**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**“Proyecto *DevFlow”***

**Curso:**

*Patrones de Software*

**Docente:**

*Mag. Patrick Cuadros Quiroga*

**Integrantes:**

*Akhtar Oviedo, Ahmed Hasan - (2022074261)*

*Anampa Pancca, David Jordan - (2022074268)*

*Salas Jimenez, Walter Emmanuel - (2022073896)*

**Tacna – Perú**

*2025*

**Proyecto DevFlow**

**Documento de Visión**

**Versión *1.0***

| CONTROL DE VERSIONES | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | AHAO, DJAP, WESJ | PCQ | - | 17/09/25 | Versión 1.0 |

# Índice General

[Índice General 3](#_heading=h.e513eudtg0jr)

[1. Introducción 4](#_heading=h.78rpkyrsbuqf)

[1.1 Propósito 5](#_heading=h.fgbu1opewolx)

[1.3. Referencias 6](#_heading=h.oc0rtkfynj4c)

[1.4. Visión General 6](#_heading=h.vqgsveqhk1rg)

[2. Posicionamiento 8](#_heading=h.v6dep2sxahlx)

[2.1. Oportunidad de negocio 8](#_heading=h.xqxlubnnxyao)

[2.2. Definición del problema 9](#_heading=h.77sflsjx1x0j)

[2.3 Propuesta de valor (MVP: futuro) 10](#_heading=h.krcelpvr2u9o)

[2.4 Indicadores de resultado (para la demo) 11](#_heading=h.slx8o17h80h3)

[3. Descripción de los interesados y usuarios 12](#_heading=h.opp34k5lfvog)

[3.1. Resumen de los interesados 12](#_heading=h.nfk28r5q4shx)

[3.2. Perfiles de los usuarios (MVP) 12](#_heading=h.rysf2pa4apal)

[3.3. Necesidades de interesados y usuarios 13](#_heading=h.6a7nc92d2sys)

[3.4 Matriz RACI (MVP) 13](#_heading=h.xunx3j6walfr)

[4. Vista general del producto 14](#_heading=h.tu8o508xwa6n)

[4.1 Resumen de capacidades 15](#_heading=h.v6b9c8m32l6e)

[4.2 Suposiciones y dependencias (MVP) 15](#_heading=h.36glx9d6cnm)

[4.3 Descripción de costos (orientativa para MVP) 15](#_heading=h.dzjn4jn3we8c)

[4.4 Licenciamiento e instalación 16](#_heading=h.bbuv4nfah5qt)

[4.5 Características del producto (MVP) 16](#_heading=h.vksoupe1vfia)

[4.6 Restricciones (MVP) 16](#_heading=h.bk66jtsgfruh)

[4.7 Criterios/rangos de calidad del MVP 17](#_heading=h.59hs5w389t1w)

[5. Precedencia y prioridad 18](#_heading=h.yvztgkrzhg24)

[6. Lineamientos y estándares del producto 19](#_heading=h.uhlq5loaga9i)

[7. Otros requerimientos del producto 20](#_heading=h.skpde2ix5z6k)

[Conclusiones 21](#_heading=h.uzqa6yhs2vaw)

[Recomendaciones 22](#_heading=h.2jjguxtavxdh)

**Documento de Visión**

# Introducción

El presente documento describe la visión del proyecto DevFlow, una iniciativa tecnológica orientada a integrar de manera práctica y modular todas las fases del ciclo de vida del desarrollo de software (web, móvil y de escritorio). DevFlow nace como respuesta a una necesidad cada vez más evidente: la falta de un entorno unificado que acompañe a los desarrolladores, desde la concepción de la idea hasta el monitoreo de una aplicación en producción.

En el ámbito universitario, miles de estudiantes y docentes enfrentan el reto de enseñar y aprender metodologías de desarrollo en entornos fragmentados. Las prácticas suelen enfocarse en fases aisladas, dejando de lado la visión integral del ciclo de vida. Esto genera limitaciones para aplicar de manera real y completa las metodologías ágiles, DevOps y los marcos de calidad que exigen el mercado.

