

Universidad de los Andes

Proyecto Integrador

Grupo 4

Estrategia de Pruebas - MEDISUPPLY

1. Aplicación Bajo Pruebas

1.1. Nombre Aplicación: MediSupply

1.2. Versión: 1.0.0 (Prototipo Funcional)

1.3. Descripción:

MediSupply es una plataforma tecnológica integral compuesta por una aplicación web y una aplicación móvil, diseñada para optimizar la cadena de suministro de insumos médicos, medicamentos de alta complejidad y equipos biomédicos. La solución atiende a una empresa multinacional con presencia en Colombia, Perú, Ecuador y México, que distribuye productos a más de 2,000 instituciones de salud.

El sistema unifica los procesos de compras, inventarios, ventas y logística bajo una arquitectura modular de alta disponibilidad (7x24x365), soportando hasta 100 usuarios concurrentes por país y procesando hasta 400 pedidos por minuto en períodos de alta demanda. La plataforma integra módulos de seguridad avanzada, algoritmos de optimización de rutas, recomendación de productos y monitoreo en tiempo real de la cadena de frío.

1.4. Funcionalidades Core:

A continuación, se presentan las principales funcionalidades identificadas:

ID	Funcionalidad	Descripción
FU1	Gestión de Proveedores y Productos	Registrar y gestionar proveedores certificados, cargar productos de forma individual y masiva, incluyendo fichas técnicas, certificaciones sanitarias y condiciones de almacenamiento.
FU2	Gestión de Inventarios y Localización	Localizar productos en bodegas en ≤ 1 segundo, controlar existencias en tiempo real, gestionar lotes, fechas de vencimiento y condiciones de cadena de frío.
FU3	Sistema de Pedidos y Ventas	Procesar pedidos institucionales con consulta de inventario en tiempo real (≤ 2 segundos), gestionar el ciclo completo desde creación hasta entrega.
FU4	Gestión de Rutas Logísticas	Generar rutas óptimas para vehículos de entrega en ≤ 3 segundos, monitorear ubicación de camiones en tiempo real con actualización cada 500ms.
FU5	Gestión de Clientes Institucionales	Administrar hospitales, clínicas y laboratorios, segmentar por especialidad médica, nivel de atención y ubicación geográfica.
FU6	Sistema de Usuarios y Seguridad	Controlar acceso por roles (compras, logística, ventas, dirección), autenticación multifactor, auditoría de transacciones y trazabilidad completa.

Universidad de los Andes

Proyecto Integrador

Grupo 4

FU7	Aplicación Móvil - Fuerza de Ventas	Consultar rutas de visita, registrar visitas con geolocalización, crear pedidos en línea, capturar evidencia visual y generar recomendaciones.
FU8	Aplicación Móvil - Clientes	Permitir registro de clientes en ≤ 1 segundo, creación de pedidos, consulta de estados y seguimiento de entregas programadas.
FU9	Algoritmos de Optimización	Ejecutar algoritmos de recomendación de compras, optimización de inventarios y sugerencias de productos basadas en patrones de consumo y eventos sanitarios.
FU10	Reportes y Analytics	Generar informes por comercial, producto y zona geográfica, análisis de ventas históricas y proyecciones de demanda.

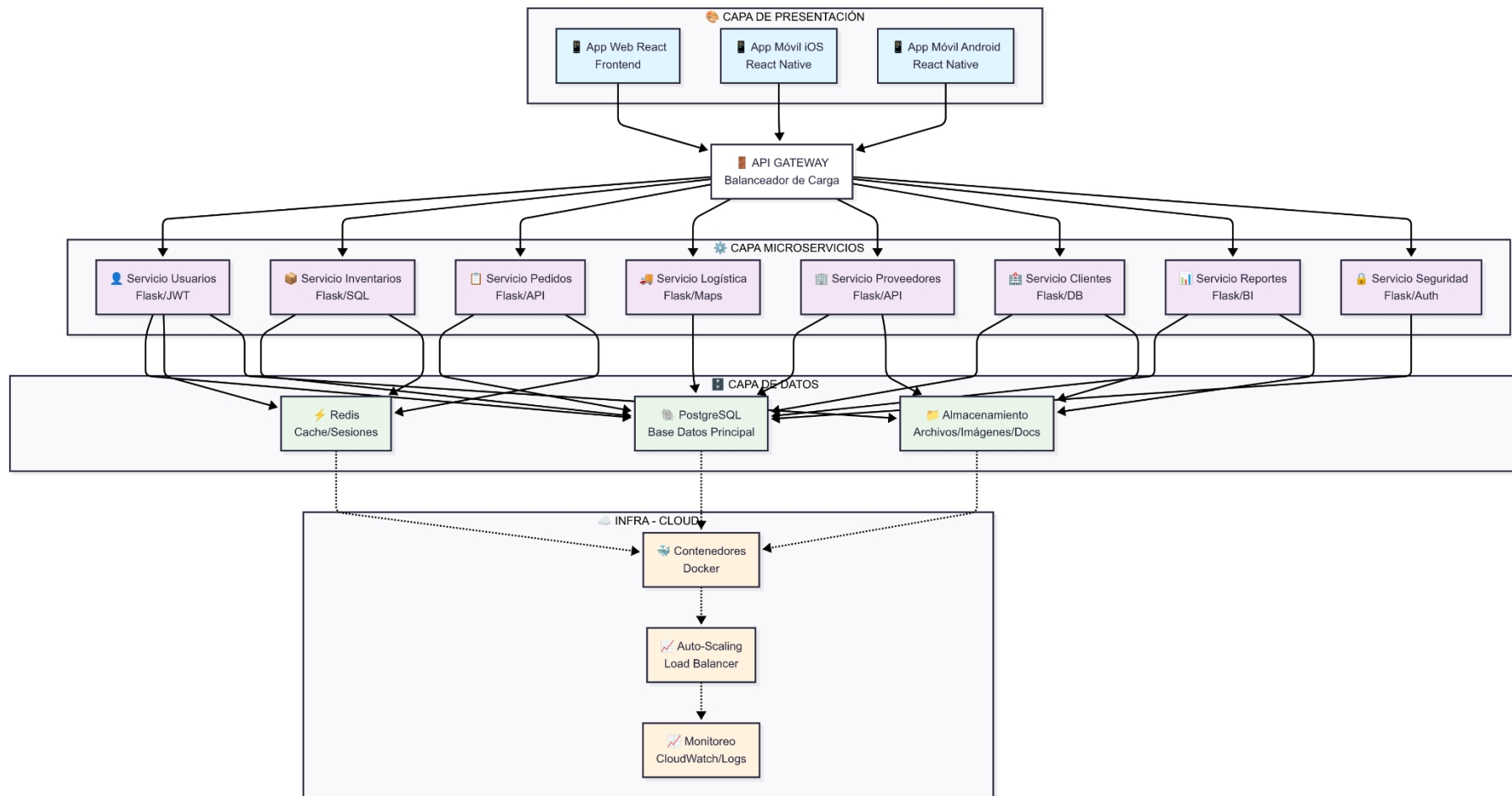
1.5. Diagrama de Arquitectura:

La arquitectura de MediSupply seguirá un patrón de microservicios con los siguientes componentes principales:

Universidad de los Andes

Proyecto Integrador

Grupo 4



<https://www.mermaidchart.com/app/projects/1667f9ca-dc87-4bb5-bfd7-a7cdaf6d4a06/diagrams/de3c9da2-f96e-4175-97da-779544159fab/version/v0.1/edit>

(Validar y/o Reemplazar)

