

# INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Electrónica, Sistemas e Informática  
Desarrollo Tecnológico y Generación de Riqueza Sustentable

PROYECTO DE APLICACIÓN PROFESIONAL (PAP)



ITESO, Universidad  
Jesuita de Guadalajara

PAP4N01A PROGRAMA DE LA INDUSTRIA DE ALTA TECNOLOGIA I

ORACLE

**PRESENTA**

Alumno: IDS, EDUARDO ANTONIO PELAYO CARRILLO

Profesor PAP: Juan Manuel Islas Espinoza, PMP®

Tlaquepaque, Jalisco, Diciembre 2025



## ÍNDICE

### Contenido

<b>REPORTE PAP</b> .....	3
Presentación Institucional de los Proyectos de Aplicación Profesional .....	3
<b>Resumen</b> .....	4
<b>1. Introducción</b> .....	5
1.1 Antecedentes.....	5
1.2 Justificación.....	6
1.3 Objetivos .....	6
1.4 Contexto .....	7
1.5 Inventario de Competencias.....	8
1.6 Plan Educativo.....	8
1.7 Entregables .....	9
1.8 Involucrados .....	9
1. Desarrollo del Proyecto PAP .....	10
2.1 Administración del Proyecto .....	10
2.2 Sustento Teórico y Metodológico.....	10
2.3 Descripción del Proyecto .....	10
2.4 Objetivos del Proyecto .....	11
2.5 Plan de Trabajo .....	11
2.6 Equipo de Trabajo .....	12
2.7 Plan de Comunicaciones .....	12
2.8 Plan de Calidad .....	13
2.9 Seguimiento y Control .....	14
2.10 Cierre del Proyecto .....	14
3.1 Productos Obtenidos .....	17
3.2 Estimación del Impacto.....	18
<b>4. Reflexiones del alumno</b> .....	20
4.1 Aprendizajes Profesionales.....	20
4.2 Aprendizajes Sociales .....	20
4.3 Aprendizajes Éticos.....	21
4.4 Aprendizajes Personales .....	21
4.5 Tareas Aprendidas.....	22

5. Conclusiones.....	23
6. Bibliografía y Anexos (solo en caso de ser necesarios).....	24

## REPORTE PAP

### Presentación Institucional de los Proyectos de Aplicación Profesional

Los Proyectos de Aplicación Profesional (PAP) son una modalidad educativa del ITESO en la que el estudiante aplica sus saberes y competencias socio-profesionales para el desarrollo de un proyecto que plantea soluciones a problemas de entornos reales. Su espíritu está dirigido para que el estudiante ejerza su profesión mediante una perspectiva ética y socialmente responsable.

A través de las actividades realizadas en el PAP, se acreditan el servicio social y la opción terminal. Así, en este reporte se documentan las actividades que tuvieron lugar durante el desarrollo del proyecto, sus incidencias en el entorno, y las reflexiones y aprendizajes profesionales que el estudiante desarrolló en el transcurso de su labor.

## Resumen

El presente Proyecto de Aplicación Profesional (PAP) se desarrollará en colaboración con Oracle México, dentro del área de desarrollo de software especializado en telecomunicaciones. El objetivo principal es participar como Software Developer Intern en actividades relacionadas con la mejora y mantenimiento de microservicios que forman parte del Policy Control Function (PCF) en entornos de red 5G.

El alcance del proyecto incluye la identificación y resolución de errores en microservicios, la implementación de pruebas unitarias y de contrato en Java, el uso de herramientas de control de versiones y la colaboración en prácticas de desarrollo ágil. A lo largo del periodo, también se fortalecerán competencias técnicas clave como Java Git, CI/CD y metodologías ágiles, además de competencias profesionales como comunicación en inglés, trabajo en equipo y adaptación a entornos corporativos de alta exigencia.

La metodología de trabajo se basará en la integración a un equipo de desarrollo global bajo el marco de metodologías ágiles (Scrum), con un esquema de retroalimentación continua mediante revisiones de código, documentación técnica y reuniones de seguimiento. Este enfoque permitirá adquirir experiencia práctica en proyectos reales de alcance global, vinculando directamente los aprendizajes académicos con la práctica profesional.