En el mundo profesional, los equipos de desarrollo —ya sean startups, freelancers o grandes empresas— suelen invertir un tiempo significativo configurando manualmente tableros, repositorios, pipelines y entornos de monitoreo, duplicando esfuerzos que podrían reutilizarse o estandarizarse. Esto genera costos innecesarios, baja productividad y dificulta mantener un flujo ordenado en proyectos de mediana y gran escala.

DevFlow busca cerrar esa brecha ofreciendo un ecosistema unificado y escalable, donde tanto un estudiante que desarrolla su primer prototipo como un equipo profesional que trabaja en un producto de alto impacto puedan encontrar un marco metodológico, artefactos preconfigurados y ejemplos prácticos listos para usar. El objetivo es claro: transformar la manera en que se aprende, planifica y ejecuta el desarrollo de software, brindando trazabilidad, orden y calidad en cada fase.

## 1.1 Propósito

El propósito de este documento es establecer de forma clara los objetivos, alcances y lineamientos del proyecto DevFlow.

DevFlow no es solo una guía, sino una plataforma metodológica con entregables concretos que permiten a los usuarios:

* Planificar proyectos de software de manera organizada con tableros Trello exportables.
* Analizar requerimientos mediante UML (diagramas de clases, secuencia y componentes).
* Diseñar interfaces y prototipos funcionales en Figma/xPRE.
* Codificar sobre bases sólidas con un monorepo que incluye ejemplos de backend, frontend, móvil y desktop.
* Probar con colecciones Postman y analizar la calidad con Sonar.
* Desplegar en entornos seguros con CI/CD en Actions y Jenkins.
* Monitorear en tiempo real con Datadog y New Relic.

De esta forma, DevFlow se convierte en un acompañante del ciclo completo, aportando valor tanto a la formación académica como a la ejecución profesional de proyectos.

**1.2 Alcance**

El alcance del proyecto en su primera fase (MVP) se centra en construir un **flujo mínimo funcional** que demuestre la integración de todas las fases del ciclo de vida:

* **Planificación**: tablero en Trello con backlog, To Do, Doing, Review y Done.
* **Análisis**: creación de diagramas UML exportables (.puml).
* **Diseño**: wireframes y prototipos en Figma y xPRE, con tokens de diseño listos para reutilizar.
* **Codificación**: repositorio monorepo con scaffolds para backend (API REST en Node.js/Express), frontend web (React), móvil (React Native) y desktop (Electron).
* **Pruebas**: colección Postman (mínimo 3 endpoints) y configuración de SonarQube.
* **Despliegue**: Dockerfile, docker-compose.yml, workflow CI/CD (GitHub Actions) y Jenkinsfile declarativo.
* **Monitoreo**: dashboards en Datadog y alertas en New Relic.

Lo que quedará fuera del alcance inicial, pero se planificará para fases posteriores, incluye:

* Escalado automático en la nube.
* Analítica predictiva para métricas de calidad.
* Automatización avanzada de reportes personalizados.
* Integraciones multiusuario con permisos diferenciados.

## 1.3. Referencias

El proyecto toma como referencia las mejores prácticas de ingeniería de software, desarrollo ágil (Scrum/Kanban), DevOps y lineamientos de calidad inspirados en ISO 9001 (gestión de calidad) e ISO/IEC 27001 (seguridad). Además, se apoya en la documentación oficial de cada herramienta: Trello, PlantUML, Figma, GitHub Actions, Jenkins, Datadog y New Relic.

## 1.4. Visión General

La visión de DevFlow es convertirse en el estándar de referencia para la enseñanza, planificación y ejecución del ciclo de vida del software, integrando en un único flujo metodológico las herramientas más usadas por la industria y las más útiles en entornos educativos.