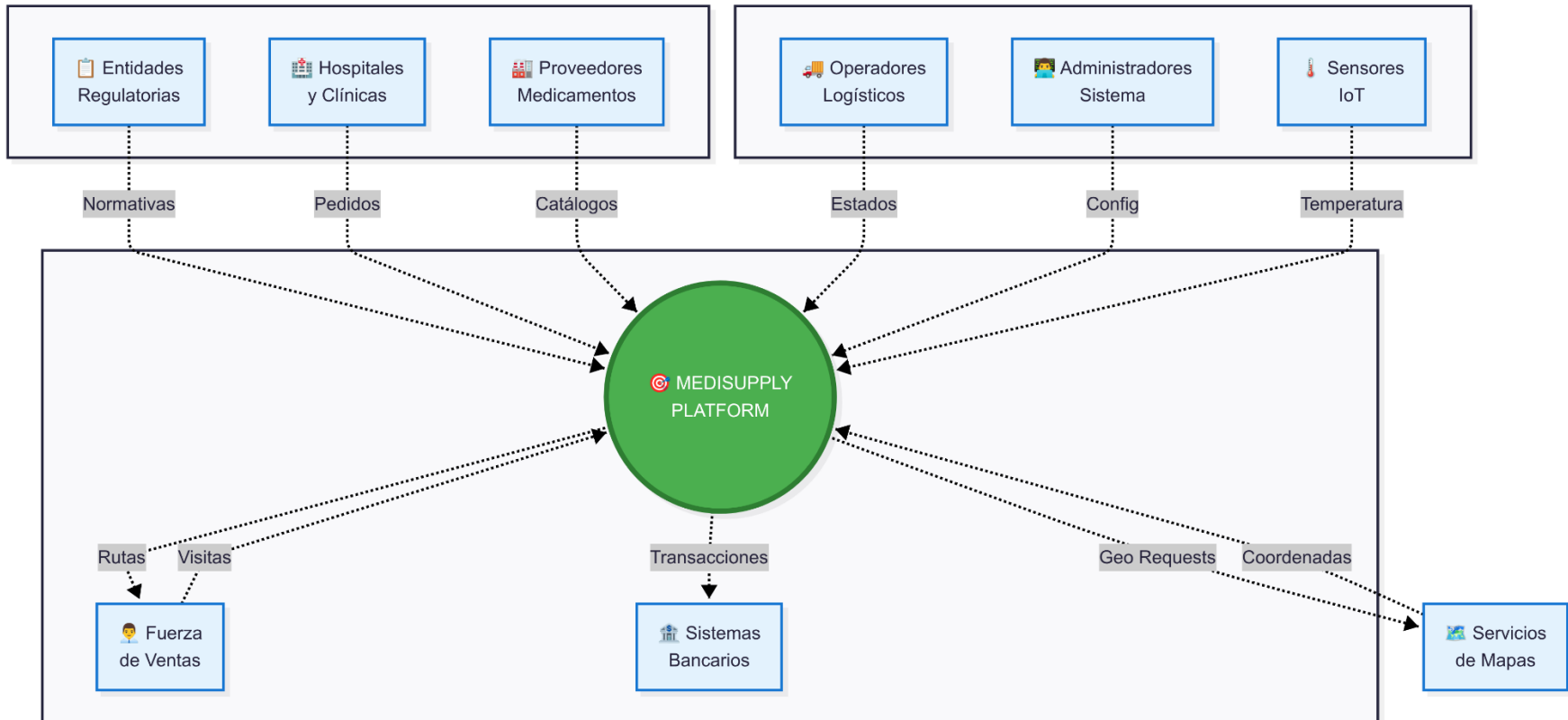
Universidad de los Andes

Proyecto Integrador

Grupo 4

1.6. Diagrama de Contexto

En este diagrama se definen los límites del sistema y mapea todas las interacciones externas críticas para la plataforma, con el fin de Definir todas las interfaces que requieren pruebas de conectividad, rendimiento y seguridad.



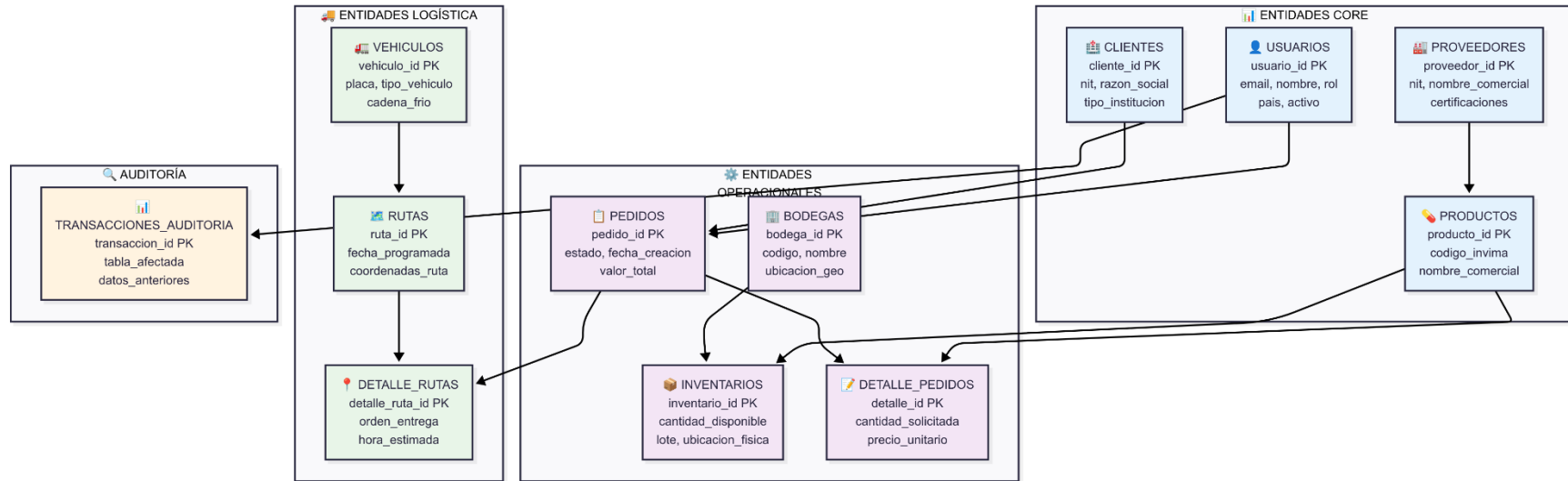
<https://www.mermaidchart.com/app/projects/1667f9ca-dc87-4bb5-bfd7-a7cdaf6d4a06/diagrams/31786914-3ef6-4c92-8a4a-ae49c930525e/version/v0.1/edit>