## 1. Introducción

El Proyecto de Aplicación Profesional (PAP) se desarrollará en colaboración con Oracle México, dentro del área de desarrollo de software para telecomunicaciones. Durante este periodo, se integrará la experiencia académica adquirida en la carrera de Ingeniería en Desarrollo de Software con la práctica profesional en un entorno corporativo de alcance global.

Este proyecto tendrá como propósito participar en el mantenimiento, corrección y validación de microservicios del Policy Control Function (PCF), un componente esencial en la arquitectura de redes 5G. Las actividades estarán enfocadas en la resolución de errores, la creación de pruebas unitarias y de contrato en Java, y la colaboración en procesos de integración continua dentro de un equipo internacional de ingeniería.

De manera paralela, este PAP permitirá reforzar competencias profesionales relacionadas con la programación en Java, manejo de bases de datos, metodologías ágiles, control de versiones con Git y comunicación en inglés, vinculando la formación académica con el aprendizaje práctico.

El desarrollo de este proyecto contribuirá a la adquisición de experiencia real en tecnologías emergentes y a la consolidación de habilidades técnicas y profesionales, con el objetivo de preparar una transición efectiva hacia el ámbito laboral al concluir los estudios universitarios.

### 1.1 Antecedentes

Organización huésped: Oracle México.

Ramas tecnológicas: Cloud Computing, bases de datos, microservicios, redes 5G, middleware empresarial, automatización y soluciones de inteligencia de negocios.

Productos y servicios: Bases de datos (Oracle Database, Autonomous Database), servicios de nube (Oracle Cloud Infrastructure), middleware para empresas, herramientas de desarrollo, y soluciones para telecomunicaciones como sistemas de facturación, control de políticas y gestión de redes 5G.

Misión y valores: Innovar en soluciones tecnológicas que impulsen a empresas y gobiernos a operar de manera más eficiente, segura y escalable, con valores de innovación, integridad, orientación al cliente y colaboración.

Clientes principales: Empresas de telecomunicaciones mexicanas y de norte américa.

## 1.2 Justificación

Mi motivación para realizar este PAP en Oracle se basa en la oportunidad de aplicar los conocimientos adquiridos durante mi formación en Ingeniería en Desarrollo de Software, especialmente en Java, programación orientada a objetos, bases de datos y metodologías ágiles. Participar en un entorno corporativo de alto nivel me permitirá adquirir competencias clave para mi desarrollo profesional, como el trabajo en proyectos de microservicios y el uso de tecnologías de vanguardia en telecomunicaciones.

El esfuerzo estimado para cumplir con este PAP será de 30 a 35 horas semanales, distribuidas en mis actividades como becario en Oracle y en capacitación paralela.

Recursos y facilidades disponibles:

- Equipo de trabajo
- Acceso a la infraestructura tecnológica de Oracle (ambientes de desarrollo, repositorios de código, herramientas de colaboración).
- Acompañamiento de un líder técnico y revisiones periódicas de avances.
- Capacitación interna (documentación, entrenamientos en línea y recursos de Oracle University).

## 1.3 Objetivos

De la empresa huésped: Integrar a estudiantes en un entorno profesional real, contribuyendo a proyectos de innovación y desarrollo de software, mientras se forman como futuros profesionales capacitados.

Personales: Fortalecer mis habilidades en Java, pruebas unitarias (JUnit, Mockito), microservicios en 5G, Git y pipelines de integración continua, así como adquirir experiencia en trabajo colaborativo dentro de un equipo de ingeniería global.

## 1.4 Contexto

Departamento / Área: Desarrollo de software en el área de telecomunicaciones (PCF – Policy Control Function).

Tipo de proyecto: Apoyo a áreas operativas y mejora de procesos en microservicios críticos para la red 5G.

Rol: Software Developer Intern (Becario).

Funciones:

- Identificación y resolución de bugs en microservicios de PCF.
- Desarrollo de pruebas unitarias y de contrato para validar correcciones.
- Documentación técnica de soluciones implementadas.
- Colaboración en prácticas de desarrollo ágil (scrums, revisiones de código, integración continua).

