**Sistema de Entrada de Vehículos para Servicio Nacional de Aduanas**

**(DAS) Documento Arquitectura de Software**

**Versión 1.0.0**

Integrantes:

David A. Ramírez Luna

Rodrigo Aedo

Asignatura: Ingeniería de Software Sección 001D

Profesora: Mabel Herrera Pino

**Identificación de Documento**

| **Identificación** | **DAS 1.0.0** |
| --- | --- |
| **Proyecto** | **Sistema de Entrada de Vehículos para Servicio Nacional de Aduanas** |
| **Versión** | **V1.0.0** |

| **Documento mantenido por** | **Rodrigo Aedo y David Ramírez** |
| --- | --- |
| **Fecha de última revisión** | **26 / 05 / 2025** |
| **Fecha de próxima revisión** | **6 / 01 / 2025** |

| **Documento aprobado por** |  |
| --- | --- |
| **Fecha de última aprobación** |  |

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| **26 / 05 / 2025** | **0.0.0** | **Versión inicial del DAS: se analizó la arquitectura base del sistema, incluyendo modelo 4+1 y componentes principales** | **David Ramirez,**  **Rodrigo Aedo** |
| **7 / 01 / 2025** | **1.0.0** | **Segunda fase: se incorporaron mejoras en usabilidad, prototipos mockup, evaluación heurística con principios de Nielsen y métricas de calidad.** | **David Ramirez,**  **Rodrigo Aedo** |
|  |  |  |  |
|  |  |  |  |

**Tabla de Contenidos**

[**1. INTRODUCCIÓN 3**](#_heading=h.xghkzn4qboxj)

[1.1 Contexto del Problema 3](#_heading=h.xtuzwnhu6sj5)

[1.2 Propósito 3](#_heading=h.cn7v02bn33l3)

[1.3 Ámbito 3](#_heading=h.5b1kzyejaqxd)

[1.4 Definiciones, acrónimos y abreviaciones 3](#_heading=h.r0ci0k6ttpps)

[1.5 Resumen ejecutivo 3](#_heading=h.qzjql8mwgdn0)

[1.6 Arquitectura del Sistema 4](#_heading=h.vlvevdd4b0f)

[**2 VISIÓN DEL SISTEMA 4**](#_heading=h.9g7y0e4zmnze)

[2.1 Descripción general del sistema 4](#_heading=h.zccghuth2yp5)

[2.2 Objetivos del sistema 4](#_heading=h.r0yqxaqg9bv6)

[2.3 Principales funcionalidades esperadas 4](#_heading=h.je963wlobjbg)

[2.4 Supuestos y dependencias 4](#_heading=h.75ejwf9zxom3)

[**3 ESTILOS Y PATRONES ARQUITECTÓNICOS 4**](#_heading=h.3k147mtgg6ri)

[3.1 Estilo arquitectónico adoptado 4](#_heading=h.do86waowd0nz)

[3.2 Justificación del estilo según el contexto del sistema 4](#_heading=h.1hnw855iqv20)

[3.3 Patrones de diseño aplicados 4](#_heading=h.5e9y03j9v6c)

[**4 MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS 4**](#_heading=h.iqcqhh6w2i9y)

[4.1 VISTA DE ESCENARIO 4](#_heading=h.migdsfjglyve)

[***4.1.1 Propósito 4***](#_heading=h.8gm1fq8qt5j)

[***4.1.2 Actores 5***](#_heading=h.uhv1w0ravazg)

[***4.1.3 Diagrama general de casos de uso 6***](#_heading=h.hgjty4ugcps9)

[***4.1.4 Diagrama de casos de uso específicos 7***](#_heading=h.n7zwf3r3cpeb)

[***4.1.5 Lista de casos de uso (salida vehículo o entrada vehículo) 7***](#_heading=h.trts00fqhwmx)

[***4.1.6 Especificación de casos de uso 8***](#_heading=h.kufbdadfpuxe)

[4.2 VISTA LÓGICA 9](#_heading=h.iu6fksmmufhb)

[***4.2.1 Propósito 9***](#_heading=h.mealewbd6cuj)

[***4.2.2 Diagrama de clases 10***](#_heading=h.52qr1oss85o2)

[***4.2.3 Descripción diagrama de clases 10***](#_heading=h.ccq3nzcsdslr)

[4.3 VISTA DE IMPLEMENTACIÓN/DESARROLLO 11](#_heading=h.6uwz3e3xzgex)

[***4.3.1 Propósito 11***](#_heading=h.n7borud1oqrk)

[***4.3.2 Diagrama de componente 11***](#_heading=h.t1bw4emhspo6)

[***4.3.3 Descripción diagrama de componente 11***](#_heading=h.mbsyoqr4b40v)

[***4.3.4 Diagrama de paquete 12***](#_heading=h.fkn0kyl3alr5)

[***4.3.5 Descripción diagrama de paquete 12***](#_heading=h.fokkc8jvc9wa)

[4.4 VISTA DE PROCESOS 12](#_heading=h.v89zgq34c9mq)

[***4.4.1 Propósito 12***](#_heading=h.5f3zia2k2adj)

