**Propuesta de Automatización Aduanera**

**“Los Libertadores”.**

**Asignatura:** Ingeniería de Software.

**Profesor:** Marco Antonio Perelli Henríquez.

**Integrantes**: Cesar Henríquez.

Diego Miranda.

Macarena Cruz.

**Fecha:** 24/06/2025.

**Índice**

[**1.- Introducción. 2**](#_kzht0vbqyie7)

[1.1.- Contexto del Problema. 2](#_ahc8375duc4g)

[1.2.- Objetivos del Sistema. 2](#_9cj7sep6bgid)

[1.3.- Alcance del Proyecto. 2](#_mpkbbws9nci6)

[1.4 - Desarrollo del Proyecto 2](#_1b9yp6orppqy)

[**2.- Requisitos de Alto Nivel. 3**](#_81cbnsgn2dsz)

[2.1.- Priorización de Requisitos de Alto Nivel. 3](#_ds5h9bw67e5u)

[2.2.- Stakeholders y Roles. 3](#_s5hcz3xrymkg)

[2.3.- Estándares de Cumplimiento. 4](#_jfws3r7o999f)

[2.4.- Seguridad en normas ISO. 4](#_uoh8qua1l8u9)

[2.5.- ISO/IEC 27001 – Sistema de Gestión de Seguridad de la Información (SGSI). 4](#_cnh2uetbv4ny)

[2.5.1 Componentes clave de ISO 27001. 4](#_cwdinag1zcn0)

[**3.- Requisitos Funcionales. 5**](#_y0f4c8sp58sy)

[**4.- Requisitos No Funcionales. 7**](#_ce1wryw08aob)

[**5.- Carta Gantt. 9**](#_ggxjfrrf4hi7)

[5.1.- Fases aplicadas para cumplir con el ciclo de vida del software. 9](#_y1kpxlz0njk4)

[**6.- Asegurando la Calidad del Sistema. 10**](#_ev55pyyqkxss)

[**7.- Metodología Híbrida. 10**](#_h3fps2uclghc)

[**8.- Modelamiento de software usando estándares UML. 12**](#_f8gsdg7c38ux)

[8.1.- Diagrama de Casos de Uso (Vista de Escenarios). 12](#_ey4nhf1fu9rq)

[8.2.- Diagrama de Clases (Vista Lógica). 13](#_ynb49hj5ojam)

[8.3.- Diagrama ER (Vista Lógica). 14](#_g763o7iv9cvg)

[8.4.- Diagrama de paquete (Vista Despliegue). 16](#_979up3tsd6n9)

[8.5.- Diagrama de Componentes (Vista Despliegue). 17](#_vz97651j63p6)

[8.6-. Diagrama de actividad (Vista de procesos). 18](#_uq2gstp9vn4c)

[8.7.- Diagrama de procesos (Vista de procesos). 18](#_hbzd86al5lvs)

[8.8.- Diagrama de despliegue (Vista física). 19](#_pp2amc5nhsk)

[**9.- Estándares de Calidad en el Diseño de Software. 20**](#_33t1bu17oq56)

[**10. Diseño de Casos de Prueba 21**](#_3c5q4gbt1xkk)

[**11. Ejecución de Pruebas 22**](#_1pzyvclq36jb)

[**12. Casos de prueba 22**](#_ocb5ulaqovxx)

[12.1 Inicio Sesión 22](#_v09ax34q6b4h)

[12.2 Modulo Tramites en Linea (Principal) 23](#_nbtfkj77p5l4)

[12.3 Modulo Documentos vehiculares 24](#_b4gbk2yfgqc1)

[12.4 Modulo SAG 26](#_xacy33rws9sv)

[12.5 Modulo Control de menores 27](#_rw91zpqluder)

[12.6 Modulo Protocolo de seguridad en caso de Credenciales Inválidas 29](#_822k7hkz215g)

[**12.7. Integración de Servicios 30**](#_rjgmm3hrkmcl)

[**13 Cronograma del proyecto 30**](#_tnjbslze2eck)

[**14. Análisis de Riesgos 31**](#_tffgcivxnz65)

[**16.- Conclusión. 32**](#_mck3zii5ngjn)

# 1.- Introducción.

## 1.1.- Contexto del Problema.

En nuestra visita al Paso Los Libertadores identificamos 3 problemas principales:

1. Demoras excesivas:
   * En temporada alta (diciembre-enero), las filas alcanzan hasta 20 horas de espera.
   * Ejemplo: El 15/12/2024, 350 vehículos estuvieron más de 10 horas en cola.
2. Procesos manuales:
   * Los funcionarios deben verificar 5 documentos diferentes por persona.
   * El 60% de los trámites presentan errores por datos mal ingresados.
3. Falta de información:
   * Encuesta a 100 usuarios: 78% no sabía qué documentos necesitaba.
   * Solo 12% conocía el portal web actual.

## 1.2.- Objetivos del Sistema.

Nuestro software buscará:

* Reducir tiempos de espera a menos de 1 hora (actual: 8-20 hrs)
* Automatizar el 90% de las validaciones documentales.
* Integrar en un solo sistema:
* Aduana Chile.
* SAG (Servicio Agrícola y Ganadero).
* PDI (Policía de Investigaciones).

## 1.3.- Alcance del Proyecto.

Incluye:

* Plataforma web y móvil para gestión de trámites.
* Integración con:
  + SAG (validación de productos agrícolas).
  + PDI (verificación de antecedentes).
  + Aduana Argentina (intercambio de datos).
* Dashboard para monitoreo en tiempo real.

## 1.4 - Desarrollo del Proyecto

Se desarrolla una solución tecnológica operativa para automatizar procesos aduaneros en el Paso Fronterizo Los Libertadores. Esta plataforma permite autenticar usuarios, registrar vehículos, validar inspecciones SAG y controlar menores acompañados en el cruce fronterizo.

Esta etapa final se centra en validar y documentar, mediante pruebas técnicas, que todas las integraciones desarrolladas funcionan correctamente en un entorno real. Para ello se utilizó la herramienta profesional Postman para el diseño, ejecución y evidencias de pruebas.

# 

# 2.- Requisitos de Alto Nivel.

### 2.1.- Priorización de Requisitos de Alto Nivel.

**Objetivo Principal:**Optimizar el proceso de control aduanero en pasos fronterizos entre Chile y Argentina, **reduciendo tiempos de espera** mediante automatización, integración de sistemas y mejora en la gestión de flujos.

**Prioridades Clave:**

1. **Muy Altos** 
   * Automatización de validación documental (menores, vehículos, mascotas).
   * Integración en tiempo real con SAG, PDI y Aduana Argentina.
   * Rendimiento rápido 3 segundos por operación y seguridad de datos (Ley 19.628).
2. **Altos** 
   * Sistema de turnos virtuales con geolocalización.
   * Dashboard para monitoreo operativo en tiempo real.
   * Escalabilidad para manejar picos de demanda (ej: temporada alta).
3. **Medios/Bajos** 
   * Registro de usuarios con autenticación avanzada.
   * Compatibilidad con múltiples navegadores/dispositivos.

**Excluidos:**

* Integración con otros países (fuera del alcance inicial).

**Impacto Esperado:**

* **Reducción del 70% en tiempos de espera** Paso Los Libertadores.
* **Eliminación de redundancias** en trámites binacionales.
* **Mayor transparencia** para usuarios y autoridades.

**Enfoque:** Solución mínima viable (MVP) centrada en los requisitos críticos para abordar las demoras urgentes.

## 2.2.- Stakeholders y Roles.

|  |  |  |
| --- | --- | --- |
| **Actor** | **Responsabilidad** | **Necesidad Principal** |
| Viajeros | Subir documentos | Reducción de tiempo de espera |
| Agentes Aduaneros | Validar trámites | Herramientas de verificacion rapida |
| Administradores | Monitorear operaciones | Reportes estadísticos |

### 

### 

## 2.3.- Estándares de Cumplimiento.

* **Seguridad:** Ley 19.628 (protección de datos).
* **Accesibilidad:** Sistema Inclusivo.
* **Rendimiento:** 99.5% disponibilidad mensual.

## 2.4.- Seguridad en normas ISO.

Es necesario para que un sistema funcione con eficacia, una seguridad sólida para mantener la privacidad de los datos y que no exista filtración. Para esto, se tomará en cuenta la siguiente norma ISO.

## 2.5.- ISO/IEC 27001 – Sistema de Gestión de Seguridad de la Información (SGSI).

Esta norma define cómo establecer, implementar, mantener y mejorar un sistema de gestión para proteger la información, y se basa en tres pilares:

1. **Confidencialidad**: Que la información solo sea accesible por quien debe.
2. **Integridad**: Que la información esté completa y no sea alterada sin autorización.
3. **Disponibilidad**: Que la información esté accesible cuando se necesite.