## 1.5 Inventario de Competencias

Materia	PAP1 DESI	Semestre	20230
Profesor	Juan Manuel Islas	Horario:	Lun-Jue 16-18
Alumno:	Eduardo Antonio Pelayo Carrillo	Carrera:	IDS
Empresa:	Oracle		
Proyecto:	Policy 5g		

### Inventario de Competencias

No.	Competencia	Req	Adq	GAP	Obj	Prior
1	<b>Programacion en Java</b>	4	2	2	4	A
1.1	Uso de clases, interfaces y colecciones	4	2	2	4	A
1.2	Programacion orientada a objetos	4	2	2	4	A
1.3	Manejo de excepciones y control de flujo	4	2	2	4	A
2	<b>Manejo de contenedores con Docker</b>	3	2	1	3	A
2.1	Creacion y administraciones de imágenes docker	3	2	1	3	A
2.2	Ejecucion y monitorio de contenedores	3	2	1	3	A
3	<b>Orquestacion de contenedores con Kubernetes</b>	3	2	1	3	A
3.1	Creacion de pods, servicios y despliegues	3	2	1	3	A
3.2	Escalamiento y actualizacion de aplicaciones	3	2	1	3	A
4	<b>Pruebas Unitarias y de Contrato</b>	3	2	1	3	A
5	Implementacion de pruebas con Junit	3	2	1	3	A
6	Simulacion con Mockito	3	2	1	3	M
7	<b>Metodologias ágiles (Scrum)</b>	3	2	1	3	M
8	Participacion en ceremonias agiles	3	2	1	3	M
	Elaboracion de historias de usuario y tareas	3	2	1	3	M

## 1.6 Plan Educativo

### Plan de Actividades

No.	Actividad Educativa	Tipo Actividad	Total Hrs	Fecha Inicio	Fecha Termino	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	<b>Programacion en Java</b>																					
1.1	Curso en línea sobre colecciones y estructuras de datos en Java	Curso en línea	15	01/09/2025	21/09/2025																	
1.2	Taller de prácticas de POO en Java con ejercicios de herencia y polimorfismo	Autoestudio	8	22/09/2025	05/10/2025																	
1.3	Implementación de casos de manejo de excepciones en microservicios	Practica en proyecto	18	06/10/2025	26/10/2025																	
2	<b>Manejo de contenedores con docker</b>																					
2.1	Curso práctico de creación y administración de imágenes Docker	Curso en línea	10	08/09/2025	21/09/2025																	
2.2	Ejercicios de ejecución y monitoreo de contenedores Docker	Practica en proyecto	12	22/09/2025	05/10/2025																	
3	<b>Orquestación de contenedores con Kubernetes</b>																					
3.1	Curso introductorio de Kubernetes (Pods, servicios, despliegues)	Curso en línea	15	06/10/2025	26/10/2025																	
3.2	Práctica de escalamiento y actualización de aplicaciones en clústeres Kubernetes	Practica en proyecto	18	27/10/2025	16/11/2025																	
4	<b>Práctica de escalamiento y actualización de aplicaciones en Kubernetes</b>																					
4.1	Creación de pruebas con Mockito para simular dependencias	Autoestudio	12	29/09/2025	12/10/2025																	
4.2	Implementación de pruebas unitarias con JUnit en microservicios	Practica en proyecto	10	13/10/2025	26/10/2025																	
5	<b>Metodologías ágiles (Scrum)</b>																					
5.1	Participación en ceremonias ágiles (daily, planning, retrospective)	Practica en proyecto	48	01/09/2025	21/12/2025																	
5.2	Elaboración de historias de usuario y descomposición en tareas	Practica en proyecto	24	15/09/2025	12/10/2025																	

## 1.7 Entregables

Correcciones a microservicios de PCF (documentadas en tickets internos).

Pruebas unitarias y de contrato en Java.

Reportes de bugs resueltos con su documentación técnica.

Participación en revisiones de código y sesiones de planeación ágiles.

## 1.8 Involucrados

No	Nombre	Rol	Responsabilidad
1	Victor Alvidrez	Product Owner	Planning and PO activities
2	Alejandro Gras	QA/DEV	Uplifts, development, bug backlog, fortify, automation
3	Lakshmi Sahithi	QA/DEV	Build from source
4	Guillermo Pineda	QA/DEV	CPU, security customer activities, QA, fortify
5	Manasvi Alimchandani	QA/DEV	Business approval
6	Dhruvi Shah	QA/DEV	Business approval
7	Tanvi Bahedia	QA/DEV	Business approval
8	Eduardo Antonio Pelayo Carrillo	QA/DEV (INTERN)	Uplifts, development, automation

## 1. Desarrollo del Proyecto PAP

### 2.1 Administración del Proyecto

PROCESO	Num. Aprox. Horas
INICIO	2hr
PLANEACIÓN	2hr
EJECUCIÓN	6hr
SEGUIMIENTO Y CONTROL	4hr
CIERRE	2hr

### 2.2 Sustento Teórico y Metodológico

Sustento teórico:

Ingeniería de software orientada a objetos (principios SOLID) y buenas prácticas de clean code.