[***4.4.2 Diagrama de actividad 13***](#_heading=h.kew6jxkfhwe3)

[***4.4.3 Descripción diagrama de actividad 13***](#_heading=h.2uarkcgbufxg)

[4.5 VISTA FÍSICA 13](#_heading=h.csk1h2ebkm2y)

[***4.5.1 Propósito 13***](#_heading=h.fiickj9ytfjf)

[***4.5.2 Diagrama de despliegue 14***](#_heading=h.t8re6l1m3f51)

[***4.5.3 Descripción diagrama de despliegue 14***](#_heading=h.5igm5vwvt96d)

[**5 REQUISITOS DE CALIDAD 14**](#_heading=h.cl02povumqgl)

[5.1 Propósito 14](#_heading=h.609lnafl4bhk)

[5.2 Atributos de calidad 15](#_heading=h.ts2ubq2wfj2a)

[5.3 Reglas y criterios de evaluación de Calidad 15](#_heading=h.xcqyebu4kxv)

[**6 PRINCIPIOS DE DISEÑO APLICADOS 15**](#_heading=h.eeaovlpm4r8x)

[6.1 Propósito 15](#_heading=h.nva9as6hwmdi)

[6.2 Principios de diseño 15](#_heading=h.r5im8wexmdpt)

[**7 PROTOTIPO 16**](#_heading=h.vhkkmoypou2f)

[7.1. Propósito 16](#_heading=h.yxfqrjc2001m)

[7.2. Mockups 16](#_heading=h.kguacqbyc145)

[7.2.1 Recorrido que debe seguir un turista para registrar una solicitud de ingreso vehicular 16](#_heading=h.acm0ednmi04h)

[7.2.2 Proceso que realiza un agente aduanero para revisar dichas solicitudes y proceder a validarlas, rechazarlas o marcarlas como pendientes 20](#_heading=h.mto4fn8sfaq7)

[7.3. Justificar herramientas de prototipado 24](#_heading=h.5r8ujgr10ibm)

[**8 EVALUACIÓN DE CALIDAD HEURÍSTICA DE NIELSEN 24**](#_heading=h.wnt687qcspsp)

[8.1. Propósito 24](#_heading=h.cmojjzqrri1s)

[8.2. Lista de verificación 25](#_heading=h.wmslt4fyq9mq)

[8.3. Análisis y métricas de resultados 25](#_heading=h.ixmkbu44pocb)

[**9 CONTROL DE VERSIONES 25**](#_heading=h.d1fapyaqf8yz)

[9.1. Propósito 25](#_heading=h.p8tr6r4vejjp)

[9.2. Control de versión utilizado 25](#_heading=h.tfv6yxrwdh22)

[9.3. Justificar herramientas de versionamiento 25](#_heading=h.heqhxzx6lc0w)

[**10 CONCLUSIONES 26**](#_heading=h.fpevbfb0vba9)

# 1. INTRODUCCIÓN

## Contexto del Problema

El Sistema Nacional De Aduanas es el ente que se encarga de facilitar las operaciones de comercio exterior, tanto de exportación como importación; además de fiscalizar dichas operaciones.

El SNA actualmente busca fortalecer su institucionalidad, mediante mecanismos de gestión, eficientes y eficaces. Además, desea mejorar la calidad y trazabilidad de las operaciones de despacho aduanero. En este contexto, el SNA busca implementar un sistema de software que abarque dichos objetivos.

## Propósito

El proyecto busca modernizar los procesos aduaneros y migratorios en el paso fronterizo de los libertadores, mediante tecnologías digitales.

La reducción de los tiempos de espera beneficiará tanto a usuarios como a transportistas.

Mejorando la experiencia del cruce y optimizando la logística comercial entre Chile y Argentina.

Con estas mejoras se logrará la transparencia en la documentación y seguridad en los procedimientos aduaneros, reduciendo los márgenes de error que actualmente presenta la zona aduanera, optimizando también los recursos humanos en los puntos de control.

## Ámbito

El alcance principal incluye la digitalización y procesamiento de formularios, la validación de la información de vehículos, conductores y viajes, la interoperabilidad con sistemas externos como PDI para la validación de datos, la gestión de estados de las solicitudes y el registro de motivos de cambio de estado, y finalmente la implementación de mecanismos para adjuntar y validar documentación.

## Definiciones, acrónimos y abreviaciones

| **ACRÓNIMO** | **DESCRIPCIÓN** |
| --- | --- |
| SNA | Servicio Nacional de Aduanas |
| MVC | Modelo Vista Controlador |

## Resumen ejecutivo

Este Documento de Arquitectura de Software (DAS) detalla la estructura fundamental y las decisiones de diseño para un nuevo sistema de modernización aduanera y migratoria en pasos fronterizos. El sistema busca optimizar el proceso de ingreso de vehículos, permitiendo completar formularios digitalmente y validar y gestionar eficientemente solicitudes. Se busca una solución que no solo reduzca los tiempos de espera y mejore la trazabilidad, sino que también garantice la seguridad y transparencia en las operaciones aduaneras.