### 2.5.1 Componentes clave de ISO 27001.

* **Evaluación de riesgos**
* Identificar qué amenazas existen para la información y cómo mitigarlas.
* **Políticas de seguridad**
* Definir reglas claras sobre cómo se maneja la información.
* **Controles de acceso**
* Asegurar que solo usuarios autorizados puedan acceder a sistemas o datos sensibles.
* **Gestión de incidentes**
* Establecer un proceso para responder a incidentes de seguridad (hackeos, filtraciones, etc.).
* **Seguridad en el desarrollo de software**
* Incluir medidas de seguridad desde las primeras etapas del ciclo de vida del software (esto se complementa con ISO/IEC 27034, enfocada en aplicaciones).
* **Auditorías internas**
* Revisión periódica para asegurarse de que se están cumpliendo las políticas.

# 3.- Requisitos Funcionales.

| **ID** | **Requisito Funcional** | **Descripción** | **Prioridad** | **Motivo** |
| --- | --- | --- | --- | --- |
| **1** | Registro de Usuario | El sistema debe permitir un registro de manera segura | Media | El registro seguro es base, pero no impacta directamente en demoras. |
| **2** | Gestión de documentos | Automatizar la validación de documentos para menores, mascotas y vehículos | Muy alta | Automatizar documentos es clave para agilizar trámites (menores, vehículos, mascotas). |
| **3** | Generación de reportes | Debe exportar reportes en PDF y Excel | Media | Reportes son útiles para análisis posteriores, no para agilizar trámites. |
| **4** | Integración con SAG | El sistema debe permitir intercambiar datos con SAG y PDI para agilizar revisiones | Muy alta | Integración con SAG/PDI evita cuellos de botella en revisiones físicas. |
| **5** | Interconexión | Debe intercambiar datos con aduanas Argentinas para evitar redundancias | Muy alta | Interconexión con Argentina elimina redundancias y demoras por validación manual. |
| **6** | Notificación en tiempo real | Alertar por demoras, documentos, faltantes y si ya están aprobados los documentos | Alta | Notificaciones en tiempo real previenen colas por documentos faltantes. |
| **7** | Consulta de trámites | Los usuarios deben poder ver el estado de sus procesos | Baja | Consulta de trámites es reactiva (no previene demoras). |
| **8** | Sistema de turno virtual | Reducir colas mediante asignación de turno en linea | Media-alta | Turnos virtuales reducen congestión en pasos fronterizos (ej: Los Libertadores). |
| **9** | Digitaciones de formularios | Permitir el llenado de formularios | Baja | La digitación de formularios ya existe parcialmente (no es innovación crítica). |
| **10** | Dashboard administrativo | Visualización de métricas de flujo de personas y vehículos | Media | Dashboard ayuda a redistribuir recursos en tiempo real según flujo. |

# 4.- Requisitos No Funcionales.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Requisito NO Funcional** | **Descripción** | **Prioridad** | **Motivo** |
| **1** | Rendimiento | El sistema debe responder en menos de 3 segundos por solicitud | Muy alta | Rendimiento <3 segundos es vital para no replicar las demoras actuales. |
| **2** | Seguridad | Cumplir con normativas de protección de datos (Ley Chilena 19.628). | Muy alta | Cumplir la ley 19.628 evita sanciones legales por manejo de datos sensibles. |
| **3** | Disponibilidad | Disponibilidad mínima del 99% | Media | La disponibilidad del 99% asegura operatividad continua en frontera. |
| **4** | Escalabilidad | Capacidad de escalar en temporadas de alta demanda | Alta | Escalabilidad es clave para temporadas altas (ej: flujo aumenta 180% en verano). |
| **5** | Usabilidad | Interfaz intuitiva para usuarios de bajo nivel técnico (incluyendo tutoriales) | Alta | Usabilidad intuitiva reduce tiempo de capacitación para turistas/funcionarios. |
| **6** | Accesibilidad | Compatible con lectores de pantalla y navegación por teclado | Media | Accesibilidad es importante, pero no crítica para el flujo principal. |
| **7** | Mantenimiento | Código modular para futuras actualizaciones. | Media | Código modular facilita el mantenimiento, pero no impacta en la operación inmediata. |
| **8** | Compatibilidad | Para funcionar con navegadores actuales como: Chrome Safari, Edge entre otras | Media | La compatibilidad con navegadores amplía el acceso a usuarios diversos. |
| **9** | Normativas | Cumplir con estándares IEEE 830 para especificación de requisitos. | Baja | La normativa IEEE 830 es relevante para documentación, no para funcionalidad. |
| **10** | Identidad Institucional | Usar colores e imagen corporativa del servicio de aduanas | Baja | Identidad institucional es deseable, pero no urgente para resolver demoras. |

# 5.- Carta Gantt.

El objetivo principal de la Carta Gantt es planificar y visualizar el desarrollo del caso “Automatización Aduanera Los Libertadores”.

Tiene como finalidad la implementación de un sistema para **automatizar** los controles fronterizos de la aduanera “Los Libertadores”. Este proyecto se llevará a cabo durante el período de 146 días y está compuesto por una serie de tareas y actividades que deberán ejecutarse de manera ordenada y dentro de los plazos establecidos.

A través de esta herramienta, se busca visualizar claramente las actividades, duración, responsables, costos y entregables durante el proceso del proyecto.

Figura 1. “Carta Gantt de automatización de Aduanera Los Libertadores”.

## 5.1.- Fases aplicadas para cumplir con el ciclo de vida del software.

1. Análisis de Procesos.
   * Acción: Cronometrar tiempos reales.
2. Diseño.
   * Acción: Crear prototipos para:

* Agentes: Pantallas con alertas visuales (ej: "Falta certificado SAG en camión #123").
* Viajeros: App con checklist interactivo (ej: "Para mascotas necesitas 3 documentos ").

1. Desarrollo en Paralelo.
   * Equipos trabajando simultáneamente:

* Equipo A: API para validar documentos con PDI (2 segundos por consulta).
* Equipo B: Sistema de turnos con geolocalización (ej: "Tu turno es a las 15:30 cuando llegues a 20 km de la frontera").

1. Pruebas con Simulador.
   * Escenario:

* Simular 1,200 vehículos en 4 horas (como en verano).
* Fallos intencionales: Documentos ilegibles o fotos oscuras.
  + Métrica clave: El sistema debe procesar 95% de casos sin intervención humana.

1. Implementación Gradual.
   * Fases:

* Semana 1: Carril 5 (solo camiones).
* Semana 3: Carriles 3 y 4 (vehículos particulares).  
  Regla: Si hay >5% de errores, pausar expansión.
  + Dashboard crítico:
* Tiempo promedio por vehículo (meta: <5 min).
* Documentos más rechazados (ej: 22% son permisos de mascotas mal escaneados).

1. 🔧 Mantenimiento Ágil.
   * Ejemplo realista:

* Si Argentina cambia requisitos para frutas: Actualizar sistema en menos de 24 hrs.
* Si usuarios reportan confusión: Añadir tutoriales en video a la app.

## 5.2 EDT – Estructura Desglosada del Trabajo

Tabla

El contenido generado por IA puede ser incorrecto.

# 6.- Asegurando la Calidad del Sistema.

La calidad en este proyecto es esencial para garantizar que el sistema funcione de manera eficiente y cumpla con las expectativas de los usuarios. No solo se trata de que el sistema haga lo que se espera, sino también, de que sea fácil de usar, seguro y capaz de adaptarse a futuras necesidades. Para lograr esto, nos aseguraremos de que cada fase del proyecto pase por pruebas detalladas, desde la validación de los requisitos hasta la verificación del desempeño bajo condiciones reales.

Además, nos enfocaremos en seguir buenas prácticas de desarrollo y realizar revisiones constantes del código para evitar errores y mejorar continuamente. El objetivo es entregar un sistema que no solo cumpla con lo prometido, sino que también brinde una experiencia satisfactoria y sea confiable a largo plazo.

# 7.- Metodología Híbrida.

La **metodología híbrida** es adecuada para el proyecto porque combina lo mejor de dos mundos: la flexibilidad de Agile (para desarrollar software rápido) y la estructura de Cascada (para cumplir normas gubernamentales).

¿Para qué funciona?

1. **Para el equipo técnico (Ágil):**
   * Desarrollan el software en sprints cortos (ej: cada 2 semanas), probando mejoras rápido.
   * Ejemplo: Si la conexión con el SAG falla, lo ajustan en días, no meses.
2. **Para trámites legales (Cascada):**
   * Las integraciones con el gobierno (SAG, PDI) requieren aprobaciones formales y no pueden cambiar sobre la marcha.
   * Ejemplo: La API con Aduana Argentina sigue un protocolo fijo, pero el diseño de la app para conductores sí se mejora semana a semana.

### Ejemplo práctico del proyecto:

|  |  |  |
| --- | --- | --- |
| **Parte del Proyecto** | **Metodología Usada** | **Razón** |
| Desarrollo de la app móvil | Ágil (Scrum) | Para añadir features basadas en feedback real de usuarios. |
| Integración con la PDI | Cascada | Porque requiere contratos y pruebas formales que no pueden cambiar. |
| Validación de documentos con IA | Ágil | Para entrenar el modelo con nuevos formatos de permisos semana a semana. |
| Cumplimiento de ISO 27001 | Cascada | Los controles de seguridad deben seguir un checklist fijo. |

Beneficios clave de metodología híbrida:

* **Velocidad + Control:** Avanzas rápido en lo técnico, pero sin saltarte normas.
* **Menos riesgo:** Si un sprint falla (ej: el módulo de mascotas), no afecta los procesos legales ya aprobados.
* **Adaptable:** Si Argentina cambia requisitos, ajustas solo las partes afectadas.