Pruebas de software con enfoque piramidal (unitarias, integración/contrato y validaciones en ambientes controlados).

Control de versiones con flujos de ramas tipo trunk-based o Gitflow (según lineamientos internos).

Observabilidad básica: logging estructurado y métricas para diagnóstico.

Metodología de trabajo:

Marco ágil (Scrum) para la gestión de trabajo y seguimiento: sprints quincenales, dailies, planning, refinements, reviews y retrospectivas.

Ciclo de vida de desarrollo interno para microservicios: análisis del ticket, diseño técnico, implementación en Java, pruebas unitarias y de contrato, code review, validaciones en pipeline CI/CD, despliegue a ambientes internos.

Aseguramiento de calidad: estándares de revisión de código, análisis estático y cumplimiento de políticas internas de seguridad y cumplimiento.

### 2.3 Descripción del Proyecto

Participación en el mantenimiento, corrección y validación de microservicios que integran la Policy Control Function (PCF) en entornos 5G. Actividades principales: resolución de bugs, creación/mejora de pruebas (unitarias y de contrato), automatización básica en pipelines y documentación técnica.

Estructura (EDT a primer nivel, descripción textual):

1. Análisis y refinamiento de tickets
2. Diseño técnico y criterios de aceptación
3. Implementación y pruebas unitarias en Java
4. Pruebas de contrato e integración en ambientes internos
5. Code review y aseguramiento de calidad (análisis estático)
6. Automatización en CI/CD y validaciones
7. Documentación técnica y handover
8. Soporte a incidencias y mejoras continua

Participación principal: 3), 4), 5), 6) y 7).

Recursos más importantes:

Herramientas internas de repositorios y CI/CD (control de versiones y pipelines).

Entornos corporativos para pruebas/validación y monitoreo.

Ecosistema Java (JDK, framework de pruebas interno y librerías aprobadas).

## 2.4 Objetivos del Proyecto

Mantenimiento y mejora de microservicios backend (Java) con foco en aseguramiento de calidad y automatización (pruebas unitarias/contrato y CI/CD) dentro del dominio de telecomunicaciones (PCF 5G).

## 2.5 Plan de Trabajo

No.	Competencia	Nivel Adquirido al Inicio	Nivel Objetivo al final PAP	Objetivo final PAP	Prior
1	Programación Java (POO, colecciones, testing)	2	3	3	A
2	Manejo de contenedores docker	1	3	3	A
3	Orquestacion con kubernetes	0	3	3	A
4	Escalamiento con kubernetes	0	2	2	M
5	Metodología Agil	1	3	3	M

## Plan de Actividades

No.	Actividad Educativa	Tipo Actividad	Total Hrs	Fecha Inicio	Fecha Termino	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	<b>Programacion en Java</b>																				
1.1	Curso en línea sobre colecciones y estructuras de datos en Java	Curso en línea	15	01/09/2025	21/09/2025																
1.2	Taller de prácticas de POO en Java con ejercicios de herencia y polimorfismo	Autoestudio	8	22/09/2025	05/10/2025																
1.3	Implementación de casos de manejo de excepciones en microservicios	Practica en proyecto	18	06/10/2025	26/10/2025																
2	<b>Manejo de contenedores con docker</b>																				
2.1	Curso práctico de creación y administración de imágenes Docker	Curso en línea	10	08/09/2025	21/09/2025																
2.2	Ejercicios de ejecución y monitoreo de contenedores Docker	Practica en proyecto	12	22/09/2025	05/10/2025																
3	<b>Orquestación de contenedores con Kubernetes</b>																				
3.1	Curso introductorio de Kubernetes (Pods, servicios, despliegues)	Curso en línea	15	06/10/2025	26/10/2025																
3.2	Práctica de escalamiento y actualización de aplicaciones en clústeres Kubernetes	Practica en proyecto	18	27/10/2025	16/11/2025																
4	<b>Práctica de escalamiento y actualización de aplicaciones en Kubernetes</b>																				
4.1	Creación de pruebas con Mockito para simular dependencias	Autoestudio	12	29/09/2025	12/10/2025																
4.2	Implementación de pruebas unitarias con JUnit en microservicios	Practica en proyecto	10	13/10/2025	26/10/2025																
5	<b>Metodologías ágiles (Scrum)</b>																				
5.1	Participación en ceremonias ágiles (daily, planning, retrospective)	Practica en proyecto	48	01/09/2025	21/12/2025																
5.2	Elaboración de historias de usuario y descomposición en tareas	Practica en proyecto	24	15/09/2025	12/10/2025																