## Arquitectura del Sistema

Este documento contempla la elaboración de diagramas provenientes del Modelo 4+1 y diagramas de Vistas arquitectónicas como diagramas de casos de uso para vista de escenario, diagrama de clases para vista lógica, diagrama de componente y paquetes para la vista de implementación, diagrama de actividad para la vista de procesos, y diagrama de despliegue para la vista física. Cada uno de estas vistas va dirigía a un tipo de usuario involucrado tanto en el uso del producto final, como en el desarrollo de este proyecto.

# VISIÓN DEL SISTEMA

## Descripción general del sistema

El sistema de software está diseñado para modernizar y agilizar el proceso de control fronterizo entre Chile y sus países limítrofes, especialmente con Argentina.

## Objetivos del sistema

Su objetivo principal es reducir los tiempos de espera en las aduanas mediante la digitalización de trámites y la interoperabilidad entre los organismos involucrados (Aduanas, SAG y PDI).

## Principales funcionalidades esperadas

El sistema permitirá gestionar múltiples aduanas y usuarios, incluyendo ciudadanos (chilenos y extranjeros) y personal aduanero con diferentes roles operativos. Se contempla el registro y validación de vehículos, productos no considerados como equipaje y mascotas (solo gatos y perros), tanto para su ingreso como salida del país.

## Supuestos y dependencias

Se espera que el sistema tenga capacidad de validación de documentos y conexión con bases de datos externas como el Registro Civil.

# ESTILOS Y PATRONES ARQUITECTÓNICOS

## Estilo arquitectónico adoptado

El estilo arquitectónico elegido es monolítico, lo que significa que todos los componentes se desarrollan y despliegan como una única aplicación unificada.

## Justificación del estilo según el contexto del sistema

Se eligió esta arquitectura por su simplicidad, rapidez de desarrollo y menor complejidad en el despliegue, ideal para un sistema inicial con alta integración entre módulos y equipo de desarrollo pequeño.

## Patrones de diseño aplicados

Se eligió el patrón Modelo-Vista-Controlador (MVC) porque permite separar la lógica de negocio, la interfaz de usuario y el control de flujo, facilitando el mantenimiento, la escalabilidad y la reutilización del código.

# MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS

## VISTA DE ESCENARIO

### **Propósito**

El propósito de la vista de escenario es definir el alcance de las funcionalidades del sistema, identificando los principales actores (ciudadanos, agentes aduaneros, sistemas externos como PDI y SAG) y las interacciones que estos tendrán. Esto permitirá entender cómo el sistema manejará el flujo de salida y entrada de vehículos, productos y mascotas en la frontera.

### **Actores**

**Agente Aduanero**: encargado de fiscalizar y validar la información y los documentos presentados por los ciudadanos para el ingreso o salida de vehículos, productos y mascotas.

Administrador del Sistema: gestiona las distintas aduanas, incluyendo su información (nombre, latitud, longitud, país fronterizo), y realizar operaciones CRUD sobre ellas.

**Jefe Operativo**: tiene acceso a la generación automática de informes estadísticos de ingresos y egresos de personas y vehículos.

**Ciudadanos**: Son los ciudadanos (chilenos o extranjeros) que hacen uso del paso fronterizo ingresando al país desde Argentina, o saliendo del país hacia Argentina. Deben llenar formularios de flujo de vehículos, productos, y adjuntar certificados para mascotas.

### **Diagrama general de casos de uso**

**Diagrama

El contenido generado por IA puede ser incorrecto.**

### **Diagrama de casos de uso específicos**

**Diagrama, Esquemático

El contenido generado por IA puede ser incorrecto.**

### **Lista de casos de uso (salida vehículo o entrada vehículo)**

| **Código** | **Nombre** | **Actores** |
| --- | --- | --- |
| **CU-CE-01** | Acceder a formulario de ingreso de vehículos | Ciudadano Entrante |
| **CU-CE-01-01** | Llenar información del vehículo | Ciudadano Entrante |
| **CU-CE-01-02** | Llenar información del conductor | Ciudadano Entrante |
| **CU-CE-01-03** | Llenar información del viaje | Ciudadano Entrante |
| **CU-CE-01-04** | Marcar vehículo de arrastre | Ciudadano Entrante |
| **CU-CE-01-05** | Adjuntar poder notarial | Ciudadano Entrante |
| **CU-AA-01** | Acceder a validación de entrada de vehículos | Agente Aduanero |
| **CU-AA-01-01** | Digitar patente del vehículo | Agente Aduanero |
| **CU-AA-02** | Acceder a formulario admisión temporal de vehículos | Agente Aduanero |
| **CU-AA-02-01** | Validar datos del conductor | Agente Aduanero |
| **CU-AA-02-02** | Validar datos del vehículo | Agente Aduanero |
| **CU-AA-02-03** | Validar información del viaje | Agente Aduanero |
| **CU-AA-02-04** | Validar existencia de poder notarial | Agente Aduanero |
| **CU-AA-02-05** | Cambiar estado de solicitud | Agente Aduanero |

