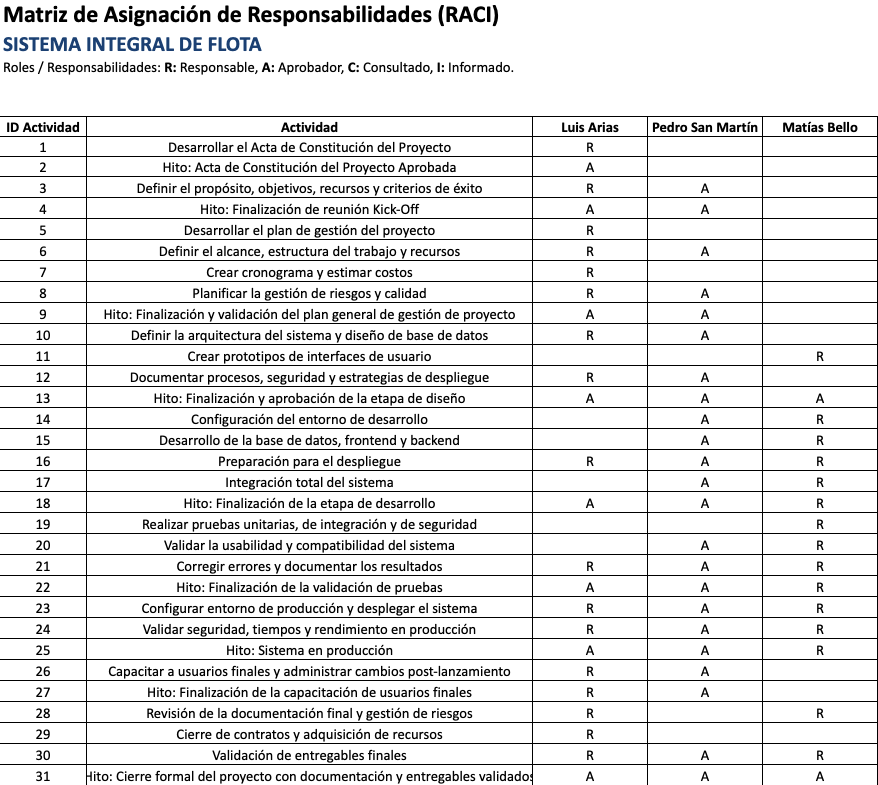
* **Gerente de Proyecto**: Luis Arias  
  + Supervisará el cumplimiento de los estándares de calidad y la ejecución del plan de calidad.
  + Asegurará la correcta gestión de los recursos para las actividades de calidad.
* **Equipo de Desarrollo**: Matias Bello  
  + Es responsable de implementar las funcionalidades del sistema siguiendo las mejores prácticas de desarrollo de software y asegurando que el código esté libre de errores.
* **Equipo de QA**: Pedro San Martin
  + Se encargará de diseñar y ejecutar las pruebas necesarias para verificar la calidad del sistema, incluyendo pruebas unitarias, de integración y de aceptación del usuario.
* **Stakeholders**:  
  + Serán responsables de validar el cumplimiento de los requisitos de calidad y aprobar los entregables conforme a los criterios establecidos.

### 8. Herramientas y Técnicas de Calidad

Se utilizarán las siguientes herramientas y técnicas para la gestión de calidad:

* **Herramientas de pruebas automáticas**: Para la ejecución de pruebas unitarias e integración (por ejemplo, Selenium).
* **Control de versiones**: Para gestionar el código fuente y los cambios, utilizando herramientas como Git, Github Desktop

**Plan de gestión de recursos**



**Plan de gestión de comunicaciones**

| Tipo de Comunicación | Descripción / Propósito | Frecuencia | Canal/Método Principal | Audiencia Principal | Responsable |
| --- | --- | --- | --- | --- | --- |
| Reunión Semanal de Equipo | Seguimiento interno del progreso de las tareas de la fase actual, identificación de problemas, coordinación. | Semanal | Videoconferencia/Presencial (30-45 min) | Equipo Capstone | Líder/PM |
| Comunicación Asíncrona Interna | Consultas técnicas, compartir información específica, avisos puntuales. | Según necesidad / Diaria | Chat Grupal [Herramienta: Discord | Equipo Capstone | Todo el Equipo |
| Gestión de Tareas y Progreso | Seguimiento del avance en las tareas definidas para cada fase en la EDT. | Continua | Herramienta Gestión Proyectos ([Herramienta: MsProyect]) | Equipo Capstone | Todo el Equipo |
| Control de Versiones y Código | Versionar entregables de código y documentación al final de cada fase. | Hitos de Fase / Continuo | Repositorio Git ([Herramienta: GitHub/GitLab]) | Equipo Capstone | Todo el Equipo |
| Documentación Compartida | Almacenamiento y acceso a documentos del proyecto (ERS, Diseño, Plan Pruebas, Manuales, etc.). | Continuo / Hitos de Fase | Plataforma Docs ([Herramienta: Google Drive -) | Equipo Capstone, Stakeholders | Integrante asignado |
| Revisión Formal de Fase | (Phase-Gate Review) Presentación de entregables de la fase, demo (si aplica), revisión de cumplimiento, decisión Go/No-Go para la siguiente fase. | Al final de cada fase (Análisis, Diseño, Implementación, Pruebas) | Reunión Formal (Presencial/) | Stakeholders  , Equipo | Líder/PM (Coordina) |
| Informe de Estado (a Profesores) | Reporte formal del estado del proyecto, usualmente previo a la Revisión de Fase. | Previo a Revisiones de Fase | Email / Documento Formal (PDF) | Stakeholders | Líder/PM |
| Entrega Formal de Fase | Entrega de los artefactos/documentos/código correspondientes a la fase completada. | Al final de cada fase | Plataforma U / Email | Stakeholders | Líder/PM |
| Alertas y Escalado | Comunicación urgente sobre problemas críticos que impiden el avance de la fase actual. | Según necesidad | Email / Mensaje Directo / | Stakeholders | Líder/PM |