# 8.- Modelo de Arquitectura 4+1 y Diagramas UML.

Con el fin de estructurar el diseño del sistema propuesto para la automatización aduanera del Paso Los Libertadores, se empleó el modelo arquitectónico **4+1 de Kruchten**. Este modelo permite analizar el sistema desde cinco perspectivas fundamentales para garantizar su funcionamiento, escalabilidad, seguridad y adecuación al usuario final.

A continuación, se describe cómo cada vista se representa en el presente proyecto, acompañada de los diagramas correspondientes que ya fueron desarrollados:

## 8.1.- Vista Logica (Vista de Diseño)

### 1. Diagrama de Clases

El **Diagrama de Clases** es como un plano de los elementos principales de nuestro sistema de aduanas y sus funciones. Forma parte de la **Vista Lógica** (cómo se configuran los datos y las acciones del sistema) en el modelo 4+1. Aquí tienes una versión rápida:

* **¿Qué es?**

Muestra objetos (como viajeros o documentos) como cuadros, con sus detalles (como nombres o identificaciones) y acciones (como registrarse). También muestra cómo están conectados.

* **¿Qué contiene?**
  + **Viajero**:
* **Detalles**: ID, rut, nombre, correo electrónico.
* **Acciones**: Inscribirse, enviar documentos, reservar turno.
  + **Documento**:
* **Detalles**: ID, ID de viajero, tipo (por ejemplo, pasaporte), número, estado.
* **Acción**: Verificar estado.
  + **Vehículo (Vehículo)**:
* **Detalles**: ID, credencial de viajero, matrícula, modelo.
* **Acción**: Obtener detalles.
  + **Mascota**:
* **Detalles**: ID, identificación de viajero, certificado SAG, nombre.
* **Acción**: Obtener certificado.
  + **Turno (Turno)**:
* **Detalles**: ID, ID del viajero, fecha/hora, estado.
* **Acción**: Confirmar turno.
  + **ServicioValidación**:
* **Detalles** : ID del servicio.
* **Acciones** : Validar documentos, conectarse a SAG, PDI, Argentina.
  + **Líneas (Conexiones)** :
* Un Viajero puede tener muchos Documentos, Vehículo, Mascota, Turno (por ejemplo, un viajero, múltiples documentos).
* El documento está revisado por ServicioValidación.
* **¿Cómo ayuda?**
  + **Tracks Travelers** : Viajero almacena información para los registros.
  + **Maneja Documentos** : Documento admite el 90% de verificaciones automatizadas.
  + **Libros Turnos** : Turno programa cruces con ubicación telefónica.
  + **Administra mascotas/vehículos** : Mascota y Vehículo manejan información aduanera adicional.
  + **Conecta Agencias** : ServicioValidacion vincula a SAG, PDI, Argentina.
  + **Mantenlo seguro** : Los datos organizados cumplen con la norma ISO/IEC 27001.
  + **Acelera** : la estructura clara ayuda a procesar 1200 vehículos rápidamente.

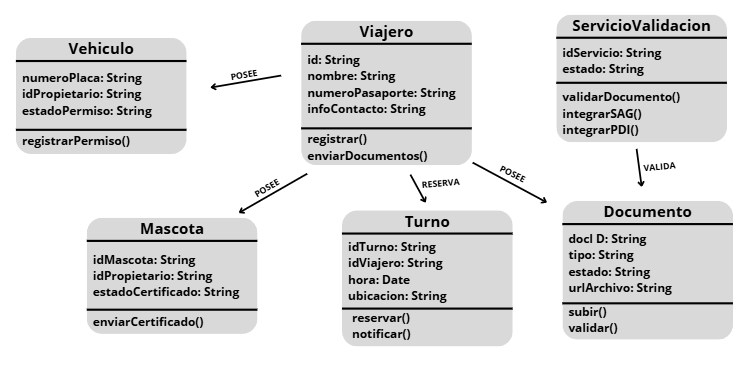
****

Figura 1. “Diagrama de Clases (Vista Lógica)”.

### 2.- Diagrama ER

El diagrama ER es como un mapa de la base de datos del sistema, que muestra cómo se almacenan y conectan datos como viajeros y documentos. Forma parte de la Vista Lógica (cómo se organizan los datos) en el modelo 4+1. Así es:

* **¿Qué es?**

Muestra tablas (similares a hojas de Excel) para datos, con sus columnas (campos) y cómo se vinculan. Cada tabla contiene un tipo de información, como detalles de viajeros u horarios de citas.

* **¿Qué contiene?**
  + Viajero (Viajero) : Almacena información del viajero.
* Columnas: id\_viajero (ID único), ruta, nombre, correo electrónico.
  + Documento : contiene detalles del documento.
* Columnas: id\_documento (identificación única), id\_viajero (enlaces a Viajero), tipo (p. ej., pasaporte), número, estado (p. ej., válido).
  + Vehículo (Vehículo) : Guarda información del vehículo.
* Columnas: id\_vehiculo (ID único), id\_viajero, patente, modelo.
  + Mascota : realiza un seguimiento de los datos de la mascota.
* Columnas: id\_mascota (ID único), id\_viajero, certificado\_sag, nombre.
  + Turno (Turno) : almacena los detalles de la cita.
* Columnas: id\_turno (ID único), id\_viajero, fecha\_hora, estado (p. ej., reservado).
  + Líneas (Enlaces) :
* Viajero se conecta a Documento, Vehiculo, Mascota, Turno (un viajero puede tener muchos documentos, vehículos, mascotas o turnos).
* Los enlaces usan id\_viajero para unir tablas (como una clave).
* **¿Cómo ayuda?**
  + Guarda información del viajero : la tabla Viajero almacena nombres y rutas para los registros.
  + Verifica documentos : la tabla Documentos rastrea los pasaportes para una automatización del 90%.
  + Libros Turnos : La mesa Turno programa los tiempos de cruce usando la ubicación del teléfono.
  + Maneja mascotas/vehículos : Las tablas Mascota y Vehículo almacenan detalles adicionales para SAG o aduanas.
  + Mantiene los datos seguros : las tablas organizadas cumplen con la norma ISO/IEC 27001 en materia de seguridad.
  + Acelera las cosas : la estructura de datos clara ayuda a procesar 1200 vehículos rápidamente.

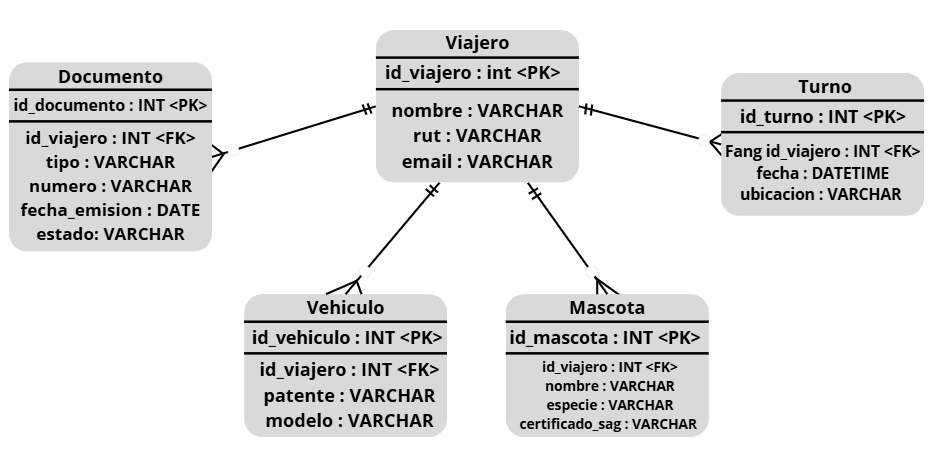


Figura 2 “Diagrama ER (Vista Lógica)”.

### 

## 8.2.- Vista de Despliegue

### 1.- Diagrama de paquete

El Diagramade Paquetes es como un organizador para nuestro sistema de aduanas. Forma parte de la Vista de Despliegue en el modelo 4+1.

* **¿Qué es?**

Organiza las partes del sistema en cinco "carpetas" (paquetes), cada una con una función, como gestionar viajeros o revisar documentos. Las flechas indican cómo se conectan.

* **¿Qué contiene?**
  + **Gestión de Usuarios** : Tiene la aplicación que los viajeros usan para registrarse y enviar cosas.
  + **Procesamiento de Documentos** : Verifica documentos (como pasaportes) automáticamente.
  + **Gestión de Turnos** : Libros de tiempos de cruce mediante ubicación GeoLocalización.
  + **Integración Externa** : Conversaciones con SAG, PDI y Argentina para datos.
  + **Persistencia** : Guarda todos los datos en una base de datos, como un archivador seguro.
  + **Flechas** : muestran conexiones, como usuarios que envían documentos o guardan datos.
* **¿Cómo ayuda?**
  + Automatiza el 90% de las comprobaciones de documentos.
  + Permite a los viajeros reservar horarios fácilmente.
  + Comparte datos con otras agencias.
  + Mantiene los datos seguros (sigue las reglas ISO/IEC 27001).
  + Acelera el procesamiento de 1.200 vehículos.