### **Especificación de casos de uso**

| **Caso de Uso** | **Acceder a validación de entrada de vehículos** | **Identificador:**  **CU-AA-01** |
| --- | --- | --- |
| **Actores** | **Agente Aduanero** | |
| **Tipo** | **Caso de uso primario** | |
| **Referencias** |  | |
| **Precondición** | **-Agente Aduanero debe haber iniciado sesión en el sistema**  **-El ciudadano ha completado y enviado previamente el formulario de ingreso de vehículos** | |
| **Postcondición** | **-Información del vehículo, conductor, y viaje ha sido validada como correcta, incorrecta o incompleta**  **-el estado de la solicitud cambia**  **-Se genera un registro de la validación y el cambio de estado** | |
| **Descripción** | **Proceso mediante el cual el Agente Aduanero accede al sistema para inspeccionar un vehículo que intenta ingresar al país** | |
| **Resumen** | **El agente identifica el vehículo mediante su patente, revisa y valida la información del formulario de ingreso previamente completado por el ciudadano (conductor, vehículo, vehículo de arrastre opcional, información del viaje y documentos adjuntos como poder notarial si aplica). Finalmente, el agente actualiza el estado de la solicitud de ingreso en el sistema.** | |

**CURSO NORMAL**

| **Nro.** | **Ejecutor** | **Paso o Actividad** |
| --- | --- | --- |
| **1** | **Agente Aduanero** | **El Agente selecciona la opción en el sistema para acceder a la validación de entrada de vehículos** |
| **2** | **Sistema** | **El sistema solicita la patente del vehículo** |
| **3** | **Agente Aduanero** | **El Agente digita la patente del vehículo a inspeccionar** |
| **4** | **sistema** | **El Sistema busca y muestra el formulario de ingreso de vehículos correspondiente** |
| **5** | **Agente Aduanero** | **El Agente Aduanero valida los datos del vehículo y del dueño/conductor, que ya debería estar autorizada por PDI** |
| **6** | **Agente Aduanero** | **Si el vehículo no viaja con su dueño, el Agente verifica la existencia y validez del poder notarial** |
| **7** | **Agente Aduanero** | **El Agente constata que toda la información y documentación es correcta** |
| **8** | **Agente Aduanero** | **El Agente Aduanero selecciona la opción para cambiar el estado de la solicitud y elige "aceptado"** |
| **9** | **Sistema** | **El sistema guarda el nuevo estado y registra la validación exitosa** |

**CURSO ALTERNATIVO: Documentación Faltante o Errónea**

| **Nro.** | **Descripción de acciones alternas** |
| --- | --- |
| **1** | **El Agente Aduanero sigue los pasos 1-6 del Curso Normal** |
| **2** | **Durante la validación, el Agente Aduanero detecta que falta algún documento** |
| **3** | **El agente selecciona la opción para cambiar el estado de la solicitud y elige "pendiente"** |
| **4** | **El sistema guarda el nuevo estado "pendiente"** |

**CURSO ALTERNATIVO: Documentación Faltante o Errónea**

| **Nro.** | **Descripción de acciones alternas** |
| --- | --- |
| **1** | **El Agente Aduanero sigue los pasos 1-6 del Curso Normal** |
| **2** | **Durante la validación, el Agente determina que la información no cumple con los requisitos fundamentales para el ingreso (ej. prohibiciones, inconsistencias graves con PDI, poder notarial inválido)** |
| **3** | **El Agente Aduanero selecciona la opción para cambiar el estado de la solicitud y elige "rechazado"** |
| **4** | **El sistema permite ingresar los motivos del rechazo** |
| **5** | **El sistema guarda el nuevo estado "rechazado" y los motivos** |

**CURSO ALTERNATIVO: Vehículo no Encontrado en el Sistema**

| **Nro.** | **Descripción de acciones alternas** |
| --- | --- |
| **1** | **El Agente Aduanero sigue los pasos 1-3 del Curso Normal** |
| **2** | **El sistema no encuentra ningún formulario de ingreso asociado a la patente** |
| **3** | **El sistema muestra un mensaje indicando que no se encontró el vehículo** |

## VISTA LÓGICA

### **Propósito**

El propósito de este diagrama de clases es definir las entidades principales del sistema, sus atributos y relaciones entre ellas; con un enfoque en la gestión de usuarios, vehículos, y trámites para el control fronterizo. Este diagrama de clases busca servir como una guía para implementar la base de datos y la arquitectura de software.

### **Diagrama de clases**

**Diagrama

El contenido generado por IA puede ser incorrecto.**

### **Descripción diagrama de clases**

El diagrama de clases está agrupado para representar lógicamente diferentes aspectos del sistema. Cada clase presente en el diagrama posee atributos y métodos para crear, leer, actualizar y eliminar registros:

**Gestión de usuarios**: incluye clases como Ciudadano y Agente, las cuales heredan de Usuario. Un Usuario es una entidad genérica con nombre de usuario y contraseña, mientras que Usuario tiene atributos específicos como nombre, apellido, dirección y nacionalidad y está asociado a un tipo de identificación

**Gestión de vehículos**: posee clases como Vehículo, Modelo, Marca y Tipo de vehículo.

**Gestión de Formularios y Conductores**: La clase Formulario representa un documento con información detallada sobre un trámite, incluyendo la fecha, observaciones y la declaración de normas. Un Formulario está asociado a un Tipo de conductor específico (propietario o autorizado)

La clase Trámite Aduanero es importante para los procesos de aduana, con atributos como id de trámite, tipo de trámite y código de aduana. Está asociado a un tipo de trámite.