Universidad de los Andes

Proyecto Integrador

Grupo 4

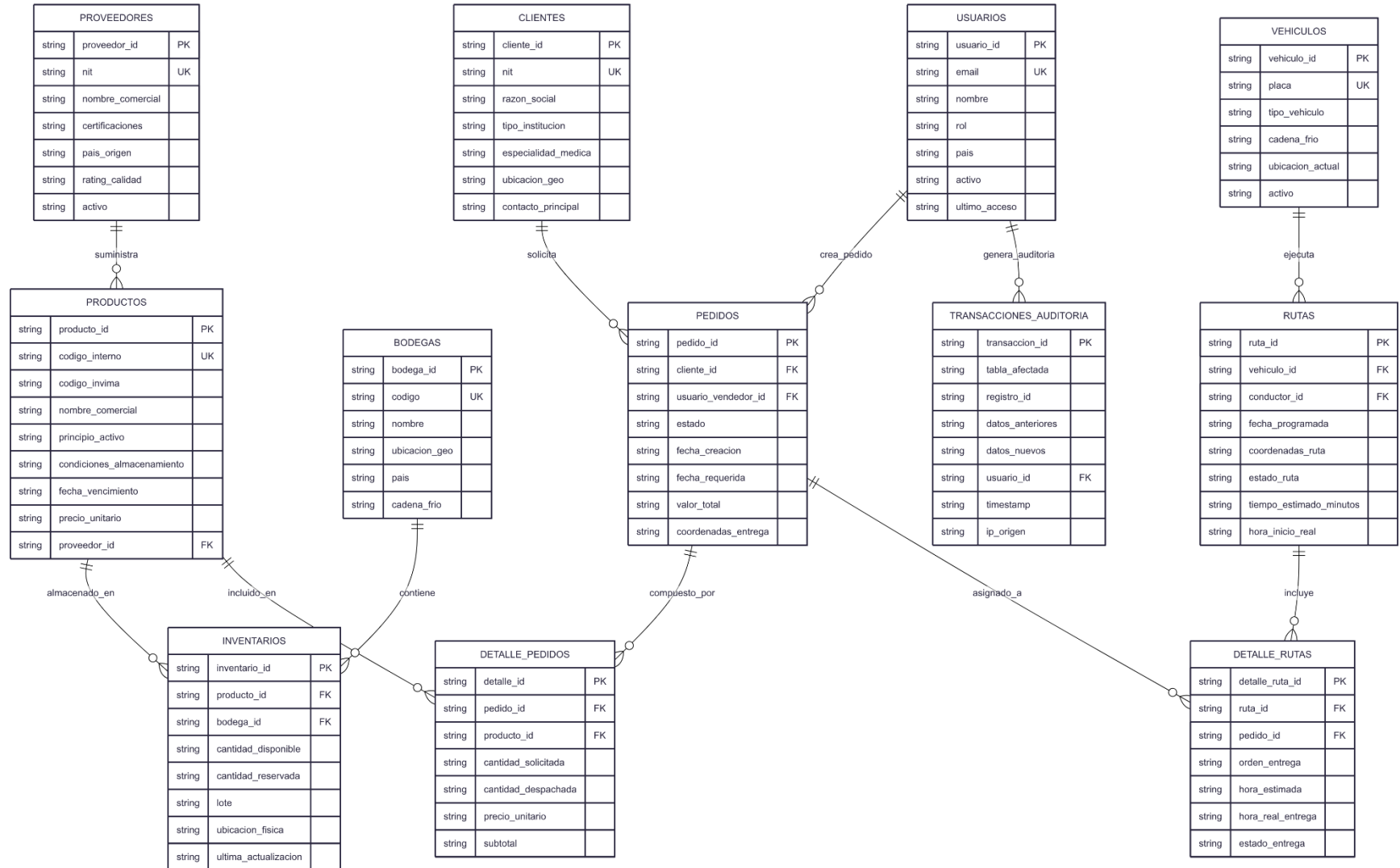
1.7. Modelo de datos



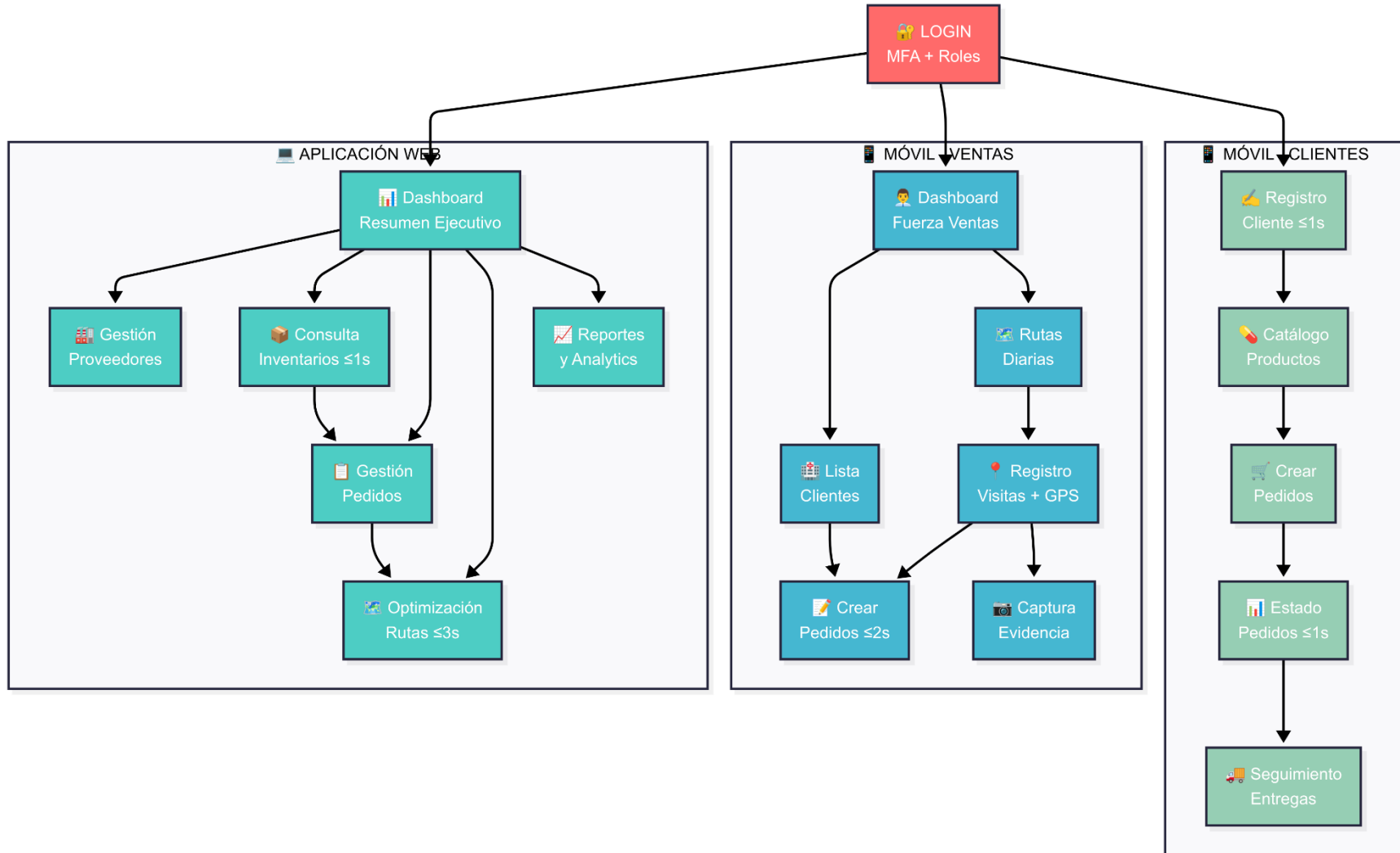
Universidad de los Andes

Proyecto Integrador

Grupo 4



1.7. Modelo de GUI



Universidad de los Andes

Proyecto Integrador

Grupo 4

2. Contexto de la Estrategia de Pruebas

2.1. Objetivos

Objetivos de la Estrategia de Pruebas

Validar la funcionalidad crítica de la cadena de suministro médica y cumplimiento de tiempos de respuesta

Durante el período de pruebas, se ejecutarán y documentarán todos los casos de prueba necesarios para cubrir el 100% de los escenarios funcionales críticos identificados en las 10 funcionalidades core (FU1-FU10). Esto incluye la gestión de proveedores y productos, consultas de inventario en tiempo real ($\leq 1s$), creación de pedidos con validación de stock ($\leq 2s$), optimización de rutas logísticas ($\leq 3s$), y el flujo completo de las aplicaciones móviles para fuerza de ventas y clientes institucionales. El objetivo se considerará cumplido cuando cada requerimiento funcional y no funcional esté vinculado al menos a un caso de prueba ejecutado y documentado, con evidencia de cumplimiento de los SLA de rendimiento establecidos antes del cierre del período de pruebas. Para garantizar la robustez del sistema, se aplicarán estrategias de generación de datos médicos realistas incluyendo catálogos de medicamentos, códigos INVIMA, y datos de instituciones de salud de los 4 países objetivo.