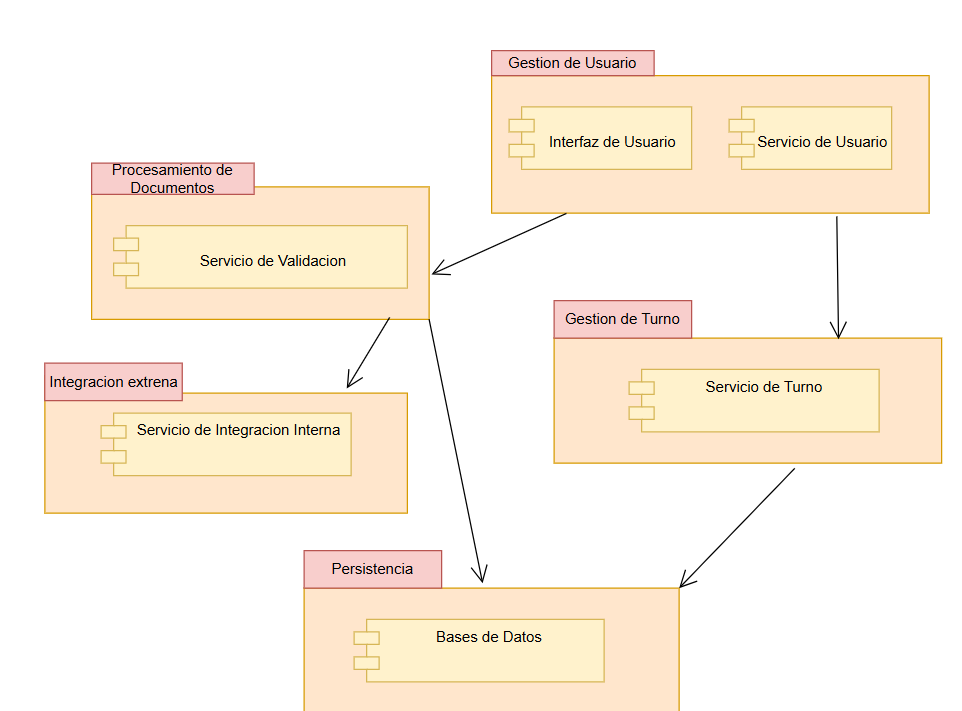


Figura 3. “Diagrama de paquete (Vista Despliegue)”.

### 2.- Diagrama de Componentes

El Diagrama de Componentes es como un plano de las piezas principales del sistema aduanero, mostrando cómo encajan. Forma parte de la Vista de Despliegue en el modelo 4+1.

* **¿Qué es?**

Muestra las partes principales del sistema (como la aplicación y la base de datos) como cuadros y cómo se conectan con líneas. Cada parte tiene una función, como revisar documentos o guardar datos.

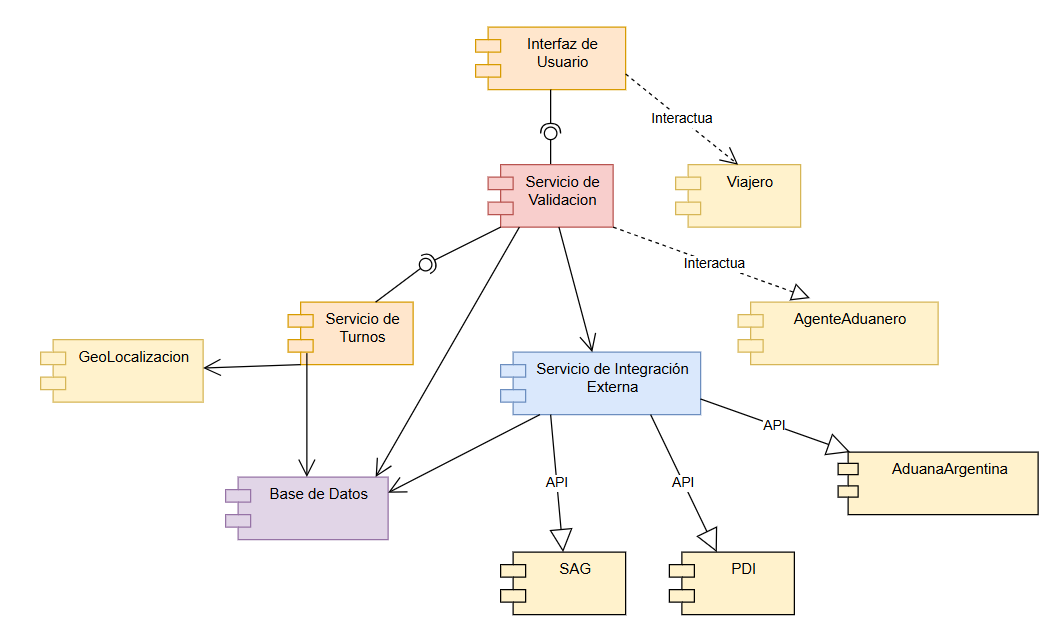
* **¿Qué contiene?**
  + **Interfaz de usuario** : la aplicación que utilizan los viajeros para registrarse, enviar documentos o reservar horarios.
  + **Servicio de Validación** : Verifica documentos (como pasaportes) automáticamente.
  + **Servicio de Turnos** : Establece tiempos de cruce utilizando la ubicación del teléfono.
  + **Servicio de Integración Externa** : Se conecta a SAG, PDI y Argentina para compartir información.
  + **Base de datos** : almacena todos los datos, como información de viajeros y documentos.
* **¿Cómo ayuda?**
  + **Verificaciones Rápidas** : El Servicio de Validación automatiza el 90% de las revisiones de documentos.
  + **Reserva fácil** : el Servicio de Turnos permite a los viajeros elegir horarios, sin largas esperas.
  + **Conecta a Todos** : Servicio de Integración Externa comparte datos con otras agencias.
  + **Datos seguros** : Base de Datos mantiene la información segura (cumple con la norma ISO/IEC 27001).
  + **Procesamiento rápido** : ayuda a gestionar 1200 vehículos en menos de una hora.

Figura 4. “Diagrama de Componentes (Vista Despliegue)”.

## 8.3-. Vista de Procesos

### 1.- Diagrama de actividad

En este diagrama, se representa visualmente cómo se realizan los procesos en la empresa, mostrando las actividades y las decisiones involucradas en cada paso. Lo que al llevarlo al contexto del caso de aduana “Los libertadores”, nos permite apreciar de forma detallada los procesos colaborativos de ésta dentro del sistema.

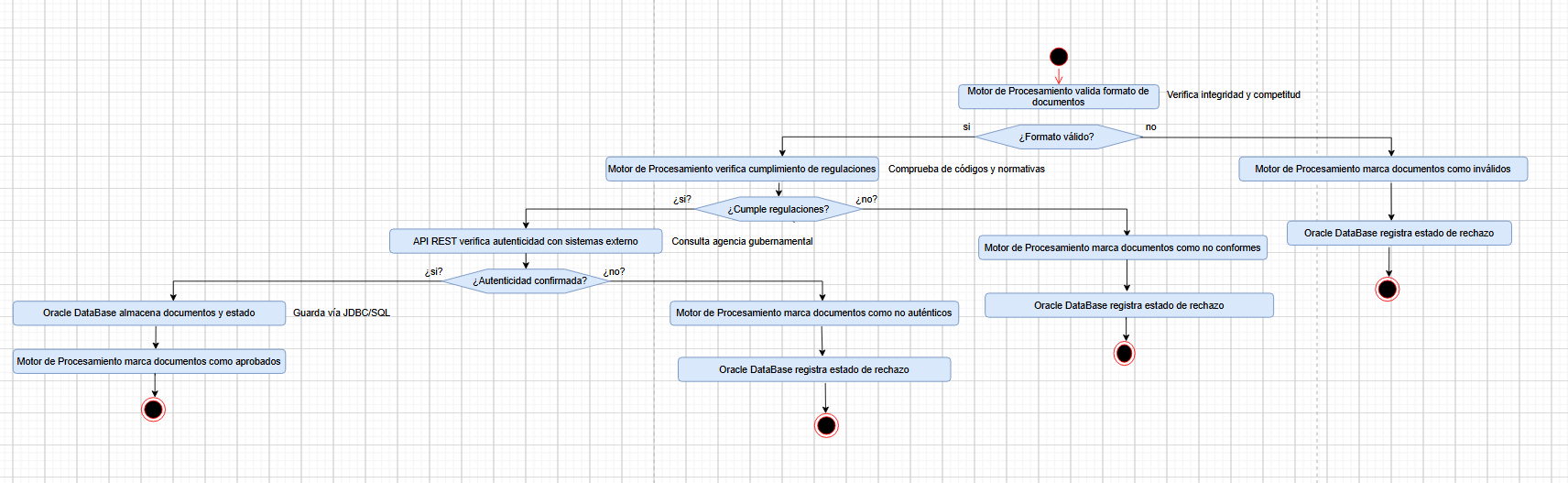


Figura 5. “Diagrama de actividad (Vista de procesos)”.