## VISTA DE IMPLEMENTACIÓN/DESARROLLO

### **Propósito**

El propósito de este diagrama de componentes es representar la estructura modular del sistema de control de entrada en pasos fronterizo, mostrando cómo se dividen las responsabilidades entre los diferentes componentes de software y cómo se comunica entre sí. Este diagrama guía el desarrollo, mantenimiento y despliegue del sistema, asegurando la correcta integración entre la interfaz usuario, la lógica de negocio y los servicios externos presentado con anterioridad como PDI, Notaría y Registro civil.

### **Diagrama de componente**

Diagrama

El contenido generado por IA puede ser incorrecto.

### **Descripción diagrama de componente**

Actor (Usuario o ciudadano externo): interactúa directamente con el sistema interfaz web.

**Interfaz Web**: presenta los formularios al usuario y se conecta con el gestor de formularios.

**Gestor Formulario**: el gestor Formulario coordina la recolección y envío de datos del formulario. Este se comunica con validar datos, para verificar la información ingresada.

Almacenamiento de solicitudes, guardará el formulario.

Formulario con extensión java, es una clase de implementación que interactúa con el componente validar datos.

**Validar Datos**: verifica los datos recibidos. Consulta el almacenamiento de solicitudes para revisar datos previos. se comunicará con tres componentes de integración externos.

**Integración PDI - integración Notaria - integración Registro civil**: Conectan con sus respectivos sistemas externos, permite validar antecedentes y registro oficiales del ciudadano.

**Almacenamiento de solicitud**: guarda los datos enviados por el usuario, es utilizado por el gestor de formularios, como y para validar datos junto con el gestor de solicitudes.

**Gestor de solicitudes**: el gestor de solicitudes es utilizado por el agente aduanero para consultar, revisar y aprobar o rechazar solicitudes ingresadas.

**Agente aduanero**: Actor que interactúa con el gestor de solicitudes desde el lado operativo del sistema.

### **Diagrama de paquete**

Diagrama

El contenido generado por IA puede ser incorrecto.

### **Descripción diagrama de paquete**

Este diagrama de paquetes ilustra la organización del sistema siguiendo una arquitectura que parece ser Modelo-Vista-Controlador (MVC) con una capa DAO

**Vista** contiene las interfaces de usuario, encargándose de la presentación.

**Controlador** gestiona la lógica de la aplicación y las interacciones entre la Vista y el Modelo

**Modelo** representa los datos y la lógica de negocio, con entidades como FORMULARIO, VEHICULO y una conexión a ORACLEDB.

**Clase** contiene las clases de entidad Java que representan los objetos como Vehiculo.java o Usuario.java.

**DAO** se encarga de la comunicación y persistencia de los datos con la base de datos, utilizando las clases del paquete Clase e interactuando con el Modelo.

## VISTA DE PROCESOS

### **Propósito**

El objetivo de este diagrama de actividad es modelar el flujo de trabajo involucrado en el proceso de control fronterizo de vehículos. Representa las actividades que se realizan desde la llegada del vehículo al paso fronterizo hasta su autorización o denegación del ingreso. Este diagrama busca guiar la implementación de la lógica de negocio y las validaciones necesarias en el sistema.

### **Diagrama de actividad**

Diagrama

El contenido generado por IA puede ser incorrecto.

### **Descripción diagrama de actividad**

Este diagrama de actividad muestra el flujo del proceso de control de entrada de vehículos en un paso fronterizo. Inicia con la llegada de un vehículo, seguido por la verificación de datos del conductor y del vehículo, la validación se realiza a través de un formulario que debe ser completado con anterioridad por el ciudadano entrante o extranjero.

Si el formulario no cumple con los requisitos establecidos, es sistema denegará automáticamente el acceso, este diagrama apoya la definición de reglas de negocio y su automatización.

## VISTA FÍSICA

### **Propósito**

El propósito de este diagrama de despliegue es mostrar cómo los componentes de software del sistema de control fronterizos se distribuyen físicamente en la infraestructura tecnológica. Mostrando las distribuciones de los componentes de software de los distintos nodos. Esto incluye (servidores, clientes, bases de datos y clientes externos). El diagrama de despliegue ayuda a entender cómo interactúan las aplicaciones con las bases de datos externas como la Policía de investigaciones “PDI” y el Registro Civil y así mantener la infraestructura del sistema, asegurando rendimiento, seguridad y disponibilidad.

### **Diagrama de despliegue**

Diagrama

El contenido generado por IA puede ser incorrecto.