## 2.6 Equipo de Trabajo

No	Nombre	Rol	Responsabilidad
1	Victor Alvidrez	Product Owner	Planning and PO activities
2	Alejandro Gras	QA/DEV	Uplifts, development, bug backlog, fortify, automation
3	Lakshmi Sahithi	QA/DEV	Build from source
4	Guillermo Pineda	QA/DEV	CPU, security costumer activities, QA, fortify
5	Manasvi Alimchandani	QA/DEV	Businnes approval
6	Dhruvi Shah	QA/DEV	Businnes approval
7	Tanvi Bahedia	QA/DEV	Businnes approval
8	Eduardo Antonio Pelayo Carrillo	QA/DEV (INTERN)	Uplifts, development, automation

## 2.7 Plan de Comunicaciones

Daily stand-up:

Emisor: Cada integrante

Mensaje: Avances, bloqueadores, plan del día

Receptor: Equipo

Medio: Zoom

Frecuencia: Diario (d)

Sprint planning/refinement:

Emisor: PO y equipo

Mensaje: Objetivos de sprint, estimaciones y criterios de aceptación  
Receptor: Equipo  
Medio: Zoom  
Frecuencia: Quincenal (q) y semanal para refinements (s)

Code review:

Emisor: Desarrollador que crea el PR  
Mensaje: Cambios propuestos y evidencias de pruebas  
Receptor: Revisores asignados  
Medio: Plataforma de repositorios  
Frecuencia: Según entregables

Estado académico PAP:

Emisor: Alumno  
Mensaje: Avances, evidencias y dudas  
Receptor: Profesor PAP  
Medio: teams  
Frecuencia: según acuerdo

## 2.8 Plan de Calidad

Entrega: PR con implementación y pruebas:

Receptor: Revisores (pares/lead)

Criterios: Cumplir estándares de código, cobertura mínima de pruebas, sin warnings críticos de análisis estático, aceptación de criterios funcionales

Siguiente paso: Merge condicional a rama de integración

Validación en CI/CD:

Receptor: Pipeline automatizado

Criterios: Compilación, suite de pruebas exitosa, escaneo de seguridad interno, verificación de convenciones

Siguiente paso: Despliegue a ambiente interno de prueba

Aceptación funcional:

Receptor: QA/PO (según el caso)

Criterios: Cumplimiento de criterios de aceptación, no regresiones, evidencia registrada

Siguiente paso: Cierre de ticket y preparación de release interno (si aplica)

Documentación:

Receptor: Equipo

Criterios: Notas de cambios, pasos de prueba, impactos y riesgos

Siguiente paso: Publicación en repositorio/documentación interna

## 2.9 Seguimiento y Control

Con el equipo:

Dailies para visibilidad de avances y bloqueadores.

Sprint planning y refinements para ajustar alcance, estimaciones y prioridades.

Revisión semanal/quincenal del tablero (estado de tickets, WIP, throughput).

Retrospectivas para acciones de mejora (p. ej., elevar cobertura, mejorar tiempos de revisión, clarificar criterios).

Gestión de riesgos y bloqueadores escalando al lead o PO cuando corresponde.

Con el Profesor PAP:

Reporte breve de avances y aprendizajes.

Evidencias (capturas de tablero, PRs aprobados, notas de revisión).

Ajuste del plan educativo según brechas detectadas (por ejemplo, reforzar pruebas de contrato o CI/CD si hubo hallazgos).

Cambios al plan:

Replanificación de tickets ante prioridades corporativas.

Ajustes en objetivos de aprendizaje según retro de code reviews y retrospectivas.