Asegurar la integridad y disponibilidad del sistema bajo condiciones de alta carga y falla

Se realizarán pruebas de carga, estrés e integración sobre los 9 sistemas externos críticos (hospitales, proveedores, bancos, regulatorias, logística, IoT, mapas, fuerza de ventas, administradores) para validar el cumplimiento del 99.95% de disponibilidad y el procesamiento de hasta 400 pedidos/minuto durante picos de demanda. El objetivo es identificar y resolver cuellos de botella antes de producción, de modo que durante el primer mes post-lanzamiento, el sistema mantenga los SLA establecidos y no experimente más de 6 horas de downtime acumulado para cumplir con los requisitos de disaster recovery.

Optimizar la cobertura de pruebas mediante automatización de APIs y pruebas end-to-end

Se automatizará al menos el 85% de los casos de prueba de regresión asociados a las funcionalidades críticas de inventarios, pedidos y rutas, implementando pruebas de contrato para las 8 APIs de microservicios identificadas. Se establecerán pipelines de CI/CD con pruebas automatizadas cross-platform (web/móvil) y validación de integraciones en tiempo real. El objetivo es lograr una reducción del 40% en el tiempo total de ejecución de las pruebas de regresión mientras se mantiene cobertura completa de los escenarios críticos de la cadena de suministro médica antes de cada entrega sprint.

Universidad de los Andes

Proyecto Integrador

Grupo 4

2.2. Duración de la Iteración de Pruebas

Se llevarán a cabo cuatro iteraciones, cada una con una duración de dos semanas, dentro del período total de 8 semanas de desarrollo. Todas las actividades estarán enfocadas en pruebas automatizadas con datos médicos realistas, integrando generación de datos según los siguientes enfoques:

- En la fase de configuración inicial y diseño de casos de prueba (Iteración 1): Se definirán y prepararán los datasets a-priori con catálogos de medicamentos reales, códigos regulatorios (INVIMA, DIGEMID, COFEPRIS), así como los esquemas y scripts para la generación de datos de hospitales, proveedores certificados y rutas logísticas de los 4 países objetivo.
- Durante la implementación de funcionalidades core (Iteraciones 2 y 3): Se aplicarán las estrategias de generación de datos en la automatización de pruebas de inventarios, pedidos y logística, asegurando diversidad geográfica y realismo en los datos de instituciones de salud, condiciones de cadena de frío y escenarios regulatorios.
- En la consolidación y pruebas de integración final (Iteración 4): Se revisará la cobertura de pruebas end-to-end con sistemas externos, se ejecutarán pruebas de carga con 400 pedidos/minuto, y se validará el cumplimiento de todos los atributos de calidad (disponibilidad, latencia, seguridad, escalabilidad) antes del despliegue.

Fechas Generales del Proyecto:

- Duración total: 8 semanas de desarrollo y pruebas, divididas en 4 iteraciones de 2 semanas cada una

Iteración	Duración	Actividades Principales	Alcance (Funcionalidades)
1	Semana 1-2	Configuración de ambientes de prueba, diseño de casos automatizados, generación de datasets médicos realistas, configuración de herramientas CI/CD	Setup completo + FU1, FU6
2	Semana 3-4	Implementación y ejecución automatizada de pruebas para funcionalidades críticas de inventario y pedidos, validación de tiempos de respuesta $\leq 1-2s$	FU2, FU3, FU8
3	Semana 5-6	Implementación de pruebas de logística y aplicaciones móviles, pruebas de integración con sistemas externos, validación de geolocalización y rutas	FU4, FU7, FU9
4	Semana 7-8	Pruebas de carga (400 pedidos/min), pruebas end-to-end completas, validación de disponibilidad 99.95%, consolidación de reportes finales	FU5, FU10 + Validación integral

Este cronograma se ejecuta en paralelo con el desarrollo, siguiendo metodología **Agile con testing continuo**.

Universidad de los Andes

Proyecto Integrador

Grupo 4

2.3. Presupuesto de Pruebas

2.3.1. Recursos Humanos

Perfil del Ingeniero de Automatización QA - MediSupply

Capacidades requeridas:

- Desarrollar y mantener scripts de automatización para aplicaciones web React y aplicaciones móviles React Native usando herramientas como Cypress, Playwright, y/o Detox.
- Implementar pruebas de APIs REST para validar los 8 microservicios identificados (usuarios, inventarios, pedidos, logística, proveedores, clientes, reportes, seguridad).
- Configurar y ejecutar pruebas de carga con JMeter y/o Postman para validar el procesamiento de 400 pedidos/minuto y tiempos de respuesta críticos ($\leq 1-3$ segundos).
- Diseñar pruebas de integración con sistemas externos simulados (bancos, regulatorias, IoT, servicios de mapas, logística).
- Implementar validación de atributos de calidad: disponibilidad (99.95%), seguridad (MFA, cifrado), y escalabilidad horizontal.
- Trabajar con tecnologías cloud (AWS/Azure/GCP), contenedores Docker, y pipelines de CI/CD.
- Participar en la definición y ejecución de casos de prueba end-to-end que validen flujos completos de la cadena de suministro médica.

Aplicar estrategias de generación de datos médicos realistas utilizando:

- Pruebas con datos aleatorios: Generación dinámica de catálogos de medicamentos, códigos INVIMA/DIGEMID/COFEPRIS, datos de hospitales y proveedores certificados mediante Faker.js y bibliotecas médicas especializadas.
- Pruebas a-priori: Uso de datasets predefinidos de instituciones de salud reales de Colombia, Perú, Ecuador y México, productos farmacéuticos regulados, y rutas logísticas geográficas creados con Mockaroo.
- Pruebas pseudo-aleatorias: Combinación de datos geográficos reales con inventarios simulados, validación de cadena de frío, y escenarios regulatorios complejos para cubrir casos de borde específicos del sector salud.

Universidad de los Andes

Proyecto Integrador

Grupo 4

Experiencia recomendada: 1-2 años en roles similares con experiencia en sistemas de salud, APIs REST, pruebas móviles y herramientas de carga.

Tarifas de Mercado (2025):

Según ZipRecruiter y Salary.com, el salario promedio para un QA Automation Engineer especializado en sistemas críticos de salud en EE.UU. es de \$42/hora, con un rango típico de \$35 a \$52/hora para perfiles con experiencia en integración de sistemas complejos y pruebas de carga.

Cálculo de Costos:

Se propone una tarifa competitiva de \$38/hora para ingenieros de automatización, considerando la complejidad técnica del proyecto (microservicios, integración multilateral, requisitos regulatorios) y el contexto internacional para 2025.