### **Descripción diagrama de despliegue**

Este diagrama presenta la arquitectura física del sistema de control en el paso fronterizo, incluyendo los siguientes nodos principales.

Pcs o dispositivos móviles (Cliente): Utilizados por los usuarios (agentes aduaneros o funcionarios) para acceder, mediante navegador web o apps, a portales institucionales como www.pid.cl, www.registrocivil.cl y www.sna.cl.

Servidores web: Ejecución de componentes como el servidor HTTP/HTTPS, el historial de acciones, Logs del sistema y una interfaz de base datos para conectar con la base de datos externa.

Bases de datos (DBM y SQL) distribuidas en varios servidores.

una base de datos central del sistema.

una base de datos en el servidor de la PDI.

una base de datos en el servidor del registro civil.

Servicios externos: como PDI y Registro Civil, con sus propios servidores y base de datos.

Actor externo: representa al usuario (actor 1), quien interactúa con distintos portales para completar el proceso de registro o validación antes de presentarse físicamente en el paso fronterizo Sistema Nacional de Aduanas (SNA).

# REQUISITOS DE CALIDAD

## Propósito

El propósito de definir requisitos de calidad es establecer criterios claros y medibles que el sistema debe cumplir para satisfacer las expectativas de los usuarios y las partes interesadas.

## Atributos de calidad

| **ATRIBUTO DE CALIDAD** | **DESCRIPCION** | **JUSTIFICACIÓN** |
| --- | --- | --- |
| **Rendimiento** | **Rapidez de respuesta** | **La rapidez de respuesta del sistema en operaciones críticas es vital para minimizar tiempos de espera de vehículos y viajeros** |
| **Escalabilidad** | **Manejar cargas crecientes** | **La capacidad de manejar cargas crecientes sin degradación asegura la adaptación a flujos de tráfico variables y futuras expansiones** |
| Interoperabilidad |  | **La integración con sistemas externos (Aduana Argentina, PDI, SAG) deben hacerce utilizando APIs con protocolos estándar (REST/JSON), seguras, bien documentadas y actualizadas** |
| **Mantenibilidad** | **Facilidad y rapidez para corregir errores** | **La facilidad y rapidez para corregir errores o realizar modificaciones permite una adaptación constante y una efectividad operativa a largo plazo** |

## Reglas y criterios de evaluación de Calidad

Rendimiento: tiempo de respuestas <2 segundos para la mayoría de operaciones críticas.

Tiempo de procesamiento de documentos vehiculares <5 segundos.

Pruebas de escalabilidad:

soporte para duplicar las soluciones sin degradación del servicio <10% de pérdida de rendimiento.

Monitoreo continuo con Prometheus:

Disponibilidad: tiempo de disponibilidad > 99,9% mensual.

Mantenibilidad: tiempo medio de corrección de errores críticos 4 horas, revisión de códigos a estos errores, seguimiento con herramientas de gestión de incidencias (jira).

Pentesting: pruebas de seguridad con accesos auditables y autenticados, revisión de códigos, escáneres de seguridad basado en cumplimiento OWASP.

# PRINCIPIOS DE DISEÑO APLICADOS

## Propósito

El propósito de esta sección es establecer los principios fundamentales que guían las decisiones arquitectónicas y diseño del sistema, estos principios aseguran que el sistema sea robusto, seguro, mantenible y alineado con los objetivos estratégicos de interoperabilidad, escalabilidad y continuidad operativa exigidos por el contexto de pasos fronterizos y sus políticas públicas.

## Principios de diseño

| **PRINCIPIO** | **DESCRIPCIÓN** | **APLICACIÓN EN EL SISTEMA** |
| --- | --- | --- |
| Cohesion | Cada módulo o clase tiene una única responsabilidad bien definida. | Los servicios están diseñados para realizar tareas específicas y no múltiples funciones |
|  |  |  |
| Escalabilidad horizontal | la arquitectura soporta escalabilidad horizontal para adaptarse a las variantes de alto flujo de personas y vehículos | control de identidad y documentación, reconocimiento vehicular y validación de patente. |