## 2.10 Cierre del Proyecto

Corte de sprint y consolidación técnica

Freeze de trabajo: se establece la hora de cierre; no se agregan nuevas historias al sprint.

Consolidación de PRs: se finalizan y fusionan PRs que cumplan Definition of Done (DoD) y criterios de aceptación. Los cambios que no cumplen se postergan.

Etiquetado y versionado: creación de tag/versión interna o candidate (RC) según la política del repositorio.

Ejecución de pipelines finales: build limpio, pruebas unitarias/contrato, análisis estático y escaneo de seguridad. Cualquier hallazgo crítico debe resolverse o documentarse con plan de acción.

Notas de versión internas: se registran cambios relevantes, impactos, migraciones

#### Sprint Review (enfoque a resultados)

Historias completadas: conteo de historias finalizadas vs. comprometidas; se demuestra funcionalidad a stakeholders.

Métricas de desempeño: velocidad del sprint, burndown/burnup, tasa de finalización, defectos encontrados/cerrados, cobertura de pruebas y cumplimiento de calidad.

Aceptación: el PO valida cumplimiento de criterios de aceptación; historias aceptadas se cierran; las no aceptadas se replanifican.

#### Gestión de arrastre (carryover) y preparación para el siguiente ciclo

Identificación de historias no concluidas: se analiza el origen (bloqueos, alcance subestimado, dependencias, calidad).

Reestimación y división: las historias pendientes se reestiman y, si conviene, se dividen en sub-tareas o historias más pequeñas.

Priorización: el PO decide si pasan al siguiente sprint o, si es fin de PI, se programan para el siguiente PI según prioridad, dependencias y valor de negocio.

Actualización de dependencias: se documentan cambios en dependencias internas/externas y se actualiza el tablero con la información más reciente.

#### Retrospectiva (mejora continua)

Análisis de qué funcionó/qué no: prácticas técnicas (testing, revisiones de código, CI/CD) y de colaboración (comunicación, estimaciones).

Acciones concretas: 1–3 acciones priorizadas con responsables y fecha objetivo (por ejemplo, elevar cobertura mínima, acortar tiempos de revisión, clarificar DoD).

Salud del equipo y flujo: revisión de WIP, tiempos de ciclo y bloqueadores recurrentes; decisiones para reducir cuellos de botella.

#### Comunicación y documentación

Demo/Showcase: demostración a stakeholders internos con foco en resultados y valor.

Documentación técnica: actualización de notas de cambio, diagramas si aplica, guías de despliegue y pruebas.

Estado del backlog: limpieza de tickets obsoletos/duplicados y actualización de descripciones y criterios de aceptación.

#### Puente hacia el siguiente sprint/PI

Capacidad y objetivos preliminares: se estima capacidad del equipo, se borrador objetivos de sprint; si es cierre de PI, se preparan objetivos para el siguiente PI.

Decisión de release: de acuerdo con políticas internas, se define si la versión va a un ambiente superior o queda como candidate

### 3. Resultados del Trabajo Profesional

#### 3.1 Productos Obtenidos

##### Correcciones en microservicios de PCF

Objetivo: resolver defectos funcionales y técnicos priorizados en el backlog del equipo.

Alcance: correcciones en endpoints y flujos internos; manejo de errores y validaciones; ajustes menores de configuración.

Evidencias: tickets cerrados con aceptación del PO/QA; PRs aprobados; pipelines "en verde"; notas de cambio.

Estado: desplegado en ambientes internos y listo para integración conforme a la política de release

##### Suite de pruebas unitarias y de contrato (Java)

Objetivo: aumentar la cobertura, validar contratos entre microservicios y reducir regresiones.

Alcance: creación de pruebas unitarias enfocadas en lógica de negocio y de contrato para interfaces; mocks/stubs donde aplica.

Evidencias: reportes de cobertura; ejecución automática en CI; documentación de casos de prueba.

Estado: integrada en el pipeline; cobertura base establecida; monitoreo por sprint

##### Mejoras en CI/CD y aseguramiento de calidad

Objetivo: estandarizar validaciones automáticas y elevar el "calidad primero".

Alcance: jobs de build/ test, análisis estático, escaneo de seguridad, convenciones de branch/tagging, gates de calidad.

Evidencias: definición de pasos de pipeline; reportes por ejecución; checklist de DoD aplicado a PRs.