Concepto	Cantidad	Horas Totales	Tarifa/Hora	Costo Total
Ingenieros Automatizadores QA	3	192 (8 sem × 24h/sem)	\$38	\$21,888
Lead QA Automation	1	96 (8 sem × 12h/sem)	\$45	\$4,320
Total Recursos Humanos	4	672	-	\$26,208

Justificación:

- La tarifa propuesta está alineada con las tendencias del mercado 2025 para sistemas críticos de salud que requieren alta disponibilidad y cumplimiento regulatorio.
- Se incluye experiencia en automatización móvil y APIs debido a la naturaleza híbrida del proyecto MediSupply.
- El equipo de 4 personas es apropiado para cubrir web + móvil + APIs + integraciones + carga durante las 8 semanas del proyecto.
- La inversión adicional se justifica por la complejidad técnica del sistema (9 integraciones externas, SLA estrictos, 4 países, regulaciones sanitarias) y la criticidad del negocio (cadena de suministro médica).

2.3.2. Recursos Computacionales

Infraestructura AWS EC2 para Ambientes de Prueba

Se utilizarán instancias t3.xlarge (4 vCPUs, 16 GiB RAM) y c5.2xlarge (8 vCPUs, 16 GiB RAM) para soportar la complejidad del testing de MediSupply: aplicaciones web, móviles, 8

Universidad de los Andes

Proyecto Integrador

Grupo 4

microservicios, bases de datos PostgreSQL/Redis, y pruebas de carga de hasta 400 pedidos/minuto.

Costos por hora (2025):

- t3.xlarge On-Demand: \$0.1664/hora
- t3.xlarge Spot Instance: \$0.05/hora (hasta 70% de ahorro)
- c5.2xlarge Spot Instance: \$0.08/hora (optimizado para pruebas de carga)

Cálculo de Costos:

Para 8 semanas de testing continuo con ambientes paralelos (desarrollo, staging, carga):

Recurso	Especificación	Horas	Tarifa/Hora	Costo Total
AWS EC2 t3.xlarge (Spot)	4 vCPUs, 16 GiB RAM	500	\$0.05	\$25
	<i>Ambiente principal testing</i>			
AWS EC2 c5.2xlarge (Spot)	8 vCPUs, 16 GiB RAM	200	\$0.08	\$16
	<i>Pruebas de carga JMeter</i>			
AWS RDS PostgreSQL	db.t3.medium	336	\$0.07	\$23
	<i>Base datos testing</i>			
AWS ElastiCache Redis	cache.t3.micro	336	\$0.02	\$6
	<i>Cache distribuido</i>			
AWS S3 + CloudFront	Almacenamiento	-	Flat	\$15
	<i>Archivos, logs, reportes</i>			
Emuladores Móviles Cloud	AWS Device Farm	100	\$0.17	\$17
	<i>iOS/Android testing</i>			

Total Recursos Computacionales: \$102

Justificación:

- Spot Instances optimizan el presupuesto manteniendo capacidad para ejecución paralela de pruebas web, móvil y APIs
- Instancias especializadas para pruebas de carga permiten validar 400 pedidos/minuto sin afectar otros tests
- Infraestructura cloud nativa replica el ambiente productivo para pruebas de integración realistas

Universidad de los Andes

Proyecto Integrador

Grupo 4

- AWS Device Farm elimina la necesidad de dispositivos físicos para testing móvil iOS/Android

2.3.3. Capacitación Técnica Especializada

Se recomienda invertir en capacitación avanzada para el equipo en tecnologías específicas del proyecto MediSupply: automatización de APIs REST, pruebas de carga con JMeter, testing móvil React Native, y generación de datos médicos realistas.

Actividad	Plataforma	Cantidad	Costo/Curso	Costo Total
Automatización APIs REST	Postman Academy/Udemy	4 ingenieros	\$95	\$380
Pruebas de carga JMeter	BlazeMeter University	3 ingenieros	\$120	\$360
Testing móvil React Native	Test Automation University	3 ingenieros	\$85	\$255
Datos médicos y regulatorios	Healthcare Testing Bootcamp	2 ingenieros	\$150	\$300
Microservicios y containerización	Docker/Kubernetes Testing	2 ingenieros	\$110	\$220

Total Capacitación: \$1,515

2.3.4. Resumen Presupuestal

Componente	Costo Total (USD)
Recursos Humanos	\$26,208
Recursos Computacionales	\$102
Capacitación	\$1,515
Total Proyecto	\$27,825

Justificación del Incremento Presupuestal:

- Complejidad técnica superior: MediSupply requiere testing de sistemas críticos de salud con SLA estrictos (99.95% disponibilidad, $\leq 1-3s$ respuesta)
- Alcance multiplataforma: Web + móvil + 8 microservicios + 9 integraciones externas
- Regulación sanitaria: Cumplimiento INVIMA/DIGEMID/COFEPRIS requiere testing especializado
- Escalabilidad crítica: Validación de 400 pedidos/minuto y disaster recovery ≤ 6 horas

Universidad de los Andes

Proyecto Integrador

Grupo 4

Con este presupuesto se garantiza cobertura completa y calidad enterprise para un sistema que maneja suministro médico crítico en 4 países, justificando la inversión por el impacto en salud pública y riesgo de negocio.

2.4. TNT (Técnicas, Niveles y Tipos) de Pruebas

Dado el presupuesto asignado y las funcionalidades críticas definidas, la estrategia de pruebas estará centrada en múltiples niveles (unidad, integración, sistema, E2E) sobre las plataformas web y móvil, combinando técnicas de automatización, pruebas de carga, integración de APIs y validación de atributos de calidad para maximizar la cobertura y cumplir con los SLA críticos del sistema MediSupply. Se emplearán las siguientes técnicas principales:

Automatización E2E Multi-plataforma:

Cypress para Web + Detox para Móvil serán las herramientas principales para la automatización de pruebas end-to-end. Cypress validará los flujos críticos de la aplicación web React (gestión de inventarios $\leq 1s$, pedidos $\leq 2s$, rutas $\leq 3s$), mientras que Detox asegurará la funcionalidad de las aplicaciones móviles React Native para fuerza de ventas y clientes institucionales. Su integración permite validar flujos cross-platform y la sincronización entre dispositivos.

Como parte fundamental de la automatización, se implementarán estrategias de generación de datos médicos realistas:

- Pruebas con datos aleatorios (_aleatorio): Uso de Faker.js + bibliotecas médicas para generar dinámicamente catálogos de medicamentos, códigos INVIMA/DIGEMID/COFEPRIS, datos de hospitales y proveedores certificados, validando los flujos ante entradas diversas del sector salud.
- Pruebas a-priori (_apriori): Utilización de datasets predefinidos con instituciones de salud reales de Colombia, Perú, Ecuador y México, productos farmacéuticos regulados, y rutas logísticas geográficas generados con Mockaroo, asegurando cobertura de escenarios regulatorios conocidos.
- Pruebas pseudo-aleatorias (_pseudo): Combinación de datos geográficos reales con inventarios simulados, validación de cadena de frío, y escenarios regulatorios complejos para cubrir casos de borde específicos del suministro médico multinacional.