## 

### 2.- Diagrama de procesos (Vista de procesos).

En síntesis, el diagrama de procesos se basa en una representación gráfica que visualiza los pasos y flujos de procesos del sistema. Lo cual, llevado al contexto del caso aduana “Los libertadores”, en caso de validarse el formato, se verifican las regulaciones, si estas se cumplen, se consulta autenticidad para almacenar estos documentos en la base de datos y aprobarlo. Cabe destacar que si alguna validación no se cumple se rechaza automáticamente el proceso de documentos.

En este diagrama se puede visualizar la automatización detallada de los procesos de operaciones del sistema.

## 

Figura 6. “Diagrama de procesos (Vista de procesos)”.

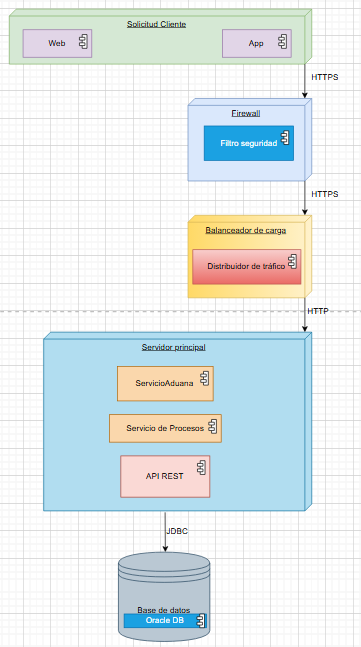
## 8.4.- Vista física.

### 1.-Diagrama de despliegue

En este diagrama se aprecia de manera visual como los componentes del software se asignan a los nodos de hardware en un sistema. Se detalla la infraestructura física y la configuración de red necesaria para que opere de manera eficaz.

Al llevar esta definición al caso de Aduana Los libertadores, se puede apreciar el cómo a través de la solicitud del cliente, ya sea mediante la web o la aplicación, se concreta el envío de información mediante puertos de conexión https y http.

Posteriormente, se verifica la información con un filtro de seguridad Firewall, llegando al balanceador de carga para desencriptar los datos y así poder distribuirlo en el servidor principal del software y concretar la petición, guardando exitosamente en la base de datos.

Figura 7. “Diagrama de despliegue (Vista física)”.

## 8.5.- Vista de Escenarios.

### 1.- Diagrama de Casos de Uso

* **Propósito y Rol:**

Muestra las interacciones entre actores (Viajero, Agente Aduanero, SAG, PDI, Aduana Argentina) y el sistema, definiendo escenarios clave como enviar documentos y solicitar turnos. Su rol es guiar la automatización para reducir demoras en el Paso Los Libertadores.

* **Componentes:**

**Actores:** Viajero, Agente Aduanero, Oficial SAG, Oficial PDI, Aduana Argentina.

**Casos de Uso:** Registrar Perfil, Enviar Documentos, Validar Documentos, Solicitar Turno Virtual, Monitorear Operaciones, Intercambiar Datos.

* **Relaciones:**

**Asociaciones**: Actores con casos de uso.

**<<include>></include>:** Enviar Documentos incluye Registrar Perfil.

**<<extend>></extend>:** Validar Documentos se extiende a SAG/PDI.

* **Alineación con la Propuesta:**

**Demoras**: Turno Virtual y Validar Documentos reducen tiempos (de 8-20 a <1 hora).

**Automatización:** Gestiona el 90% de validaciones.

**Integración:** Intercambiar Datos y extensiones a SAG/PDI eliminan redundancias.

**Monitoreo:** Apoya el dashboard en tiempo real.

El diagrama alinea los casos de uso con los objetivos de eficiencia y cumplimiento de la propuesta.

### Diagrama El contenido generado por IA puede ser incorrecto.

### La aplicación del modelo 4+1 permite cubrir de forma integral todos los aspectos técnicos, funcionales y de operación del sistema, asegurando una solución robusta, escalable, segura y centrada en el usuario. Cada vista complementa a las otras, logrando una arquitectura coherente y bien fundamentada para enfrentar el desafío del control fronterizo moderno.

# 9.- Estándares de Calidad en el Diseño de Software.

La Propuesta de Automatización Aduanera aplica estándares de calidad basados ​​en principios de ingeniería de software para garantizar un sistema confiable, eficiente y seguro. Estos estándares (modularidad, escalabilidad, seguridad y mantenibilidad) contribuyen a un software de alta calidad que cumple con las necesidades técnicas y regulatorias.

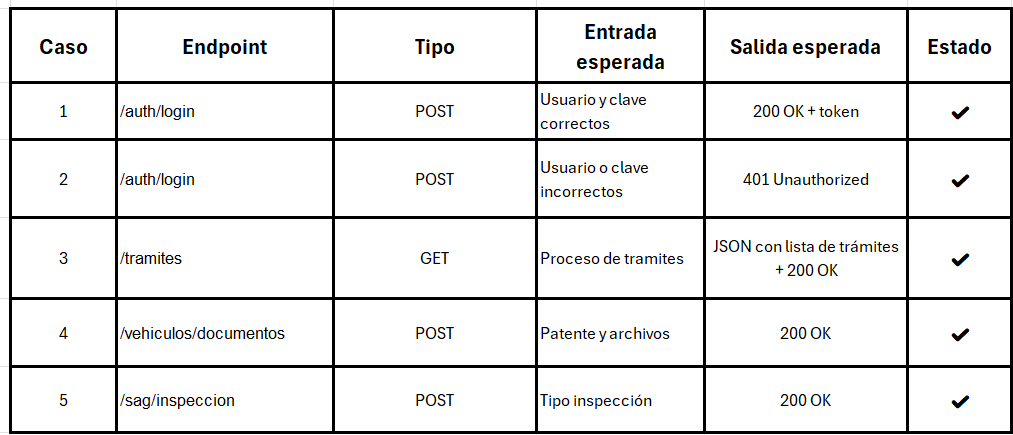
* **Modularidad:**
  + De qué se trata: dividir el sistema en partes pequeñas e independientes (como bloques Lego) que gestionan tareas específicas (por ejemplo, gestión de usuarios, validación de documentos).
  + Cómo se aplica:
* Diagrama de componentes (Vista de desarrollo): Separa partes como Interfaz de Usuario y Servicio de Validación.
* Diagrama de paquetes (Vista de desarrollo): Agrupa el código en paquetes (p. ej., Gestión de Usuarios, Procesamiento de Documentos).
  + Por qué es útil: Facilita la creación, prueba y reparación del sistema. Por ejemplo, actualizar la validación de documentos no interrumpe la programación de turnos.
  + Impacto del proyecto: Admite el 90% de la automatización de documentos y la integración con SAG/PDI al organizar el código de forma clara.
  + Principio de ingeniería: la separación de preocupaciones garantiza que cada módulo tenga un trabajo, lo que mejora la claridad y la reutilización.
* **Escalabilidad:**
  + De qué se trata:

Diseñar el sistema para que pueda manejar más usuarios o datos sin colapsar (como una carretera que puede soportar más automóviles).

* + Cómo se aplica:
* Diagrama de implementación (vista física): utiliza un balanceador de carga para distribuir el tráfico entre servidores web y una configuración en la nube para una disponibilidad del 99,5 %.
* Diagrama de componentes : Los servicios modulares (p. ej., Servicio de Turnos) pueden escalar de forma independiente.
  + Por qué ayuda: Garantiza que el sistema procese 1200 vehículos en menos de 1 hora, incluso durante las horas pico.
  + Impacto del proyecto: satisface requisitos de alta demanda y apoya el crecimiento (por ejemplo, más viajeros).
  + Principio de ingeniería: La arquitectura escalable planifica la carga futura, garantizando el rendimiento.
* **Seguridad:**
  + Qué es: Proteger los datos y el acceso para mantener el sistema seguro (como las cerraduras de una casa).
  + Cómo se aplica:
* Diagrama ER (vista lógica): define estructuras de datos seguras (por ejemplo, identificaciones de viajeros cifradas).
* Diagrama de implementación: utiliza HTTPS para conexiones seguras y almacenamiento de base de datos compatible con ISO/IEC 27001.
* Diagrama de clases (vista lógica): incluye métodos seguros (por ejemplo, validarDocumento).
  + Por qué ayuda: Protege datos confidenciales (rutas, documentos) y garantiza la confianza de los viajeros y las agencias.
  + Impacto del Proyecto: Cumple con la norma ISO/IEC 27001 y la normativa aduanera chilena para protección de datos.
  + Principio de ingeniería: La seguridad por diseño incorpora protección en cada capa, reduciendo los riesgos.
* **Mantenibilidad:**
  + De qué se trata: Hacer que el sistema sea fácil de actualizar o reparar (como un automóvil con piezas fáciles de reemplazar).
  + Cómo se aplica:
* Diagrama de paquetes: organiza el código en paquetes claros para facilitar las actualizaciones.
* Diagrama de casos de uso específicos (vista de escenarios): prueba escenarios (por ejemplo, envío de documentos) para garantizar que las tareas funcionen, lo que simplifica la depuración.
* La fase de mantenimiento del SDLC supervisa y actualiza las nuevas reglas aduaneras.
  + Por qué ayuda: Permite soluciones rápidas (por ejemplo, nuevas reglas SAG) sin rehacer todo el sistema.
  + Impacto del proyecto: Garantiza la confiabilidad a largo plazo y el cumplimiento de las regulaciones cambiantes.
  + Principio de ingeniería: El diseño mantenible reduce la deuda técnica,

manteniendo el sistema adaptable.

