



UNIVERSIDAD DEL VALLE

FACULTAD DE INGENIERÍA

Gestión de riesgos en proyectos de software

AUTORES:

Manuel Alexander Serna Jaraba - 202259345

Santiago Villa Salazar - 202259527

Adrian Felipe Velasquez Arias - 202259456

Edgar Fabian Rueda Colonia - 2259606

Kevin Alejandro Marulanda Escobar - 2380697

DOCENTE:

SALAZAR VICTORIA, Alvaro Hernando

CURSO:

Introducción a la Gestión de Proyectos de Software

Tuluá, Valle del Cauca

Noviembre de 2025

Actividad A: Lista inicial de Riesgos

ID	Categoría	Causa	Riesgo	Objetivo afectado	Impacto (medida)	Disparador	Dueño
R-01	Prov./Infra.	Límites de cuota en Firebase Auth (p.ej., SMS, email).	Bloqueo de usuarios durante registro o recuperación (RF-01)	Funcionalidad	>50 usuarios no pueden registrarse.	Log de "Quota Exceeded" en Firebase.	Santiago Villa
R-02	RRHH / Téc.	Poca experiencia del equipo en Jetpack Compose ⁹ .	Desarrollo de UI lento o con bajo rendimiento (afecta RNF-02)	Cronograma / Calidad	Retraso de +10 días en Sprints de UI.	Tareas de UI > 20% sobreestimadas.	Kevin Marulanda
R-03	Calidad / CI	Configuración estricta de SonarCloud (maxIssues: 0)	Bloqueo de PRs por issues menores (calidad de código).	Cronograma	>2 días para mergear un PR simple.	PR bloqueado por Sonar	Adrian Velasquez
R-04	Requisitos	Ambigüedad en RF-04 (NEQUI) vs Alcance (sin pasarela real)	Re-trabajo en el módulo de checkout.	Alcance / Cronograma	+5 días de retrabajo.	Reunión de validación de Sprint.	Manuel Serna
R-05	Datos / Téc.	Diseño de esquema de Firestore	Consultas lentas al cargar historial (RF-06)	Rendimiento (RNF-02)	Carga de catálogo > 2s (falla RNF-02)	Pruebas de carga (JMeter)	Santiago Villa

R-06	Seguridad	Manejo incorrecto de secretos en GitHub Actions	Exposición de SONAR_TO KEN	Seguridad (RNF-04)	Acceso no autorizado a la BD.	Reporte de auditoría de seguridad.	Adrian Velasquez
R-07	Arquitectura	No se usa Inyección de Dependencias (Hilt/Koin)	Alta acoplamiento y dificultad para realizar tests unitarios	Calidad / Mantenibilidad	Cobertura de tests < 30%.	Reporte de Sonar (cobertura)	Kevin Marulanda
R-08	QA	Baja cobertura de tests unitarios (no garantizada)	Introducción de regresiones en features (p.ej. Carrito RF-03)	Calidad	>5 bugs críticos en producción.	Reporte de bug post-deploy.	Edgar Rueda
R-09	Requisitos	Baja disponibilidad de stakeholders (estudiantes/cafétería)	Retraso en la validación de prototipos/flujos (JAD)	Cronograma	+7 días por sprint.	Stakeholder no responde en 72h.	Manuel Serna
R-10	Prov./Costo	Carga de imágenes (RF-09) de alta resolución en Firebase Storage	Costos de almacenamiento y transferencia (egress) exceden el presupuesto.	Presupuesto	+15% costo de Firebase.	Alerta de presupuesto Firebase.	Santiago Villa

Actividad B: Ubicación y Priorización

Priorización cualitativa usando las escalas de probabilidad (a-e) e impacto (1-5) definidas en el codelab.

ID	Categoría	Riesgo	Prob. (a-e)	Impacto (1-5)	Prioridad
R-01	IA/API	Falla en API de Gemini	c	4	Alto
R-02	RRHH / Téc.	Poca experiencia en Compose	d	3	Alto
R-03	Calidad / CI	Bloqueo de PRs por Sonar	d	2	Moderado
R-04	Requisitos	Ambigüedad en Checkout (RF-04)	c	3	Moderado
R-05	Datos / Téc.	Consultas lentas en Firestore	d	3	Alto
R-06	Seguridad	Exposición de secretos (CI)	b	5	Moderado
R-07	Arquitectura	Alta acoplamiento (No DI)	e	2	Moderado
R-08	QA	Baja cobertura de pruebas	e	4	Alto
R-09	Requisitos	Retraso en validación (Stakeholder)	c	3	Moderado
R-10	Prov./Costo	Costos elevados en Firebase Storage	b	2	Bajo

Actividad C: EMV (Valor Monetario Esperado)

Análisis cuantitativo ligero para los riesgos de prioridad alta.