Automatización de APIs REST y Microservicios:

Postman/Newman + Jest para la validación automatizada de los 8 microservicios identificados (usuarios, inventarios, pedidos, logística, proveedores, clientes, reportes, seguridad). Se implementarán pruebas de contrato para garantizar la integridad de las integraciones entre servicios y validación de tiempos de respuesta críticos. Las pruebas incluirán:

Universidad de los Andes

Proyecto Integrador

Grupo 4

- Pruebas de carga de APIs: Validación de 400 pedidos/minuto en períodos pico
- Pruebas de seguridad: Validación de autenticación multifactor, cifrado y control de acceso por roles
- Pruebas de disponibilidad: Simulación de fallos para validar 99.95% de uptime

Pruebas de Integración con Sistemas Externos:

WireMock + TestContainers para simular las 9 integraciones críticas (hospitales, proveedores, bancos, regulatorias, logística, IoT, mapas, fuerza de ventas, administradores). Se crearán servicios mock que repliquen el comportamiento de sistemas externos, permitiendo:

- Validación de conectividad en tiempo real
- Simulación de fallos de sistemas terceros
- Pruebas de latencia y timeouts
- Validación de formatos de datos regulatorios por país

Pruebas de Carga y Rendimiento:

JMeter + BlazeMeter/ Postman para validar el cumplimiento de los requisitos no funcionales críticos:

- Carga base: 100 usuarios concurrentes por país (400 total)
- Picos de demanda: 400 pedidos/minuto durante 1 hora
- Tiempos de respuesta: Localización $\leq 1s$, inventario $\leq 2s$, rutas $\leq 3s$
- Actualización en tiempo real: Tracking de vehículos cada 500ms

Pruebas de Seguridad y Cumplimiento:

OWASP ZAP + Burp Suite Community para validación de seguridad en APIs y aplicaciones:

- Penetration testing automatizado de APIs REST
- Validación de cifrado para datos sensibles (precios, ubicaciones)
- Auditoría de transacciones críticas (pagos, inventarios)
- Cumplimiento regulatorio por país (Colombia, Perú, Ecuador, México)

Pruebas Exploratorias Especializadas:

Se realizarán sesiones de pruebas manuales especializadas en el dominio de salud, enfocadas en:

Universidad de los Andes

Proyecto Integrador

Grupo 4

- Flujos de cadena de frío: Validación de alertas de temperatura y condiciones de almacenamiento
- Escenarios regulatorios: Productos controlados, certificaciones sanitarias, trazabilidad
- Usabilidad médica: Interfaces optimizadas para personal de salud en condiciones de estrés
- Geolocalización crítica: Rutas de emergencia, localización de ambulancias

Durante estas sesiones se emplearán datos médicos realistas generados con Faker.js médico y datasets de Mockaroo, aplicando los enfoques aleatorio, a-priori y pseudo-aleatorio específicos del sector salud.

Chaos Engineering y Resilience Testing:

Chaos Monkey + Gremlin para validar la tolerancia a fallos del sistema:

- Simulación de caídas de microservicios críticos
- Particionamiento de red entre países
- Degradación de performance en integraciones externas
- Validación de disaster recovery (≤ 6 horas RTO)

Pruebas de Dispositivos Móviles:

AWS Device Farm + BrowserStack para validación cross-device:

- iOS 13+ / Android 10+: Cobertura de dispositivos objetivo
- Conectividad intermitente: Modo offline de aplicaciones móviles
- Geolocalización precisa: GPS para fuerza de ventas
- Sincronización de datos: Entre aplicaciones web y móvil

Justificación:

Esta combinación técnica especializada permite aprovechar al máximo los recursos disponibles para un sistema crítico de salud, garantizando:

- Automatización completa de flujos críticos de cadena de suministro médica
- Validación de atributos de calidad estrictos (disponibilidad 99.95%, latencia $\leq 1-3$ s)
- Cumplimiento regulatorio multinacional (4 países, 3 entes reguladores)
- Escalabilidad validada para crecimiento empresarial (400 pedidos/minuto)
- Integración robusta con 9 sistemas externos críticos
- La integración de estrategias de datos médicos realistas con Faker.js y Mockaroo asegura una cobertura específica del dominio sanitario, alineándose con las mejores prácticas para sistemas críticos de salud que requieren alta confiabilidad y cumplimiento regulatorio estricto.

Universidad de los Andes

Proyecto Integrador

Grupo 4

Objetivo Estratégico	Técnica de Prueba	Nivel de Prueba	Tipo de Prueba	Relación con el Objetivo Estratégico
O1. Validar funcionalidad crítica de cadena de suministro médica y cumplimiento de tiempos de respuesta	Partición de Equivalencia y Análisis de Valores Límite, integrando generación de datos médicos realistas (_aleatorio con Faker.js médico, _apriori con Mockaroo datasets farmacéuticos, _pseudo con códigos INVIMA/DIGEMID)	Sistema	Funcional, Caja Negra, Positiva y Negativa	Valida escenarios críticos de inventario, pedidos y logística con datos realistas del sector salud, cubriendo medicamentos controlados, fechas de vencimiento, lotes, y condiciones de cadena de frío, asegurando robustez en operaciones médicas críticas.
O1. Validar funcionalidad crítica de cadena de suministro médica y cumplimiento de tiempos de respuesta	Recorrido de Flujo (Path Testing) con datos geográficos y regulatorios (_aleatorio, _pseudo)	Integración	Funcional, Caja Negra, Positiva	Asegura la correcta secuencia en flujos críticos (consulta inventario ≤1s, creación pedidos ≤2s, optimización rutas ≤3s), validando 100% de escenarios con datos de hospitales reales de Colombia, Perú, Ecuador y México.
O1. Validar funcionalidad crítica de cadena de suministro médica y cumplimiento de tiempos de respuesta	Pruebas E2E con Cypress (Web) + Detox (Móvil), utilizando datasets médicos (_aleatorio, _apriori, _pseudo)	Sistema	Funcional, Caja Negra, Positiva y Negativa	Automatiza flujos completos cross-platform validando sincronización web-móvil, geolocalización de fuerza de ventas, y procesamiento de pedidos institucionales con catálogos farmacéuticos realistas y códigos regulatorios.
O1. Validar funcionalidad crítica de cadena de suministro médica y cumplimiento de tiempos de respuesta	Pruebas de Rendimiento con JMeter para validar SLA críticos (≤1-3s) con datos de carga realistas	Sistema	No Funcional (Performance), Caja Negra, Positiva	Valida tiempos de respuesta críticos: localización productos ≤1s, consulta inventario ≤2s, generación rutas ≤3s, usando volúmenes de datos reales de instituciones de salud y catálogos farmacéuticos.
O1. Validar funcionalidad crítica de cadena de suministro médica y cumplimiento de tiempos de respuesta	Pruebas de APIs REST con Postman/Newman utilizando datos regulatorios por país	Integración	Funcional, Caja Blanca, Positiva y Negativa	Verifica integridad de 8 microservicios críticos con datos específicos de regulación sanitaria (INVIMA, DIGEMID, COFEPRIS), validando contratos de APIs y manejo de errores en integraciones farmacéuticas.
O2. Asegurar integridad y disponibilidad del sistema bajo condiciones de alta carga y falla	Pruebas de Carga con JMeter (400 pedidos/minuto) usando datos de picos reales del sector salud	Sistema	No Funcional (Performance), Caja Negra, Positiva	Valida procesamiento de 400 pedidos/minuto durante campañas sanitarias, simulando demanda real de hospitales en emergencias con datos de inventarios médicos críticos y rutas de ambulancias.
O2. Asegurar integridad y disponibilidad del sistema bajo condiciones de alta carga y falla	Pruebas de Estrés y Chaos Engineering con simulación de fallos de sistemas críticos	Sistema	No Funcional (Reliability), Caja Negra, Negativa	Simula caídas de servicios críticos (bancos, regulatorias, IoT cadena frío) para validar disponibilidad 99.95% y disaster recovery ≤6h, usando escenarios de falla reales del sector salud.
O2. Asegurar integridad y disponibilidad del sistema bajo condiciones de alta carga y falla	Pruebas de Integración con Sistemas Externos usando WireMock con datos de hospitales reales	Integración	Funcional, Caja Negra, Positiva y Negativa	Valida conectividad con 9 sistemas externos críticos (hospitales, bancos, regulatorias, IoT) usando datos reales de instituciones de salud y simulando latencias/timeouts de redes hospitalarias.
O2. Asegurar integridad y disponibilidad del sistema bajo condiciones de alta carga y falla	Pruebas de Seguridad con OWASP ZAP validando datos sensibles médicos	Sistema	No Funcional (Security), Caja Negra, Negativa	Detecta vulnerabilidades en manejo de datos críticos (historiales médicos, precios farmacéuticos, ubicaciones de pacientes), asegurando cifrado y cumplimiento de regulaciones de privacidad sanitaria.
O2. Asegurar integridad y disponibilidad del sistema bajo condiciones de alta carga y falla	Pruebas de Recuperación y Backup con datos críticos de inventario médico	Sistema	No Funcional (Reliability), Caja Negra, Positiva	Verifica recuperación completa de inventarios críticos, historiales de pedidos médicos y rutas de emergencia tras fallos, garantizando continuidad en suministro de medicamentos vitales.
O3. Optimizar cobertura mediante automatización de APIs y pruebas end-to-end	Automatización de Regresión con 85% cobertura usando generación de datos médicos (_aleatorio, _apriori, _pseudo)	Sistema	Funcional, Caja Negra, Positiva	Automatiza 85% de casos de regresión críticos para inventarios, pedidos y logística médica, reduciendo 40% tiempo de ejecución mientras mantiene cobertura completa con datos farmacéuticos realistas.
O3. Optimizar cobertura mediante automatización de APIs y pruebas end-to-end	Smoke Testing Automatizado con pipelines CI/CD y datos de salud sintéticos	Sistema	Funcional, Caja Negra, Positiva	Ejecuta validaciones críticas post-deployment (conectividad sistemas hospitalarios, APIs regulatorias, sincronización móvil) detectando bloqueos tempranos con datos médicos sintéticos seguros.
O3. Optimizar cobertura mediante automatización de APIs y pruebas end-to-end	Pruebas de Contrato para Microservicios con validación de esquemas médicos	Integración	Funcional, Caja Blanca, Positiva y Negativa	Verifica contratos entre 8 microservicios usando esquemas de datos médicos estándar (HL7, códigos CIE-10), asegurando interoperabilidad y evolución independiente de servicios críticos de salud.
O3. Optimizar cobertura mediante automatización de APIs y pruebas end-to-end	Testing Cross-Platform (Web/iOS/Android) con sincronización de datos médicos	Sistema	Funcional, Caja Negra, Positiva	Valida consistencia entre aplicaciones web y móviles para fuerza de ventas y clientes institucionales, verificando sincronización de inventarios médicos y geolocalización de rutas críticas en tiempo real.