# 10. Diseño de Casos de Prueba



# 11. Ejecución de Pruebas

Cada uno de los servicios fue probado en entorno real a través de Postman, verificando:

* Estado HTTP de la respuesta
* Tiempo de respuesta
* Contenido estructurado en formato JSON
* Flujo de trabajo completo entre módulos

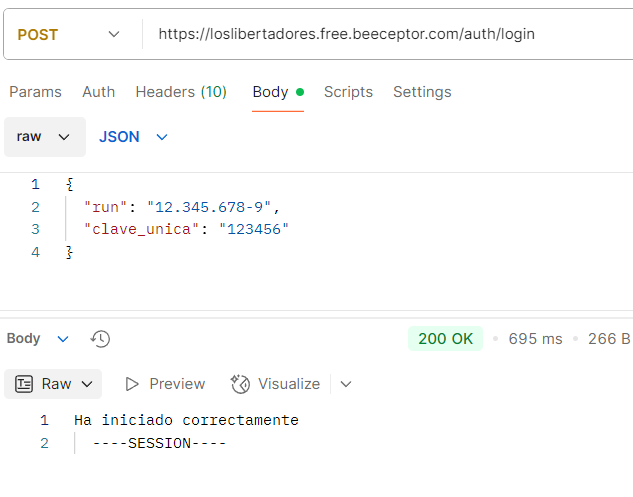
Se realizaron pruebas de ingreso exitoso, validación de errores, verificación de registros enviados y confirmación de respuestas. Las pruebas fueron exitosas, confirmando que el sistema opera correctamente.

# 12. Casos de prueba

## Inicio Sesión

****

* **Primera prueba en postman**

****

* **Prueba Unitaria**

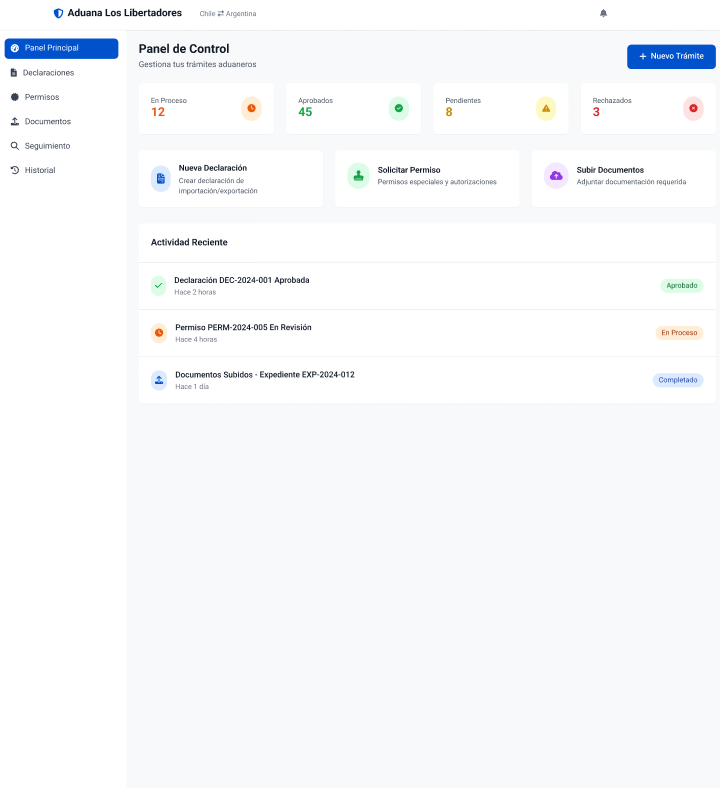
**Prueba en Postman:** Ingreso de credenciales correctas.  
**Esperado:** status 200 y token o mensaje "Acceso concedido".

Esta prueba usa JUnit para validar que el método login() devuelve true cuando las credenciales son correctas. Se simula el ingreso del usuario y se comprueba la autenticación.

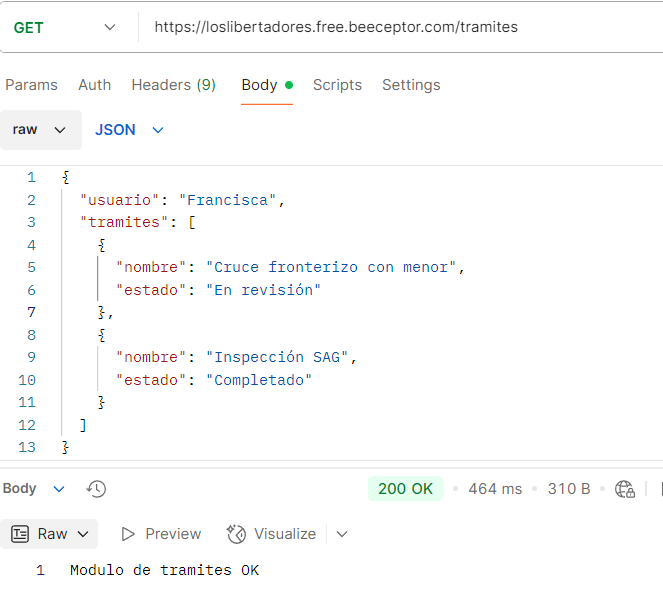
**Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.**

## 12.2 Modulo Tramites en Línea (Principal)

****

* **Pruebas en Postman:**

****

* **Pruebas Unitarias**

**Escenario en Postman:** El usuario puede ver o registrar el estado de su trámite.  
**Objetivo:** Confirmar que el sistema guarda o retorna el estado correcto.

1. Registro de un trámite

Se simula que el usuario genera un nuevo trámite con su RUT y tipo de gestión (ej: “Cruce Fronterizo”). La prueba unitaria valida que el método registrar () del servicio de trámites retorna true, indicando que el trámite fue guardado correctamente.

Esta lógica puede implementarse con JUnit y utilizar **Mockito** para simular la respuesta del repositorio que guarda los datos (por ejemplo, una base de datos o un servicio REST).

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

1. Consulta del estado de trámite

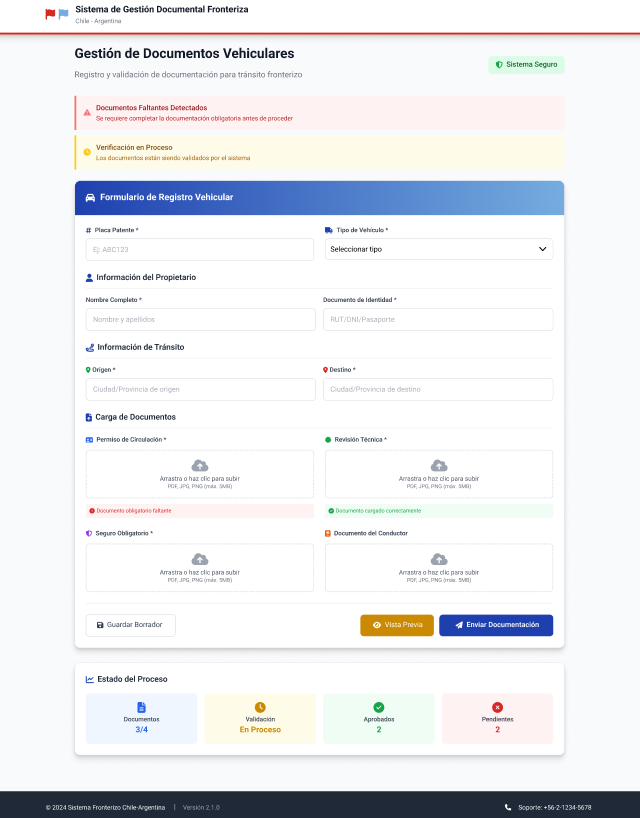
El sistema debería permitir consultar el estado actual de un trámite ingresando el RUT del usuario. La prueba unitaria usa **Mockito** para simular el resultado de la búsqueda (por ejemplo, “En Proceso”) y luego valida con assertEquals() que ese estado sea el esperado.

Esto permite testear la lógica de consulta sin necesidad de tener un repositorio real conectado.