DevFlow busca ser una plataforma puente entre el mundo académico y el profesional. Para los estudiantes universitarios, DevFlow es una guía práctica que transforma los conceptos teóricos de ingeniería de software en experiencias tangibles, con entregables que pueden mostrar como evidencia de aprendizaje. Para los docentes, es una herramienta didáctica que permite estructurar prácticas basadas en proyectos reales, fomentando un aprendizaje activo y aplicado.

En el ámbito profesional, DevFlow se presenta como un marco adaptable para startups, equipos de desarrollo ágiles y empresas que buscan estandarizar procesos. Al ofrecer un ecosistema modular, los equipos pueden comenzar con lo básico y escalar hacia configuraciones más complejas según lo demande el crecimiento del proyecto.

La propuesta de valor de DevFlow se centra en tres ejes principales:

1. **Integración paso a paso:** cada fase del ciclo de vida se cubre con artefactos concretos (tableros, diagramas, prototipos, pipelines, dashboards).
2. **Accesibilidad y escalabilidad:** pensado tanto para un estudiante que desarrolla un proyecto universitario como para un equipo profesional que despliega aplicaciones en la nube.
3. **Automatización y trazabilidad:** se busca que cada acción tenga continuidad y registro, evitando el desorden y los errores humanos.

En definitiva, DevFlow no es solo un proyecto académico, sino una herramienta con potencial de convertirse en un producto con impacto real en el mercado del desarrollo de software.

# Posicionamiento

## 2.1. Oportunidad de negocio

El Sistema de Inteligencia de Negocios SEACE surge como una respuesta innovadora a la creciente necesidad de transformar los datos abiertos del Estado en información estratégica para la sociedad. En la actualidad, miles de procesos de contratación pública se registran en el SEACE, generando un volumen de información que, aunque es de acceso público, carece de una estructura adecuada para un análisis práctico y detallado. Esta situación representa una oportunidad para aprovechar el valor oculto en dichos datos mediante técnicas de integración, categorización y visualización que permitan identificar tendencias y patrones de comportamiento en el gasto estatal.

Desde el punto de vista empresarial, la oportunidad radica en que muchas compañías desconocen dónde y en qué medida el Estado invierte, perdiendo la posibilidad de participar en procesos de contratación relevantes. Una plataforma que organice los datos de manera clara y ofrezca filtros por categorías —como Ingeniería de Sistemas, Construcción, Consultoría, Salud, Educación o Transporte— permitirá a los empresarios detectar oportunidades de negocio y planificar estrategias competitivas.

Por su parte, en el ámbito académico y de investigación, la oportunidad consiste en disponer de un sistema que facilite el acceso a información confiable y categorizada, lo cual abre la posibilidad de elaborar estudios de mercado, proyectos de investigación aplicada y análisis de políticas públicas basados en evidencia.

En suma, el proyecto no solo responde a una necesidad tecnológica, sino que abre nuevas posibilidades para el sector empresarial, académico y social, consolidándose como una herramienta de valor en la gestión de la información pública.

## 2.2. Definición del problema

En la actualidad, el desarrollo de software enfrenta una problemática constante: la fragmentación de herramientas y la falta de un marco unificado que permita seguir de manera ordenada y consistente cada fase del ciclo de vida. Tanto en el ámbito académico como en el profesional, los equipos deben invertir grandes cantidades de tiempo y esfuerzo en configurar tableros, generar diagramas, diseñar interfaces, crear repositorios, establecer pipelines de integración continua y definir sistemas de monitoreo. Estas tareas, que en muchos casos deberían estar estandarizadas, se realizan de manera repetitiva y manual, ocasionando retrasos, inconsistencias y pérdida de trazabilidad.