# 7 PROTOTIPO

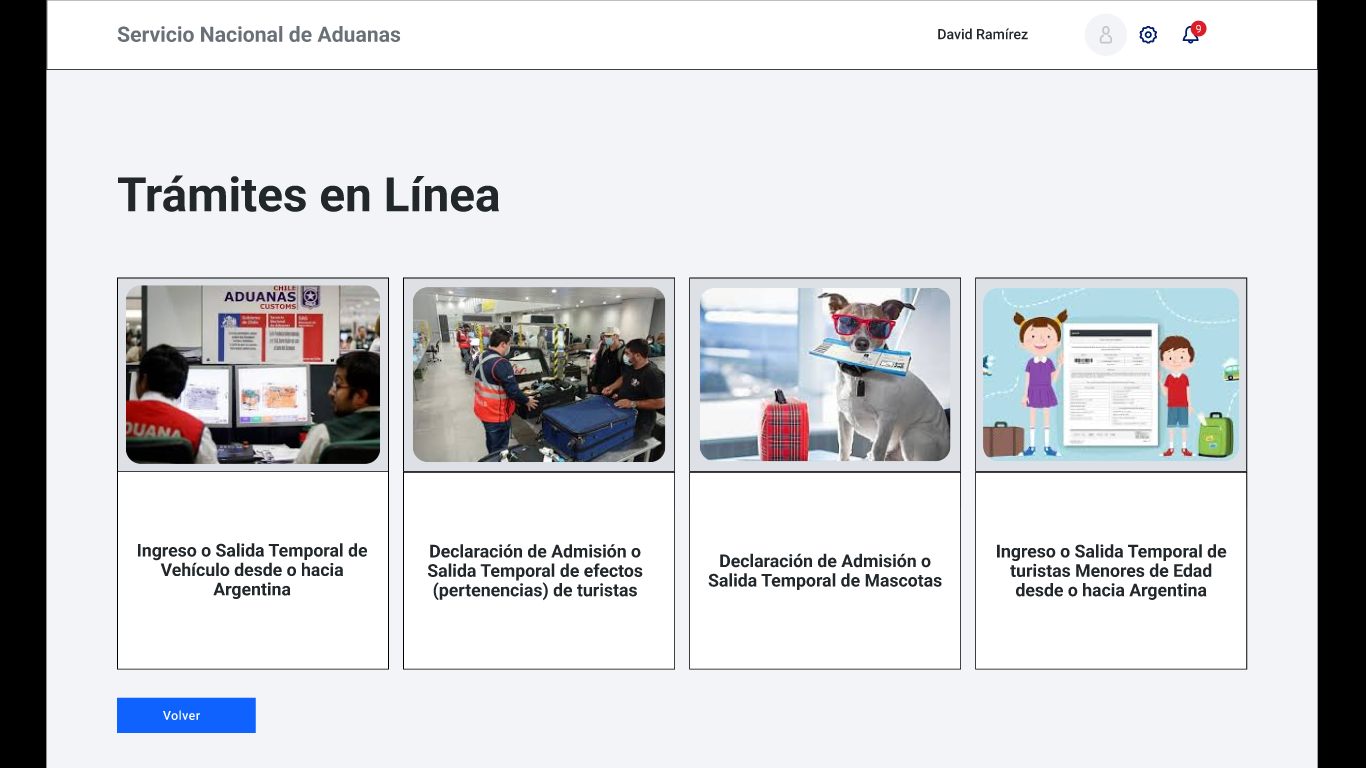
## **7.1.** **Propósito**

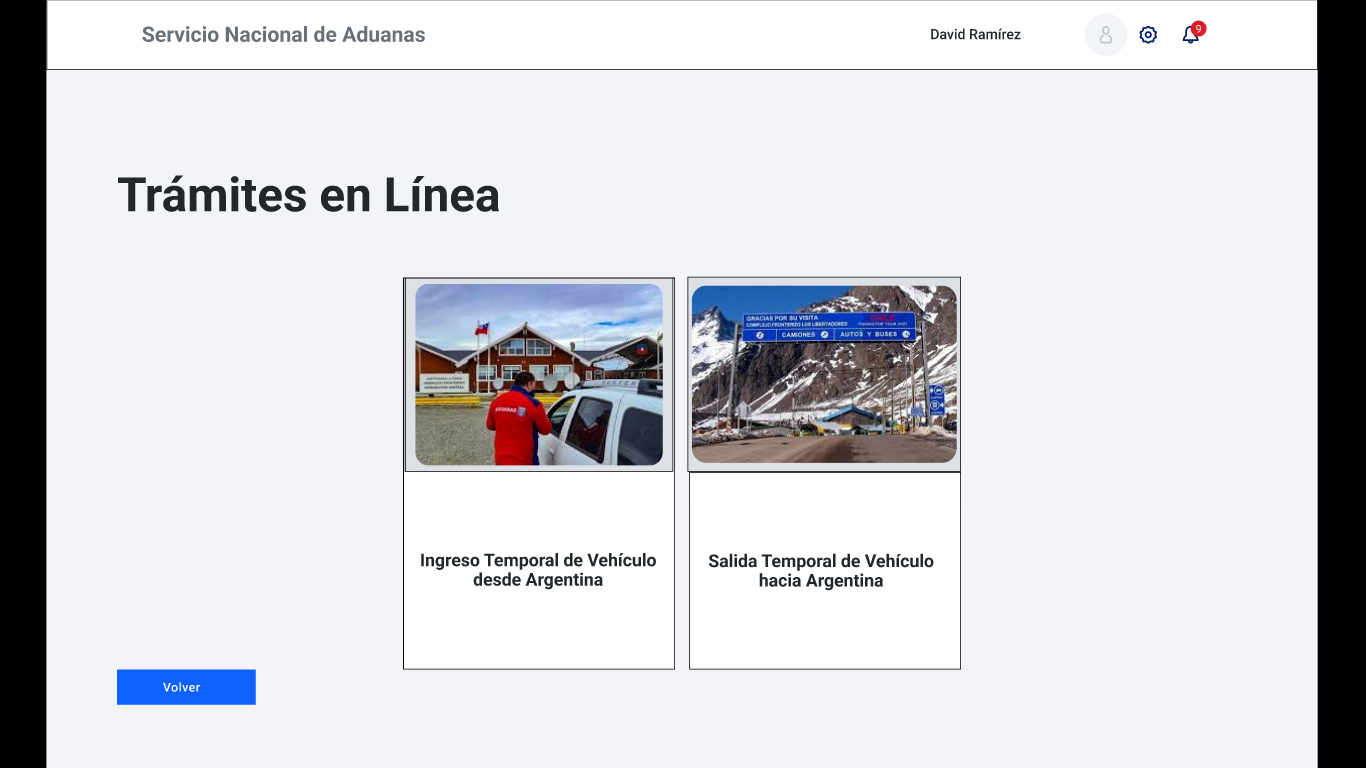
El prototipo mockup tiene como propósito representar de forma visual y funcional los flujos principales del sistema de software para el ingreso de vehículos al país. Este diseño simula el recorrido que debe seguir un turista para registrar una solicitud de ingreso vehicular, así como el proceso que realiza un agente aduanero para revisar dichas solicitudes y proceder a validarlas, rechazarlas o marcarlas como pendientes. El mockup permite identificar de manera temprana la experiencia del usuario, optimizar la navegación entre pantallas y validar la lógica operativa antes del desarrollo final del sistema.