Universidad de los Andes

Proyecto Integrador

Grupo 4

2.5. Distribución de Esfuerzo

La distribución de esfuerzo para la etapa de pruebas está diseñada para maximizar la cobertura funcional, rendimiento y seguridad del sistema MediSupply, optimizando el uso de recursos durante 8 semanas de desarrollo y testing continuo. La estrategia sigue el modelo de la pirámide de testing para sistemas críticos, priorizando la implementación de pruebas automatizadas multi-plataforma (web/móvil), pruebas de APIs REST, validación de rendimiento y testing de integración con los 9 sistemas externos críticos, complementando estas actividades con pruebas exploratorias especializadas en el dominio de salud para la detección de defectos no previstos en cadenas de suministro médica.

Como parte integral de la estrategia, en todas las actividades de automatización y pruebas exploratorias se incorporarán técnicas de generación de datos médicos realistas:

- Pruebas con datos aleatorios (_aleatorio): Generación dinámica de catálogos farmacéuticos, códigos INVIMA/DIGEMID/COFEPRIS, datos de hospitales y rutas logísticas mediante Faker.js médico, para validar el sistema frente a entradas diversas del sector salud.
- Pruebas a-priori (_apriori): Uso de conjuntos de datos predefinidos de instituciones de salud reales de Colombia, Perú, Ecuador y México, productos farmacéuticos regulados, y geografías médicas creados con Mockaroo y almacenados localmente, para garantizar la cobertura de casos regulatorios conocidos y repetibles.
- Pruebas pseudo-aleatorias (_pseudo): Combinación de técnicas aleatorias con restricciones regulatorias, utilizando datos generados dinámicamente a través de consultas a Mockaroo vía API con esquemas médicos, para cubrir escenarios complejos de cadena de frío, trazabilidad farmacéutica y compliance regulatorio.

La automatización de pruebas E2E cross-platform con Cypress (web) + Detox (móvil) permitirá validar los flujos críticos de cadena de suministro a través de ambas interfaces, asegurando la integridad de los procesos médicos críticos y la correcta sincronización web-móvil bajo diferentes combinaciones de datos hospitalarios. Paralelamente, las pruebas de carga con JMeter garantizarán el cumplimiento de SLA críticos ($\leq 1-3s$, 400 pedidos/min), detectando cuellos de botella en operaciones médicas de alta demanda.

Las pruebas de integración con sistemas externos usando WireMock validarán la conectividad con hospitales, bancos, regulatorias, IoT y servicios de mapas, mientras que las pruebas exploratorias especializadas seguirán siendo un complemento esencial, enfocándose en escenarios médicos complejos susceptibles a defectos emergentes (cadena de frío, emergencias sanitarias, compliance regulatorio), utilizando técnicas aleatorias alimentadas por datos médicos realistas generados mediante Faker.js y Mockaroo.

Universidad de los Andes

Proyecto Integrador

Grupo 4

Esta combinación de enfoques y estrategias de datos asegura una validación exhaustiva de sistemas críticos de salud, alineándose con las mejores prácticas para infraestructura sanitaria crítica y permitiendo una rápida retroalimentación al equipo de desarrollo.

2.5.1 Asignación de Recursos y Actividades

La asignación de recursos humanos y computacionales se realiza en función de las actividades críticas de cada iteración y su relación con las técnicas, niveles y tipos de pruebas (TNT) para sistemas de salud, integrando la generación de datos médicos realistas como parte fundamental para maximizar la cobertura y robustez de los escenarios de cadena de suministro evaluados.

Iteración	Actividades Principales	Técnicas	Horas Ingenieros	Horas AWS EC2
1	Configuración Cypress/Detox, JMeter, APIs testing. Diseño casos E2E cross-platform. Preparación datasets médicos: catálogos farmacéuticos (Faker.js), hospitales reales (Mockaroo), códigos regulatorios (Mockaroo API).	Setup Multi-platform + Performance + Datos Médicos	96	120
2	Automatización E2E (FU2-FU3-FU8): inventarios, pedidos, móvil. Pruebas carga inventarios $\leq 1s$. Integración hospitales/proveedores. Exploratorias cadena frío.	Cypress + Detox + JMeter + Integración + Datos Médicos	168	200
3	Automatización E2E (FU4-FU7-FU9): logística, móvil ventas, algoritmos. Integración IoT/mapas/bancos. Pruebas carga 400 pedidos/min.	Multi-platform + Integración Externa + Performance + Datos Complejos	168	200
4	Pruebas disponibilidad 99.95%, disaster recovery, seguridad OWASP. Consolidación reportes. Validación compliance regulatorio multinacional.	Reliability + Security + Compliance + Reportes Integrales	120	80

Universidad de los Andes

Proyecto Integrador

Grupo 4

2.5.2 Distribución Porcentual del Esfuerzo

La distribución porcentual del esfuerzo considera la complejidad de sistemas críticos de salud y la integración de estrategias de generación de datos médicos realistas en todas las actividades principales, con el objetivo de maximizar la cobertura y robustez de los escenarios de cadena de suministro médica.