Para los estudiantes universitarios y docentes, esta situación se traduce en experiencias de aprendizaje incompletas. Aunque los programas académicos incluyen cursos de ingeniería de software, metodologías ágiles o gestión de proyectos, en la práctica los alumnos solo alcanzan a cubrir fases aisladas, sin comprender la integración real de todo el proceso. De este modo, los proyectos quedan como ejercicios teóricos sin conexión con la realidad del mercado laboral.

En el caso de las startups y los equipos de desarrollo profesionales, el problema se refleja en tiempos prolongados de preparación inicial. Configurar entornos de desarrollo, definir pipelines de despliegue y establecer mecanismos de monitoreo son tareas que consumen semanas antes de poder enfocarse en la innovación del producto. Además, la ausencia de estandarización dificulta la colaboración entre equipos, genera duplicidad de esfuerzos y aumenta el riesgo de cometer errores humanos durante pruebas o despliegues.

La falta de un sistema integral que organice y unifique estas fases con artefactos listos para usar convierte al ciclo de vida del software en un proceso ineficiente y fragmentado. En consecuencia, los proyectos educativos no alcanzan el nivel de realismo esperado y los proyectos profesionales enfrentan retrasos, sobrecostos y pérdida de competitividad.

## 2.3 Propuesta de valor (MVP: futuro)

La propuesta de valor de DevFlow se basa en la creación de un ecosistema modular y escalable para el ciclo de vida del software. A diferencia de las soluciones que se centran en una sola parte del proceso, DevFlow ofrece una experiencia completa, desde la planificación hasta el monitoreo. Su objetivo principal es resolver los problemas de fragmentación y falta de integración que suelen enfrentar los equipos de desarrollo.

El producto mínimo viable (MVP) de DevFlow está diseñado para demostrar de manera práctica esta cadena de valor. En la fase de planificación, utiliza un tablero de Trello preconfigurado para la gestión de tareas. Para el análisis, se integra con diagramas UML en formato .puml. El diseño se maneja con wireframes de Figma y xPRE, y se acompaña de tokens de diseño para asegurar la consistencia visual. En cuanto a la codificación, DevFlow provee un monorepo con ejemplos de código para diversas plataformas (backend, frontend, móvil, etc.). Las pruebas se realizan a través de colecciones de Postman y se utiliza SonarQube para la calidad del código. Finalmente, el despliegue se automatiza con archivos como Dockerfile y workflows de GitHub Actions, mientras que el monitoreo se habilita con dashboards básicos en Datadog y alertas en New Relic.

El valor de DevFlow radica en su capacidad para hacer que el proceso de desarrollo de software sea accesible y reproducible. Este enfoque permite que tanto estudiantes como profesionales puedan recorrer el ciclo de vida completo de un proyecto sin la necesidad de reconfigurar todo desde cero. Esto no solo reduce los tiempos de inicio, sino que también garantiza un nivel de calidad constante, eliminando la necesidad de reinventar la rueda en cada nuevo proyecto.

## 2.4 Indicadores de resultado (para la demo)

* El éxito del proyecto DevFlow se medirá a través de indicadores concretos que reflejen su impacto tanto en el ámbito académico como en el profesional. En términos de accesibilidad, se buscará que un usuario pueda crear un proyecto inicial en menos de treinta minutos utilizando las plantillas y artefactos proporcionados. En cuanto a usabilidad, el objetivo será alcanzar un nivel de satisfacción superior al ochenta y cinco por ciento en pruebas piloto con estudiantes y profesionales, evaluando claridad, facilidad de uso e integración de herramientas.

La cobertura constituirá otro indicador clave, medida por la cantidad de herramientas integradas en el MVP, siendo necesario que al menos siete estén operativas de forma incremental en la primera versión para demostrar la cadena mínima de valor. En automatización, se evaluará que los pipelines de integración y despliegue continuo alcancen una tasa de éxito igual o superior al noventa por ciento en las ejecuciones de construcción, pruebas y despliegue.