Estado: validaciones ejecutándose por PR; hallazgos críticos bloquean merge hasta remediación

##### Remediación de hallazgos (análisis estático/seguridad)

Objetivo: disminuir deuda técnica y exposición a vulnerabilidades.

Alcance: fixes de severidad crítica/alta; mitigaciones documentadas para media/baja con plan de acción.

Evidencias: reportes de herramientas internas; PRs con cambios; aceptación del revisor de seguridad/calidad.

Estado: severidad crítica/alta en 0 al cierre de sprint; plan para pendientes no críticos

##### Documentación técnica y handover

Objetivo: asegurar continuidad, trazabilidad y operabilidad.  
Alcance: notas de cambio, pasos de prueba, riesgos/impacto, criterios de aceptación; guías breves de despliegue y rollback.  
Evidencias: páginas internas actualizadas, enlaces a PRs/tickets, historial de decisiones (ADR si aplica).  
Estado: vigente al cierre; actualizada en cada release candidate

## 3.2 Estimación del Impacto

### Confiabilidad y estabilidad del servicio

Beneficio: reducción de incidentes y regresiones en microservicios críticos (PCF). Mejora de disponibilidad en ambientes internos y soporte a readiness de despliegues

Indicadores sugeridos: defectos reabiertos por sprint, tiempo medio hasta reparación (MTTR) en ambientes de prueba, porcentaje de pipelines exitosos.

### Calidad y eficiencia del desarrollo

Beneficio: menor fricción en PRs, validaciones más rápidas, detección temprana de errores.

Indicadores: cobertura de pruebas (+X pp vs. inicio), tiempo de ciclo PR (creación→merge), tasa de aprobación en primera revisión, hallazgos críticos post-merge (tendencia a la baja)

### Seguridad y cumplimiento

Beneficio: reducción de vulnerabilidades críticas/altas y cumplimiento de políticas.

Indicadores: vulnerabilidades críticas/altas en 0 al cierre de sprint; tiempo de remediación por severidad; cumplimiento de checklist DoD/seguridad

### Escalabilidad del equipo y transferibilidad

Beneficio: onboarding más ágil y menor riesgo en handovers.

Indicadores: número de módulos con documentación vigente, adopción de plantillas (PR, pruebas), tiempos de diagnóstico gracias a logs/métricas básicos.

## Proyección hacia siguiente sprint/PI

Beneficio: releases internos más predecibles, backlog saludable y priorizado.

Insumos: velocidad promedio, historias carryover reestimadas y divididas, objetivos preliminares del siguiente ciclo, dependencias documentadas

## 4. Reflexiones del alumno

### 4.1 Aprendizajes Profesionales

#### Técnicos (hard)

Java backend y diseño testeable: aplicar POO, SOLID y clean code para facilitar pruebas y mantenimiento; uso efectivo de JUnit/Mockito; prácticas de inyección de dependencias.

Pruebas unitarias y de contrato: diseño de casos por comportamiento esperado, manejo de escenarios edge y errores; integración de suites al pipeline; lectura/acción sobre reportes de cobertura.

CI/CD y control de versiones: creación y defensa de PRs con evidencias; uso disciplinado de ramas; lectura de logs de build/test; gates de calidad y seguridad

Calidad y seguridad: interpretar resultados de análisis estático; priorizar fixes por severidad; documentar mitigaciones y no regresiones

#### Suaves (soft)

Comunicación técnica en inglés: claridad en dailies, PRs y minutas; pedir y dar feedback accionable.

Trabajo en equipo: coordinación con PO/QA/DEV, negociación de alcance ante bloqueos y dependencias.

Gestión del tiempo y enfoque: dividir trabajo, limitar WIP, cerrar incrementos funcionales.

#### Sorpresas y brechas

Complejidad por dependencias entre servicios; impacto real de pequeños cambios.

Brecha CI/CD avanzada y pruebas de contrato que la universidad no cubre a profundidad; aprendidas en práctica.

#### Capacidad de gestión

Me considero capaz de llevar de punta a punta tickets de complejidad media (análisis→PR→validación→documentación) y de co-liderar técnicamente pequeñas tareas; con más ciclos, proyectable a coordinar módulos.

### 4.2 Aprendizajes Sociales

#### Contribución social y beneficiarios

Mejor confiabilidad en servicios 5G apoya comunicaciones para usuarios y organizaciones (salud, educación, seguridad), lo que tiene externalidades positivas en la vida cotidiana y la economía.