Riesgo (Amenaza)	Prob.	Impacto Costo (USD)	E(Costo)	Impacto Tiempo (días)	E(Tiempo]
R-01: Bloqueo de cuota de Firebase	0.40 (c)	+\$1,500	\$600	+22	8.8
R-02: Poca experiencia en Compose	0.60 (d)	+\$2,000	\$1,200	+14	8.4
R-05: Consultas lentas en Firestore	0.50 (d)	+\$1,800	\$900	+10	5.0
R-08: Regresiones por baja cobertura	0.80 (e)	+\$2,500	\$2,000	+12	9.6
TOTALES ESPERADOS			\$4,700		31.8 días

Reserva de contingencia sugerida: Se recomienda una reserva de contingencia de **\$4,700 USD** y **32 días** para gestionar los riesgos conocidos.

Actividad D: PERT (Estimación por Tres Valores)

Estimación PERT para una tarea crítica afectada por un riesgo alto (R-02: Poca experiencia en Compose).

- **Tarea Crítica:** Implementación UI del Catálogo de Productos (RF-02).
- **O (Optimista):** 3 días (Si Compose fluye perfectamente).
- **M (Más probable):** 8 días (Desarrollo normal).
- **P (Pesimista):** 20 días (Si hay bloqueos significativos por la curva de aprendizaje de Compose).

Cálculos:

- $TE = (O + 4M + P) / 6$
 $TE = (3 + 4*8 + 20) / 6 = (55) / 6 \approx 9.17 \text{ días}$
- $\sigma = (P - O) / 6$
 $\sigma = (20 - 3) / 6 = (17) / 6 \approx 2.83 \text{ días}$

Comunicación de la estimación:

- **50% confianza:** La tarea tomará **9.17 días** (TE).
- **84% confianza:** La tarea tomará **12.0 días** ($TE + 1\sigma$).
- **97.5% confianza:** La tarea tomará **14.83 días** ($TE + 2\sigma$).

Actividad E: Plan de respuesta a riesgos

Plan de respuesta para los riesgos priorizados (R-01, R-02, R-05, R-08), siguiendo el formato del codelab

ID Riesgo (Causa → Riesgo → Impacto)	Estrategia	Acciones	Disparador	Dueño	Fecha	Presupuesto (USD)
R-01 Límites de Firebase Auth → Bloqueo de registros (RF-01) → >50 usuarios afectados.	Mitigar	Implementar monitoreo de cuotas de Firebase. Usar re-intentos (exponential backoff) en el cliente Android.	Log de "Quota Exceeded" en consola Firebase.	Santiago Villa	10/11/2025	\$600
R-02 Poca experiencia en Compose ³¹ → Desarrollo UI lento → +14 días.	Mitigar	Realizar "spikes" o PoC (Prueba de Concepto) en las UI más complejas (Catálogo/Carrito). Asignar pares (pair programming).	Tareas de UI > 20% sobreestimadas en el Sprint Planning.	Kevin Marulanda	07/11/2025	\$1,200
R-05 Esquema BD no óptimo → Consultas lentas (falla RNF-02) → +10 días.	Evitar	Validar el diseño del esquema de Firestore con un experto (o PoC de carga) <i>antes</i> de implementar los repositorios (RF-02, RF-06).	Inicio del Sprint 1 (Diseño de Arquitectura).	Santiago Villa	06/11/2025	\$900

R-08 Baja cobertura de tests → Regresiones críticas (RF-03) → +12 días.	Aceptar (Activo)	Definir plan de contingencia (QA manual intensivo) antes de cada release. Priorizar tests unitarios <i>solo</i> en la lógica crítica (p.ej. Validators lógica del Carrito).	Fin de un Sprint (Code Freeze).	Edgar Rueda	15/11/2025	\$2,000
---	-------------------------	---	---------------------------------	-------------	------------	---------