Desde la perspectiva académica, se considerará un éxito la cantidad de estudiantes y docentes que adopten DevFlow como guía para proyectos de curso o trabajos de investigación. En el ámbito profesional, el impacto se medirá a partir de startups y empresas que implementen DevFlow como marco metodológico, cuantificando el ahorro de tiempo en la preparación inicial y el aumento en la calidad de los entregables.

**Principales indicadores del éxito de DevFlow:**

* **Accesibilidad:** creación de un proyecto inicial en menos de 30 minutos.
* **Usabilidad:** satisfacción superior al 85% en pruebas piloto.
* **Cobertura:** integración mínima de siete herramientas en el MVP.
* **Automatización**: éxito igual o mayor al 90% en pipelines CI/CD.
* **Impacto académico:** número de estudiantes y docentes que adopten DevFlow.
* **Impacto profesional:** startups y empresas que lo utilicen, con ahorro de tiempo y mejora en la calidad.

# 3. Descripción de los interesados y usuarios

## 3.1. Resumen de los interesados

Los interesados en **DevFlow** son diversos y abarcan tanto el mundo académico como el profesional:

* **Estudiantes universitarios**: buscan aprender de manera práctica, con entregables que les permitan demostrar competencias técnicas.
* **Docentes e instituciones educativas**: necesitan guías estructuradas y ejemplos claros para implementar metodologías modernas en clase.
* **Startups y freelancers**: requieren rapidez en la configuración inicial para centrarse en la innovación del producto.
* **Empresas de software**: buscan estandarizar procesos y reducir errores en proyectos grandes.
* **Profesionales DevOps**: interesados en contar con pipelines y dashboards configurados desde el inicio.

## 3.2. Perfiles de los usuarios (MVP)

Dentro del conjunto de interesados, se definen perfiles de usuarios concretos que interactúan directamente con el sistema y que determinan los requerimientos de diseño y funcionalidad.

* Usuario Estudiante/Docente: desea comprender la integración de todo el ciclo de vida del software.
* Usuario Startup: requiere lanzar un producto mínimo viable en semanas, no meses.
* Usuario Profesional/Empresarial: busca calidad, estandarización y escalabilidad en sus procesos.
* Usuario DevOps: necesita pipelines CI/CD sólidos y monitoreo avanzado.

## 3.3. Necesidades de interesados y usuarios

| **Parte interesada / Usuario** | **Necesidad clave (MVP)** | **Cómo lo cubre el MVP** |
| --- | --- | --- |
| Estudiante | Ver ciclo completo aplicado | Tablero, diagramas, prototipos, repositorio funcional |
| Docente | Enseñar con herramientas reales | Artefactos exportables y ejemplos reproducibles |
| Startup | Rapidez en la configuración | Monorepo y pipelines listos |
| Empresa | Estandarización | Flujos definidos, reglas de calidad y monitoreo |
| DevOps | Automatización | Jenkinsfile, Actions, Dockerfile |

## 3.4 Matriz RACI (MVP)

**R =** Responsable (ejecuta), **A =** Aprobador (decide)**, C =** Consultado**, I =** Informado

| **Actividad** | **RSU/Proyección** | **Docente** | **Equipo dev** | **Startup** |
| --- | --- | --- | --- | --- |
| Definir criterios de demo | R | A | C | I |
| Diseño funcional mínimo | C | A | R | I |
| Desarrollo scaffolding | I | C | R | A |
| Pruebas | R | C | A | C |
| Retroalimentación | A | A | C | C |

# 4. Vista general del producto

DevFlow, en su primera fase, se concreta como un ecosistema modular de plantillas, repositorios y ejemplos configurados que permiten recorrer de principio a fin el ciclo de vida del software.

El producto inicial no pretende reemplazar herramientas ya existentes, sino integrarlas en un flujo ordenado y automatizado, demostrando cómo se conectan la planificación, el análisis, el diseño, la codificación, las pruebas, el despliegue y el monitoreo en un único caso práctico.