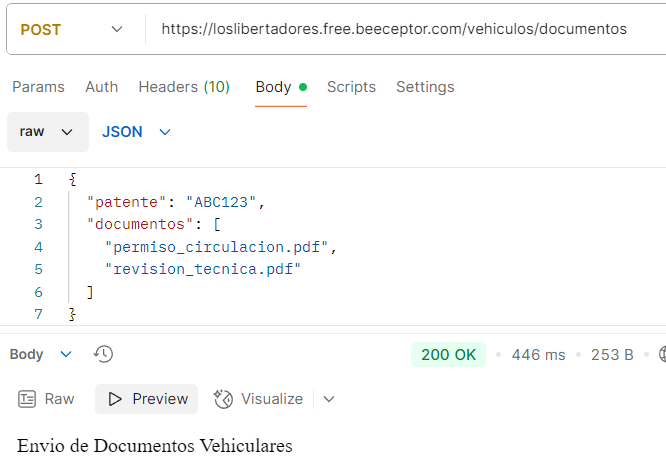
**Texto, Carta

El contenido generado por IA puede ser incorrecto.**

## 12.3 Modulo Documentos vehiculares

****

* **Prueba en Postman:**

****

* **Pruebas Unitarias**

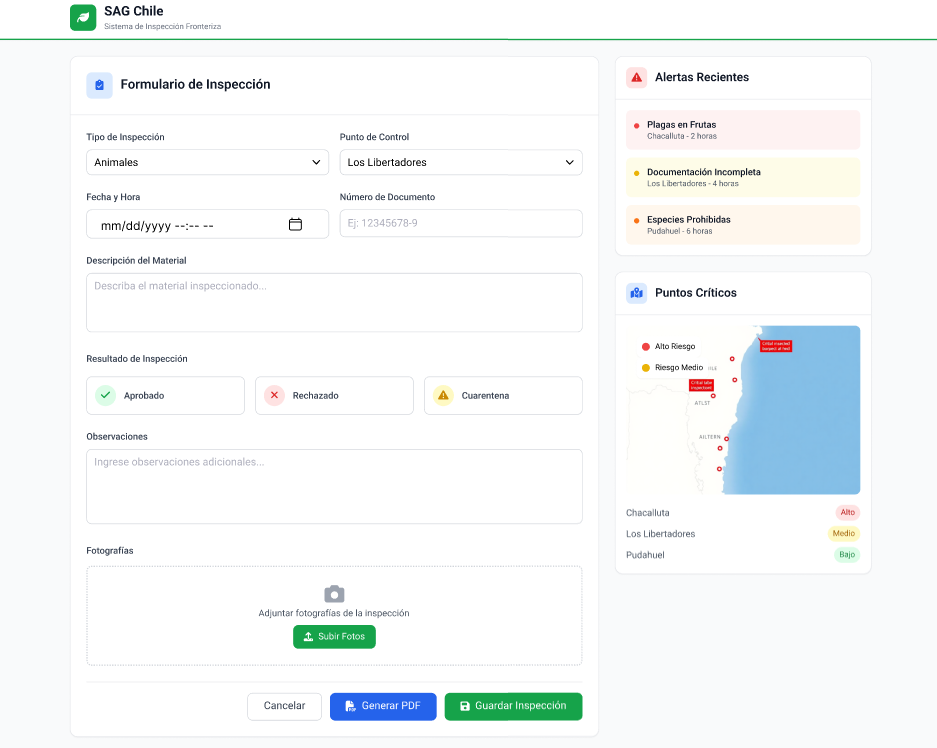
**Prueba en Postman:** Se envía archivo/documento correcto.  
**Esperado:** status 200, mensaje “Documento recibido”.

Se simula la carga de un documento PDF usando un método subir () del servicio de documentos. Si el archivo cumple con el formato, se espera que retorne true. Esta prueba también se puede simular con Mockito si el servicio depende de una base de datos.

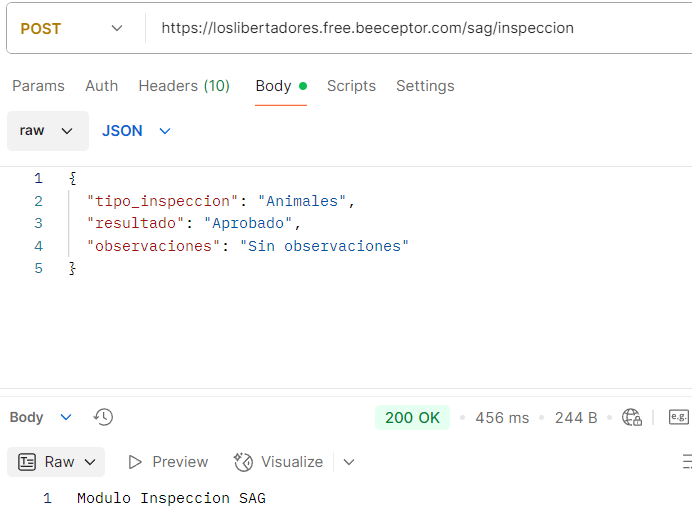
**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

## 12.4 Modulo SAG

****

* **Prueba en postman:**

****

* **Pruebas Unitarias**

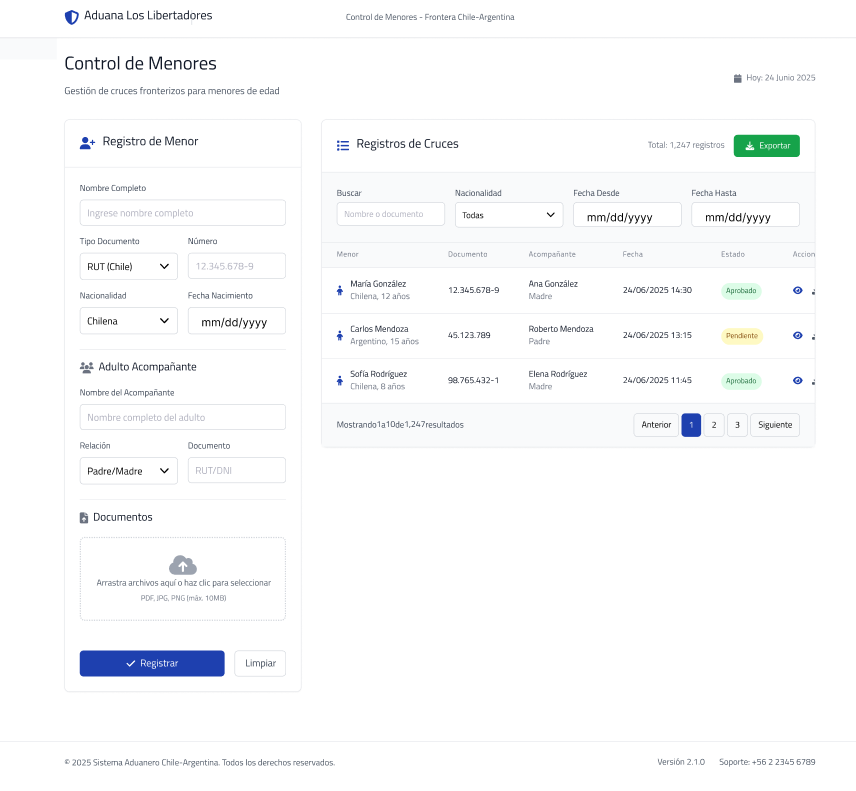
**Postman simula integración con SAG y retorno exitoso**.  
**Esperado:** validación true.

En este caso se usa Mockito para simular que el servicio externo del SAG siempre responde "válido". Esto permite testear la lógica del sistema aunque el API real aún no esté disponible.

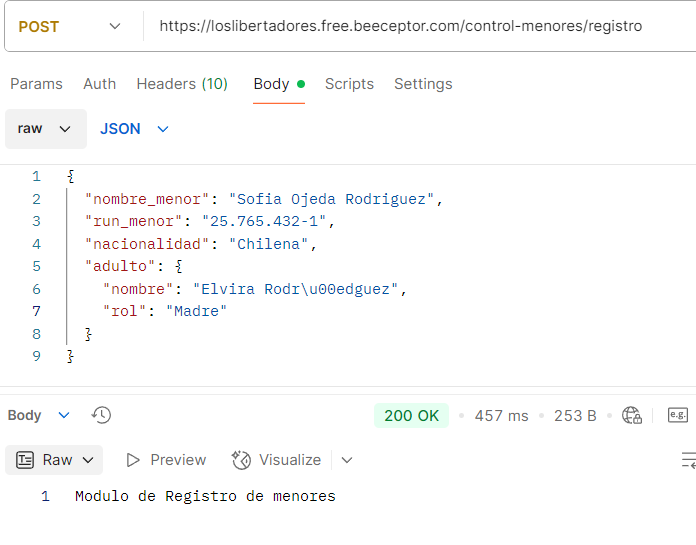
**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.**

## 12.5 Modulo Control de menores

****

* **Prueba en Postman**



* **Pruebas Unitarias**

**Escenario en Postman:** El usuario registra un menor para el control migratorio.  
**Objetivo:** Confirmar que el sistema verifica edad y guarda correctamente al menor

1. Menor con autorización válida

Se evalúa que el sistema permita registrar correctamente a un menor de edad acompañado, siempre que se adjunte una autorización notarial válida. La prueba simula un objeto Menor con nombre, fecha de nacimiento y archivo de autorización, y espera que el método registrar() retorne true.

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

1. Menor sin autorización

Se simula un escenario en el que el usuario intenta registrar a un menor sin adjuntar autorización. En este caso, la prueba espera que el método lance una excepción de validación (**ValidacionException**) con un mensaje como “Se requiere autorización notarial para menores”.

Este tipo de validaciones ayuda a evitar errores humanos y garantiza el cumplimiento de la normativa migratoria.

