

Plan de Gestión de la Calidad del Proyecto

Versión 1.0 ● 08/10/2025

Mis Viáticos

| Nombre del Proyecto | Mis Viáticos |
| --- | --- |
| Preparado Por | Daniel Iturra |
| Fecha (mm/dd/yyyy) | 08/10/2025 |

| Versión | Fecha (dd/mm/yyyy) | Comentarios |
| --- | --- | --- |
| 1.0 | 08/10/2025 | Sin comentarios |
|  |  |  |

**Plan de Gestión de Calidad del Proyecto**

| 1. Expectativas de la Calidad del Proyecto de acuerdo al cliente |
| --- |
| El proyecto Mis Viáticos busca alcanzar un nivel de calidad técnica y funcional que garantice eficiencia, confiabilidad, usabilidad y seguridad, pilares fundamentales para su implementación en entornos empresariales reales.  El cliente espera una plataforma que permita gestionar todo el ciclo de vida de los viáticos: desde el registro de gastos por parte del empleado, la lectura automatizada de comprobantes mediante OCR, hasta la aprobación y consolidación de reportes financieros.  Las expectativas de calidad se centran en cuatro dimensiones críticas:   * Confiabilidad del sistema: El backend desarrollado en Go + Fiber y desplegado sobre infraestructura Cloud escalable (PostgreSQL + Redis + Docker + Kubernetes) debe mantener un uptime superior al 99,9%. * Precisión del módulo OCR: El componente de reconocimiento de texto, integrado con Google Vision API y Tesseract.js, debe lograr al menos 85% de precisión en lectura de montos, fechas y RUTs, permitiendo la validación manual cuando sea necesario. * Calidad de la interfaz de usuario: El frontend, construido con React + TypeScript + Zustand + React Query, debe ofrecer una experiencia fluida, accesible y moderna, con tiempos de carga inferiores a 2 segundos y diseño completamente responsive, según las guías de UX definidas en el sprint 7. * Cumplimiento normativo y seguridad: La aplicación debe cumplir con buenas prácticas OWASP, cifrado de JWT, validación de entrada estricta, protección CSRF y encriptación de datos sensibles.   Además, el cliente requiere que los procesos de rendición y aprobación se ejecuten sin errores lógicos, garantizando trazabilidad total y reportes consolidados en formatos Excel, PDF y visualizaciones gráficas generadas por el módulo de analytics |

| 1. Identificación de los Interesados del Proyecto |
| --- |
| Los principales interesados que influyen directa o indirectamente en la calidad del proyecto son:   * Transportes del Sur Ltda.  Cliente principal que encarga la solución para modernizar su proceso de control de gastos. Supervisa los hitos y valida los entregables funcionales. * Equipo de Desarrollo Mis Viáticos (DUOC UC)    + Daniel Iturra – Líder Técnico y Gerente de Proyecto   + Mario Bronchuer – Desarrollador Frontend y responsable de UX/UI  El equipo es responsable de cumplir los estándares técnicos, las prácticas de documentación y la planificación por sprints definida en el backlog. * Usuarios Finales (Empleados, Aprobadores y Administradores)  Representan los roles funcionales que interactúan con la plataforma. Su retroalimentación se usa para medir la usabilidad, rendimiento y precisión de los flujos. * Docente Supervisor y Comité Evaluador DUOC UC  Monitorean la correcta aplicación de los procesos de gestión de calidad, aseguramiento de métricas y cumplimiento del modelo de proyecto APT. |

| 1. Alcance Inicial del Proyecto |
| --- |
| El alcance del proyecto se define en seis componentes funcionales principales, cada uno con criterios de calidad específicos:   1. Módulo de Gastos y OCR:  Permite registrar y digitalizar comprobantes mediante reconocimiento óptico (Google Vision + Tesseract.js). Los datos extraídos se validan, formatean y almacenan en la base de datos PostgreSQL.  El OCR incorpora preprocesamiento de imagen, corrección de contraste y eliminación de ruido para optimizar resultados. 2. Módulo de Políticas y Validaciones:  Implementa un motor de reglas (RuleEngine) capaz de evaluar montos, categorías, límites diarios/mensuales y definir aprobaciones automáticas o requeridas según políticas configuradas. 3. Módulo de Aprobaciones:  Desarrollado con una estructura de flujos multinivel, garantiza la trazabilidad y registro histórico de decisiones mediante un WorkflowEngine y almacenamiento de auditorías (ApprovalHistory). 4. Módulo de Notificaciones:  Basado en Socket.IO y RabbitMQ, gestiona alertas en tiempo real, notificaciones push (Firebase Cloud Messaging) y eventos asincrónicos con reintentos automáticos. 5. Módulo de Reportes y Analytics:  Proporciona métricas interactivas mediante Recharts, exportación de datos con xlsx y jsPDF, e integración con ClickHouse para análisis columnar de gastos, top de categorías y cumplimiento de políticas. 6. Módulo de UX, PWA y Testing:  Incluye optimizaciones visuales (Framer Motion), soporte offline (Service Workers) y validaciones automáticas (Jest, Playwright, GitHub Actions) que aseguran cobertura de pruebas superior al 80%. |