El objetivo es que los usuarios puedan validar la experiencia completa del desarrollo de software en menor tiempo y con menos fricción, logrando resultados tangibles que se adapten tanto a entornos académicos como profesionales.

## 4.1 Resumen de capacidades

**Incluido en el MVP (esta entrega):**

* Visualización en repositorios organizados y tablas simples.
* Exportación local de entregables en Excel y PDF.
* Interfaz amigable y clara.
* Sistema de mensajes para errores y estados de procesos

**Planeado para fases futuras (no incluido en el MVP):**

* Autenticación avanzada con roles.
* Reportes dinámicos y personalizables.

## 4.2 Suposiciones y dependencias (MVP)

* Dependencia de conexión a internet estable.
* Uso de navegadores modernos y dispositivos compatibles.
* Stack técnico inicial: React/Flutter, Node.js, SQL Server/MySQL.
* Seguridad básica sin almacenamiento de datos sensibles.

## 4.3 Descripción de costos (orientativa para MVP)

* **Costos generales:** papelería/materiales de apoyo y utilitarios para documentación (manual de usuario, guías).
* **Costos operativos:** consumo eléctrico, pruebas en dispositivos físicos y (si aplica) un repositorio/hosting interno para documentación.
* **Costos de personal:** dedicación del equipo de desarrollo (estudiantes) durante ~1 mes con ~2 semanas efectivas de implementación y pruebas.
* **Total (estimativo):** suma de los rubros anteriores. *(En la versión financiera se detalla CAPEX/OPEX y horas hombre por rol.)*

| *Concepto* | *Costo* |
| --- | --- |
| Costos generales | S/. 60.00 |
| Costos operativos | S/. 92.00 |
| Costos de personal | S/. 4229.90 |
| Total | S/. 4381.00 |

## 4.4 Licenciamiento e instalación

* Uso académico y demostrativo, no comercial en la fase inicial.
* Licenciamiento abierto para fines educativos e investigativos.
* Despliegue en servidores de prueba con acceso mediante navegador.

## 4.5 Características del producto (MVP)

* **Interfaz clara y funcional**: búsqueda → resultados → exportación.
* **Diseño visual consistente**: uso de colores corporativos y componentes responsivos.
* **Contenido entendible**: mensajes claros de estado y errores.
* **Soporte básico**: manual de usuario en PDF con pasos de búsqueda y exportación.
* **Optimización ligera**: app funcional en PCs y móviles de gama media.
* **Seguridad inicial**: login básico y protección contra accesos no autorizados.

## 4.6 Restricciones (MVP)

* Sin dashboards ni BI avanzado en la primera versión.
* Dependencia total de conexión a internet.
* Login básico, no institucional.
* Sin sincronización multiusuario avanzada.
* Escalabilidad limitada a datasets medianos (<50,000 registros).

## 4.7 Criterios/rangos de calidad del MVP

Para garantizar la validez del MVP, se establecerán criterios de calidad que permitan medir su desempeño en las fases iniciales de uso. En cuanto a experiencia de usuario, se buscará alcanzar una satisfacción mínima del ochenta por ciento en pruebas piloto, asegurando además que los mensajes de búsqueda y error sean claros y sin confusiones.

En términos de desempeño, el sistema deberá responder en menos de dos segundos en consultas de hasta diez mil registros, y se espera que al menos el noventa y cinco por ciento de las exportaciones realizadas sean correctas. Respecto a confiabilidad, se establece como meta una tasa de fallos críticos igual a cero durante las demostraciones, con mecanismos de recuperación guiada ante fallos de conexión o búsquedas vacías.

Finalmente, en materia de seguridad y privacidad, el MVP no almacenará datos sensibles, restringiendo el acceso únicamente a usuarios con credenciales de prueba, lo que asegura un entorno controlado y sin riesgos de exposición de información crítica.