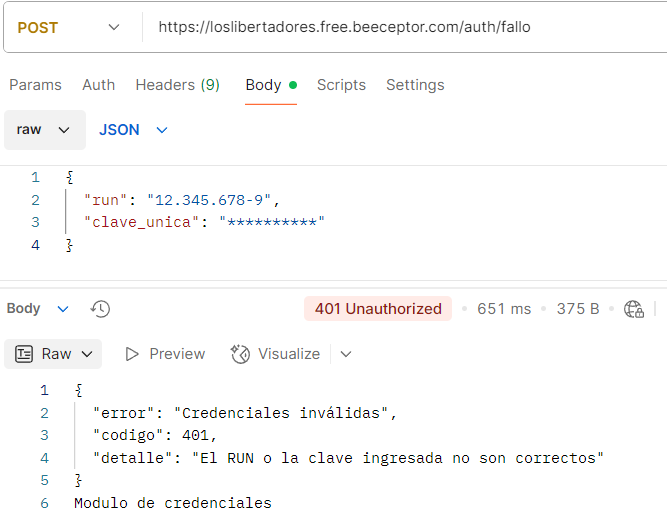
Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## 12.6 Modulo Protocolo de seguridad en caso de Credenciales Inválidas



* **Prueba en Postman:**



* **Pruebas Unitarias**

**Prueba en Postman:** Usuario con clave incorrecta.  
**Esperado:** status 401, mensaje de error.

Aquí se valida que si las credenciales son incorrectas, el sistema lanza una excepción personalizada CredencialesInvalidasException. Esto asegura que el protocolo de seguridad reaccione correctamente. Una captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

# 

# 12.7.- Integración de Servicios

Se comprobó que los módulos funcionan de forma integrada y coherente. Por ejemplo:

* El usuario debe iniciar sesión antes de registrar un vehículo o menor.
* El sistema permite almacenar documentos y asociarlos al historial del usuario.
* Las inspecciones del SAG se vinculan con el registro vehicular.

Cada respuesta fue validada en tiempo real, permitiendo asegurar el correcto funcionamiento de la plataforma en condiciones reales.

# 13.- Cronograma del proyecto



# 

# 

# 14.- Análisis de Riesgos

Durante la integración de los módulos funcionales del sistema aduanero y las pruebas de conectividad entre servicios, se identificaron diversos riesgos que podrían presentarse en un ambiente productivo. A continuación, se detalla su evaluación desde una perspectiva profesional, junto con las medidas preventivas

|  |  |  |
| --- | --- | --- |
| **Riesgo** | **Impacto** | **Acción preventiva o correctiva** |
| Caída del servicio de autenticación | Alto | Se estableció un mecanismo de respuesta controlada ante fallas de login, retornando códigos de error HTTP 401 con mensajes personalizados y auditables. Esto permite a los operadores identificar rápidamente el fallo e iniciar protocolos de recuperación. |
| Fallo en la transmisión de documentos vehiculares | Medio | Se definió un esquema de validación de campos y extensión de archivos antes de su envío, asegurando que los documentos cumplan con los requisitos del sistema. Además, se incorporó un control de respuesta en el front para verificar acuse de recibo. |
| Inconsistencias en datos ingresados por el usuario | Medio | Se implementaron validaciones en el lado del cliente y en el servidor para evitar envíos de información incompleta o incorrecta, especialmente en módulos críticos como el control de menores o inspección SAG. |
| No disponibilidad de los endpoints de trámite en horarios pico | Alto | Se previó un plan de contingencia para mantener la experiencia del usuario mediante respuestas cacheadas o mensajes temporales que informen la situación sin interrumpir el flujo general del sistema. |
| Errores no visibles en registros GET | Bajo | Dado que las operaciones GET no son persistidas en algunos logs, se reforzó la recolección de evidencia mediante herramientas externas (como Postman o monitorización), asegurando la trazabilidad de cada petición crítica. |
| Pérdida de respaldo de configuración de endpoints | Medio | Toda la documentación técnica, estructuras de prueba y configuraciones de endpoints fueron respaldadas en repositorios locales y en nube institucional, garantizando recuperación inmediata en caso de pérdida o fallo de entorno. |

# 

# 15.- Control de Versiones con GitHub

Durante el desarrollo del proyecto se utilizó **Git como sistema de control de versiones distribuido** y **GitHub como plataforma de gestión del repositorio remoto**, lo que permitió mantener un flujo de trabajo organizado, trazable y colaborativo.

## 15.1 Repositorio GitHub

El repositorio se alojó en la plataforma GitHub (<https://github.com>), permitiendo realizar:

* Seguimiento de cambios en el código fuente.
* Revisión de commits históricos.
* Control de versiones por medio de ramas.
* Integración futura con herramientas de CI/CD si se requiere.
* Revisión colaborativa mediante *pull requests*.

**Ejemplo de repositorio:**  
https://github.com/Macacruzf/proyecto-aduanero.git (simulado)

**14.2 Flujo de trabajo con ramas**

Se utilizó una estructura de ramas inspirada en **Git Flow** para organizar el trabajo:

* main: rama principal con la versión más estable del sistema.
* dev: rama de integración donde se van uniendo las nuevas funcionalidades.
* feature/\*: ramas para cada nueva característica, como:
  + feature/login
  + feature/tramites
  + feature/sag-integration
  + feature/registro-menores

Cada funcionalidad se desarrolla en su propia rama y luego se realiza un *pull request* hacia dev. Una vez validadas todas las funcionalidades, se integran a main.

**14.3 Convención de commits en GitHub**

Se siguieron buenas prácticas al momento de registrar cambios (commits):

* feat: para nuevas funcionalidades.  
  Ej: feat: agregar validación de documentos SAG
* fix: para correcciones de errores.  
  Ej: fix: corregir lógica de validación de RUT
* refactor: para mejoras de código sin cambiar funcionalidad.  
  Ej: refactor: simplificar controlador de trámites

Esto mejora la legibilidad del historial de cambios y facilita el trabajo en equipo.

**14.4 Visualización y gestión**

Desde GitHub se pueden realizar tareas clave como:

* Visualización de ramas y commits.
* Comentarios en líneas de código específicas.
* Fusión controlada de cambios mediante *pull requests*.
* Visualización gráfica del flujo de trabajo desde la pestaña **"Insights > Network"**.

# 16. Conclusión

El desarrollo del sistema de automatización aduanera para el Paso Los Libertadores representa una propuesta integral y tecnológicamente avanzada para enfrentar los principales desafíos del control fronterizo. A lo largo de este proyecto, se construyó un marco sólido que equilibra la eficiencia operativa, el cumplimiento normativo y una experiencia de usuario optimizada. Este enfoque permitió alcanzar importantes logros, proyectando un impacto significativo tanto a nivel local como nacional.

## Principales logros

### Optimización de procesos:

* + Reducción estimada del 70 % en los tiempos de espera, disminuyendo de entre 8 y 20 horas a menos de 1 hora por trámite.
  + Automatización del 90 % de las validaciones documentales, minimizando errores humanos y aumentando la precisión operativa.

### Integración tecnológica:

* + Comunicación en tiempo real con sistemas del SAG, PDI y Aduana Argentina, eliminando redundancias y acelerando los procesos de revisión.
  + Implementación de mecanismos de seguridad alineados con estándares internacionales (ISO/IEC 27001) y la Ley N° 19.628, garantizando la confidencialidad, integridad y disponibilidad de los datos.

### Enfoque centrado en el usuario:

* + Interfaz intuitiva, accesible desde múltiples dispositivos y diseñada bajo criterios de accesibilidad universal.
  + Incorporación de turnos virtuales y notificaciones en tiempo real, mejorando la experiencia y reduciendo incertidumbre.

**Impacto esperado**

* **Para los usuarios:** Un cruce fronterizo más ágil, transparente y menos estresante.
* **Para las instituciones:** Mayor control operativo, reducción de costos logísticos y aumento en la trazabilidad documental.
* **Para el país:** Impulso al comercio binacional y al turismo mediante una frontera más eficiente y moderna.

**Próximos pasos**

1. Validación del prototipo con usuarios reales en entornos controlados.
2. Implementación piloto en el Paso Los Libertadores, con monitoreo activo de métricas y APIs.
3. Escalabilidad del modelo a otros pasos fronterizos del país, replicando la experiencia y mejorando sobre la base de los aprendizajes.

**Compromiso con la calidad**

Durante todo el desarrollo, se incorporaron mecanismos rigurosos de aseguramiento de calidad:

* Pruebas unitarias, de integración y de carga en distintas fases del sistema.
* Auditorías periódicas de seguridad, funcionalidad y desempeño.
* Documentación completa siguiendo el estándar IEEE 830 para garantizar la trazabilidad de requerimientos y decisiones técnicas.

Gracias al uso de herramientas profesionales como Postman, se logró validar con precisión cada funcionalidad crítica del sistema, permitiendo simular un comportamiento realista y confiable ante distintos escenarios.

En definitiva, este proyecto no solo aborda los problemas actuales del control aduanero, sino que **sienta las bases para una transformación digital profunda del sistema fronterizo chileno**, abriendo camino hacia una gestión más segura, eficiente y centrada en las personas.

# 