| 1. Línea Base del cronograma |
| --- |
| El proyecto se ejecuta a lo largo de 8 fases técnicas (según los backlogs de frontend y backend) distribuidas en 25 sprints semanales (~6 meses):   1. Fase 1: Fundamentos de Viáticos (instalación, modelos de datos, API base) 2. Fase 2: OCR Integration (Vision API + Tesseract.js + procesamiento local) 3. Fase 3: Políticas y Validación (motor de reglas, alertas visuales) 4. Fase 4: Flujos de Aprobación (reportes, jerarquías, historial de auditoría) 5. Fase 5: Notificaciones (WebSocket, Firebase, RabbitMQ) 6. Fase 6: Reportes y Analytics (gráficos, exportaciones, métricas) 7. Fase 7: UX/UI y Optimizaciones (Dark Mode, PWA, virtualización, Skeletons) 8. Fase 8: Testing y Deploy (Jest, Playwright, CI/CD con GitHub Actions y Vercel)   Cada fase tiene entregables medibles y criterios de aceptación definidos en el backlog técnico, asegurando control total de avances y calidad. |

| 1. Registro de Riesgos |
| --- |
| | Riesgo | Probabilidad | Impacto | Estrategia de Mitigación | Propietario | Plazo | Indicadores clave | | --- | --- | --- | --- | --- | --- | --- | | Fallo en infraestructura Cloud | Media | Alto | Backups automáticos, monitoreo Prometheus, alta disponibilidad | Daniel Iturra | Implementación | Uptime ≥ 99.9% | | Errores en OCR o baja precisión | Media | Alto | Entrenamiento con dataset real y fallback local | Equipo Backend | Fase 2 | Precisión ≥ 85% | | Retrasos en fases dependientes del backend | Alta | Medio | Planificación de issues con etiquetas backend-dependency y control de merges | Daniel Iturra | Durante sprints | Cumplimiento ≥ 95% | | Fallos en flujos de aprobación | Media | Alto | Pruebas unitarias con casos simulados y validaciones E2E | Equipo QA | Fase 4 | Tasa de error < 1% | | Vulnerabilidades OWASP | Baja | Alto | Auditorías automáticas con CI/CD y revisión de dependencias | Daniel Iturra | QA Final | 0 vulnerabilidades críticas | | Baja usabilidad en frontend | Media | Medio | Feedback iterativo de usuarios y pruebas A/B en UI | Mario Bronchuer | Fase 7 | Satisfacción ≥ 90% | |

| 1. KPI Iniciales de Proyecto (Métricas de Calidad) |
| --- |
| Disponibilidad de la Infraestructura Cloud   * Esperado: 99.9% uptime * Frecuencia: Monitoreo continuo * Umbral: <0.1% de inactividad mensual   Precisión del OCR   * Esperado: ≥ 85% de reconocimiento correcto * Frecuencia: Validación semanal con muestras * Umbral: <15% de corrección manual   Tasa de Error en Flujos de Aprobación y Reportes   * Esperado: <1% de errores críticos * Frecuencia: QA en cada sprint * Umbral: ≤5 errores por 10.000 solicitudes   Cumplimiento del Cronograma por Sprint   * Esperado: ≥95% de issues completadas a tiempo * Frecuencia: Revisión quincenal * Umbral: ≤5% de desviación   Rendimiento del Frontend (UX/UI)   * Esperado: Carga <2 segundos * Frecuencia: Pruebas diarias durante QA * Umbral: 95% de vistas bajo 5 segundos   Cobertura de Testing   * Esperado: 80%+ cobertura Jest/Playwright * Frecuencia: En cada pipeline CI/CD * Umbral: 0 tests fallidos antes del deploy   Satisfacción del Usuario Final   * Esperado: ≥90% de satisfacción (encuesta postmarcha blanca) * Frecuencia: Al cierre de proyecto * Umbral: Promedio ≥4.5/5 en usabilidad y velocidad |

.