**Puntos clave:**

* Experiencia de usuario: ≥80% de satisfacción y mensajes claros.
* Desempeño: tiempo de respuesta <2 segundos y exportaciones ≥95% correctas.
* Confiabilidad: tasa de fallos críticos = 0 en la demo y recuperación guiada.
* Seguridad: sin almacenamiento de datos sensibles y acceso limitado con credenciales de prueba.

# 5. Precedencia y prioridad

La estrategia de DevFlow establece prioridades claras para asegurar una implementación incremental exitosa. En primer lugar, se dará prioridad absoluta a la integración de herramientas de planificación, análisis y diseño, ya que estas constituyen la base para estructurar cualquier proyecto. Posteriormente, se añadirá la fase de codificación con repositorios preconfigurados, seguida de la incorporación de pruebas automatizadas que aseguren la calidad del código desde etapas tempranas.

Una vez asegurada la calidad, se integrarán los pipelines de despliegue, que permitirán a los equipos contar con procesos de integración continua y entrega continua confiables. Finalmente, se incorporará la capa de monitoreo con Datadog y New Relic, garantizando que las aplicaciones desplegadas puedan ser supervisadas y mejoradas de manera proactiva. Esta progresión garantiza que los usuarios experimenten resultados tangibles en cada sprint de desarrollo, validando la propuesta de valor y asegurando un aprendizaje incremental.

**Priorización:**

* **P0 (imprescindible):** login seguro, carga de datos, filtrado por categorías.
* **P1 (fluidez):** búsquedas rápidas (<2 s), exportación básica de reportes.
* **P2 (usabilidad):** dashboards iniciales, manual breve de uso, accesibilidad mínima.
* **P3 (pulido):** identidad visual, optimización de la experiencia de usuario.

**Criterios de aceptación (demo):**

* Login y cierre de sesión totalmente operativos.
* Ejecución de tareas básicas con ≥95% de precisión en pruebas.
* Tiempo de respuesta <2 segundos en creación, consulta o exportación de artefactos.
* Exportación de entregables en PDF y Excel sin errores ni pérdidas de información.
* Mensajes claros en casos de acceso inválido, fase inexistente o error de conexión.
* Sin pérdida de registros ni fallos en operaciones críticas del flujo de trabajo.

# 6. Lineamientos y estándares del producto

El sistema DevFlow se fundamenta en lineamientos de seguridad, accesibilidad y calidad que aseguran la confiabilidad de la información procesada y presentada.

**Legales y normativos:**

* Cumplimiento de la **Ley N.° 29733 – Ley de Protección de Datos Personales**: no se almacenan datos personales sensibles, solo información pública de procesos de contratación.
* Alineamiento con la **Ley N.° 30096 – Delitos Informáticos**: se incluyen validaciones de integridad para evitar la manipulación indebida de datos y accesos no autorizados.

**Accesibilidad y usabilidad:**

* Interfaz con contraste legible, tipografía clara y menús navegables.
* Flujo intuitivo: Login → Filtro de búsqueda → Resultados → Reportes/Dashboards.

**Seguridad en el sistema:**

* Control de acceso por login con encriptación de contraseñas (hash + sal).
* Sesiones con cierre automático tras inactividad prolongada.
* Logs técnicos para depuración, sin almacenar información sensible.

**Calidad y mejora continua:**

* Evidencia de pruebas funcionales (búsqueda, filtros, reportes).
* Control de versiones del sistema (Git u otro repositorio).
* Buenas prácticas inspiradas en **ISO 9001** (gestión de calidad) e **ISO/IEC 27001** (seguridad de la información).
* Pruebas de rendimiento (tiempo de respuesta <2 s), estabilidad (sin fallas críticas en demo) y accesibilidad básica.