#### Bienes de carácter público/impacto económico

Mejor calidad reduce incidentes y costos operativos, mejora continuidad de servicios críticos y competitividad de operadores.

#### Cambios de visión

Mayor conciencia sobre infraestructura digital como habilitador social; responsabilidad de calidad/seguridad más allá del “código funciona en mi máquina”.

#### Iniciativa de mejora

Propuestas: plantillas de PR y checklists de pruebas; dividir historias antes para reducir carryover; documentar criterios de aceptación con ejemplos.

### 4.3 Aprendizajes Éticos

#### Honestidad y trazabilidad

Regla de oro: evidencias completas en PRs, límites claros de lo entregado y riesgos conocidos; no ocultar fallos ni “bypassear” checks del pipeline

#### Alineación de valores

Seguridad, calidad y confidencialidad coinciden con mis valores; principio de privilegios mínimos y necesidad de saber.

#### Dilemas comunes y manejo

Presión por tiempos vs. calidad: priorizar riesgos críticos, documentar deuda técnica y acordar plan de remediación; transparencia con PO/lead.

#### Datos y privacidad

Cuidado con datos sensibles en logs y documentación; anonimización; cumplimiento de políticas; evitar compartir información fuera de canales corporativos.

#### IA responsable

Uso con criterio, revisión humana, respeto a licencias y privacidad; alineamiento con políticas internas de la empresa.

#### Decisiones bajo incertidumbre

Ante hallazgos de seguridad próximos al cierre: decisión conjunta “fix now” para críticos y “document and plan” para medios/bajos, registrando impacto y responsables

#### Propósito profesional

Enfocar el desarrollo a soluciones seguras, mantenibles y con valor social; tomar decisiones basadas en principios y evidencia.

### 4.4 Aprendizajes Personales

#### Cambios personales

Mayor disciplina, resiliencia y proactividad ante bloqueadores; mejor manejo del tiempo.

#### Seguridad y madurez

Más confianza al exponer trabajo y solicitar apoyo con anticipación.

#### Autoconocimiento

Fortalezas: testing, CI/CD, documentación breve y útil. Áreas de crecimiento: observabilidad y performance.

#### Convivencia en diversidad

Mejora en comunicación intercultural, empatía y tolerancia a diferentes estilos de trabajo.

### 4.5 Tareas Aprendidas

#### Factores de éxito (para repetir)

DoD/DoR claros y acordados; historias acotadas y con criterios de aceptación medibles.

Code reviews con checklist y evidencia de pruebas; pipelines confiables; retro con 1–3 acciones concretas asignadas y con fecha

Documentación mínima viable: notas de cambio y pasos de prueba por cada PR.

#### Oportunidades de mejora (para evitar)

Comprometer alcance sin validar dependencias; historias demasiado grandes.

Retrasar pruebas hasta el final; no actualizar documentación tras cambios de última hora.

Falta de homogeneidad en PRs: resolver con plantillas y ejemplos.

## 5. Conclusiones

### Síntesis de la experiencia

Integré teoría y práctica en un entorno corporativo global, contribuyendo al mantenimiento y aseguramiento de calidad de microservicios críticos de PCF en 5G. Aprendí a cerrar sprints con disciplina: review con métricas, retro con acciones, consolidación técnica (tags/RC), pipelines finales y preparación del siguiente sprint/PI

### Lecciones más allá de lo técnico

La calidad y la seguridad son parte del proceso, no una etapa. La comunicación y la ética sostienen la confianza del equipo y de los interesados.

### Grado de satisfacción

Alto: el reto fue significativo, el esfuerzo constante y los resultados tangibles (menos hallazgos críticos, mejores pruebas, PRs más sólidos). Percibo crecimiento técnico y profesional claro.

### Proyección y siguientes pasos

Profundizar en pruebas de contrato avanzadas, observabilidad (logs/metrics/trace), performance y seguridad preventiva; seguir consolidando automatización y estandarización en CI/CD.

### Cierre personal

Esta experiencia confirmó mi interés en backend y plataformas, y me dio bases para asumir mayor responsabilidad técnica y, en el corto plazo, liderar módulos pequeños con enfoque en calidad y entrega continua.

## 6. Bibliografía y Anexos *(solo en caso de ser necesarios)*