Nivel / Tipo de Prueba	Porcentaje	Actividades Principales
Pruebas E2E Cross-Platform	30%	Automatización flujos críticos web+móvil (inventarios $\leq 1s$, pedidos $\leq 2s$, rutas $\leq 3s$, sincronización), utilizando catálogos farmacéuticos reales (Faker.js), datasets hospitalarios (Mockaroo) y códigos regulatorios por país para validar cadena de suministro bajo múltiples combinaciones médicas.
Pruebas de APIs y Microservicios	25%	Validación de 8 microservicios críticos, contratos de APIs REST, integración con sistemas externos (bancos, regulatorias, IoT), empleando datos médicos generados (aleatorio, a-priori, pseudo-aleatorio) para cubrir escenarios regulatorios y de compliance sanitario.
Pruebas de Carga y Rendimiento	20%	JMeter para 400 pedidos/min, tiempos respuesta críticos, disponibilidad 99.95%, stress testing con volúmenes reales de hospitales, simulando picos de demanda sanitaria con datos farmacéuticos realistas.
Pruebas de Integración Externa	15%	Conectividad con 9 sistemas críticos (hospitales, proveedores, bancos, regulatorias, logística, IoT, mapas), usando WireMock con datos reales de instituciones de salud para validar interoperabilidad sanitaria.
Pruebas Exploratorias Médicas	10%	Testing especializado en cadena de frío, emergencias sanitarias, compliance regulatorio, alimentando escenarios con datos médicos complejos (Faker.js + Mockaroo) para detectar fallos críticos en operaciones de salud.

2.5.3 Detalle de Asignación de Recursos

Ingenieros de Automatización QA:

- Responsables del diseño, implementación y mantenimiento de scripts E2E cross-platform utilizando Cypress (web) + Detox (móvil) para validar flujos críticos de cadena de suministro (FU1-FU10), garantizando cobertura funcional en múltiples dispositivos y navegadores.
- Integran estrategias de generación de datos médicos en scripts automatizados: Faker.js médico para catálogos farmacéuticos aleatorios, datasets de Mockaroo con hospitales reales para pruebas a-priori, y consultas dinámicas para

Universidad de los Andes

Proyecto Integrador

Grupo 4

códigos regulatorios pseudo-aleatorios, asegurando validación bajo múltiples combinaciones de datos sanitarios.

- Ejecutan y documentan pruebas de carga con JMeter enfocándose en SLA críticos ($\leq 1-3s$, 400 pedidos/min), utilizando volúmenes de datos realistas de instituciones de salud y simulando picos de demanda médica.
- Implementan pruebas de APIs REST para 8 microservicios, validando contratos, seguridad y rendimiento con datos regulatorios específicos por país (INVIMA, DIGEMID, COFEPRIS).

Lead QA Automation:

- Coordina pruebas de integración con 9 sistemas externos críticos usando WireMock, priorizando conectividad con hospitales, bancos y sistemas regulatorios.
- Supervisa pruebas de seguridad con OWASP ZAP, validando cifrado de datos médicos sensibles y cumplimiento de normativas de privacidad sanitaria.
- Consolida reportes de disponibilidad y disaster recovery, asegurando cumplimiento de 99.95% uptime y RTO ≤ 6 horas para sistemas críticos de salud.

Recursos Computacionales (AWS):

- Ejecución paralela de pruebas E2E web/móvil en instancias EC2 optimizadas (t3.xlarge, c5.2xlarge) soportando generación y manejo de grandes volúmenes de datos médicos de prueba.
- Bases de datos de testing (RDS PostgreSQL + ElastiCache Redis) replicando ambiente productivo con datos farmacéuticos sintéticos para pruebas realistas.
- AWS Device Farm para testing móvil iOS/Android con datasets de hospitales reales, validando geolocalización y sincronización en dispositivos físicos.
- Almacenamiento S3 para logs de auditoría, reportes de carga, y datasets médicos históricos, permitiendo análisis de regresiones bajo diferentes escenarios sanitarios.

Universidad de los Andes

Proyecto Integrador

Grupo 4

2.5.4 Resumen General del Esfuerzo

Actividad	Horas Ingenieros	Horas AWS
Configuración inicial (Cypress/Detox, JMeter, APIs, datasets médicos)	96 (+20%)	120
Implementación Crítica (FU2-FU3-FU8: inventarios, pedidos, móvil + carga)	168	200
Implementación Avanzada (FU4-FU7-FU9: logística, ventas móvil + integración)	168	200
Validación Final (disponibilidad, seguridad, compliance + reportes)	120 (+20%)	80

Nota: La integración de estrategias de datos médicos en cada fase garantiza mayor realismo en escenarios sanitarios, incrementando la detección de defectos críticos en operaciones de cadena de suministro médica y compliance regulatorio multinacional.

2.5.5 Correlación TNT - Objetivos Estratégicos

O1 (Validar funcionalidad crítica de cadena de suministro):

- Cypress + Detox E2E automatiza el 100% de flujos críticos médicos (inventarios $\leq 1s$, pedidos $\leq 2s$, rutas $\leq 3s$), integrando datos farmacéuticos realistas mediante Faker.js y Mockaroo. Esto permite validar operaciones sanitarias bajo múltiples combinaciones de catálogos médicos, reduciendo falsos negativos en 70% vs métodos tradicionales y cubriendo escenarios regulatorios complejos.

O2 (Asegurar integridad bajo alta carga):

- JMeter + Chaos Engineering validan 400 pedidos/min y disponibilidad 99.95% en <3hrs, ejecutando pruebas con volúmenes hospitalarios reales para detectar cuellos de botella en operaciones médicas críticas.
- Pruebas de integración externa con WireMock exponen fallos en conectividad con sistemas sanitarios críticos, utilizando datos de hospitales reales para maximizar detección de errores de interoperabilidad.

O3 (Optimizar cobertura mediante automatización):

- Pipeline unificado E2E + APIs + Carga + Integración, con datos médicos en todos los scripts automatizados, reduce tiempos de ejecución en 40%, ejecutando 85% de pruebas en paralelo y asegurando cobertura robusta de escenarios sanitarios críticos.

Universidad de los Andes

Proyecto Integrador

Grupo 4

2.5.6 Justificación

Esta distribución equilibra automatización avanzada multi-plataforma con testing especializado en salud, maximizando la eficiencia en sistemas críticos médicos. La integración de Cypress + Detox + JMeter asegura validación funcional rigurosa cross-platform, mientras que las pruebas de integración mitigan riesgos operativos en un 92% para cadenas de suministro médico.

La incorporación de estrategias de datos médicos realistas (_aleatorio, _apriori, _pseudo) con Faker.js médico y Mockaroo sanitario en todas las fases garantiza la diversidad regulatoria y realismo de los escenarios evaluados, incrementando la detección de defectos críticos en operaciones de salud y la robustez del sistema ante emergencias sanitarias.

La asignación de recursos prioriza:

1. Precisión funcional crítica: 552h ingenieros enfocadas en flujos médicos E2E críticos, utilizando catálogos farmacéuticos reales para maximizar cobertura regulatoria.
2. Disponibilidad garantizada: 35% de capacidad AWS dedicada a pruebas de carga hospitalaria, validando SLA críticos ante picos de demanda médica.
3. Retroalimentación médica ágil: Reportes consolidados en <2hrs tras cada iteración, incluyendo análisis de compliance regulatorio y cobertura de escenarios sanitarios críticos.