# 7. Otros requerimientos del producto

Para garantizar su operatividad, DevFlow requiere acceso a internet estable, navegadores actualizados y entornos locales compatibles con Docker. Cada artefacto estará acompañado de documentación en formato de guías breves y README, lo que asegura su fácil adopción y reutilización. La plataforma ha sido diseñada para ser escalable, de modo que los mismos artefactos que sirven para un prototipo académico puedan evolucionar hacia implementaciones empresariales en la nube.

**Esenciales del MVP:**

* **Compatibilidad**: navegadores actualizados en equipos de gama media o superior.
* **Instalación**: despliegue en localhost o servidor institucional.
* **Soporte**: manual breve para usuarios y administradores (instalación, uso, resolución de errores).
* **Almacenamiento**: base de datos relacional (MySQL o SQL Server).
* **Limitaciones iniciales**: sin notificaciones automáticas, sin sincronización en tiempo real, sin módulos predictivos avanzados.

# 

# Conclusiones

DevFlow representa una propuesta innovadora y disruptiva en el ámbito del desarrollo de software. Al integrar de forma modular y escalable todas las fases del ciclo de vida, se posiciona como una herramienta única que atiende tanto a las necesidades académicas como profesionales. En el plano educativo, se convierte en un aliado estratégico para universidades y docentes que buscan enseñar con metodologías modernas y herramientas de la industria. Para los estudiantes, constituye una oportunidad de aprender haciendo, con entregables reales que fortalecen sus competencias.

En el plano profesional, DevFlow responde a los desafíos de startups, freelancers y empresas consolidadas que requieren ahorrar tiempo, estandarizar procesos y garantizar calidad. Su enfoque incremental permite obtener resultados tangibles desde el primer sprint, validando ideas y acelerando la entrega de valor. Con un MVP que recorre de la planificación al monitoreo, DevFlow demuestra que es posible transformar la forma en que se conciben y ejecutan proyectos de software.

En definitiva, DevFlow no es solo un proyecto académico, sino una herramienta con proyección real en el mercado. Su capacidad para conectar teoría y práctica, academia y empresa, lo convierte en un marco integral que puede escalar desde el aula universitaria hasta los equipos de desarrollo más exigentes, consolidándose como un referente en el proceso de creación de aplicaciones web, móviles y de escritorio.

# Recomendaciones

1. **Validar primero en entornos académicos controlados** Antes de intentar escalar a startups o empresas, se recomienda consolidar DevFlow en universidades y cursos. Esto permitirá obtener feedback estructurado de estudiantes y docentes, validar el modelo de integración de herramientas y corregir errores con menor riesgo.
2. **Construir el MVP con un enfoque modular e incremental** DevFlow integra muchas herramientas (Trello, UML, Figma, CI/CD, monitoreo). Para evitar sobrecarga, se sugiere priorizar y liberar el sistema por fases, empezando con planificación y análisis, luego diseño, codificación y finalmente despliegue/monitoreo. Esto mostrará resultados rápidos y medibles.
3. **Enfocarse en la experiencia de usuario (UX) desde el inicio** Como DevFlow busca servir tanto a estudiantes como a profesionales, se recomienda invertir esfuerzo en la claridad de la interfaz, manuales breves y mensajes de error entendibles. Una mala experiencia en las primeras pruebas podría reducir la adopción temprana.
4. **Definir métricas de impacto académico y profesional** No solo basta con que el sistema funcione, sino que debe demostrar valor. Se recomienda medir:
   1. Número de proyectos académicos que lo usan en un semestre.
   2. Reducción de tiempo en la preparación inicial de un proyecto.
   3. Nivel de satisfacción en pruebas piloto (objetivo ≥85%).
5. **Planificar la sostenibilidad y evolución del proyecto** Para que DevFlow no se quede en un prototipo, es clave trazar una ruta clara de escalamiento: integración progresiva con bases en la nube, soporte multiusuario avanzado y funciones premium para uso profesional. Esto permitirá que pase de un producto académico a un marco metodológico aplicable en la industria.