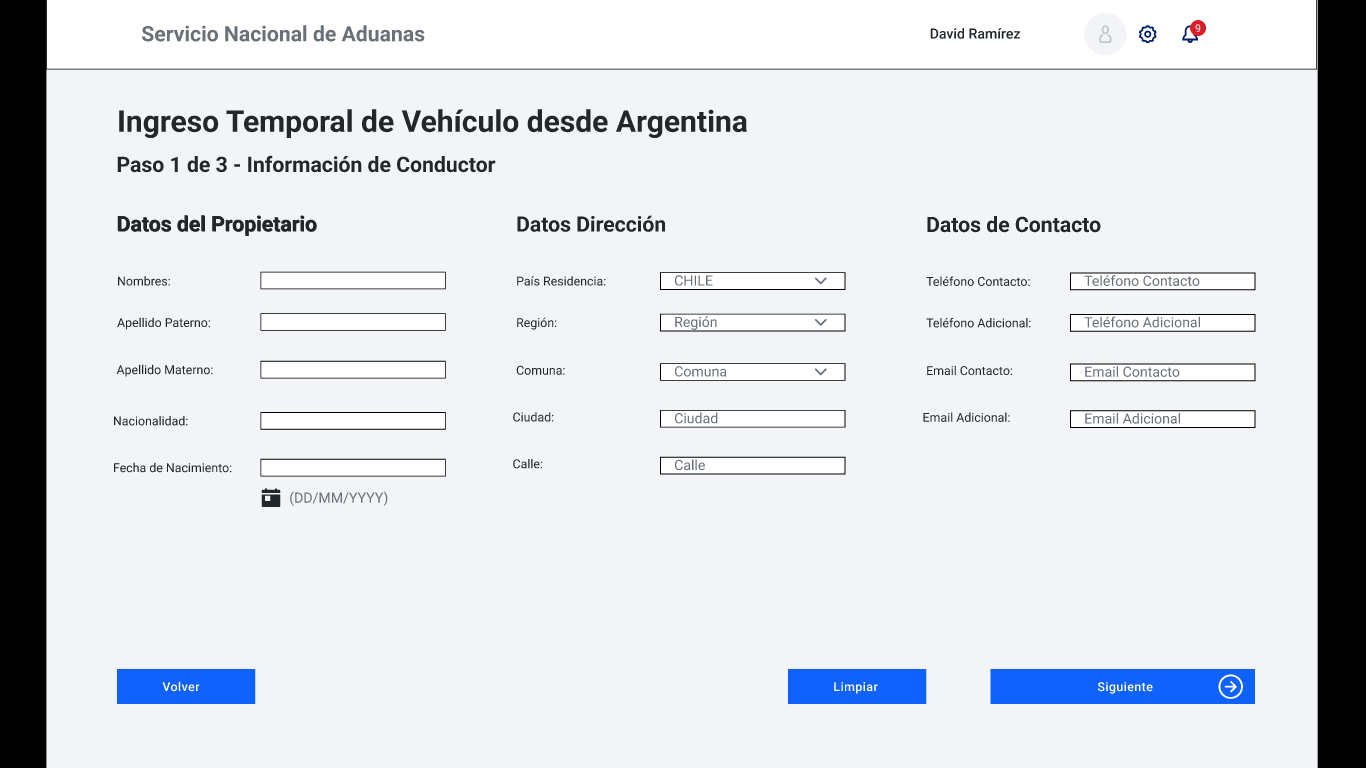
Enlace para acceder a prototipo elaborado en Figma: [Sistema Ingreso de Vehículos - Figma](https://www.figma.com/proto/sy7fY9iF4YJ8wtq3AberYQ/PROTOTIPO?node-id=93-240&t=cKu4wXoSBdX1af0K-1)

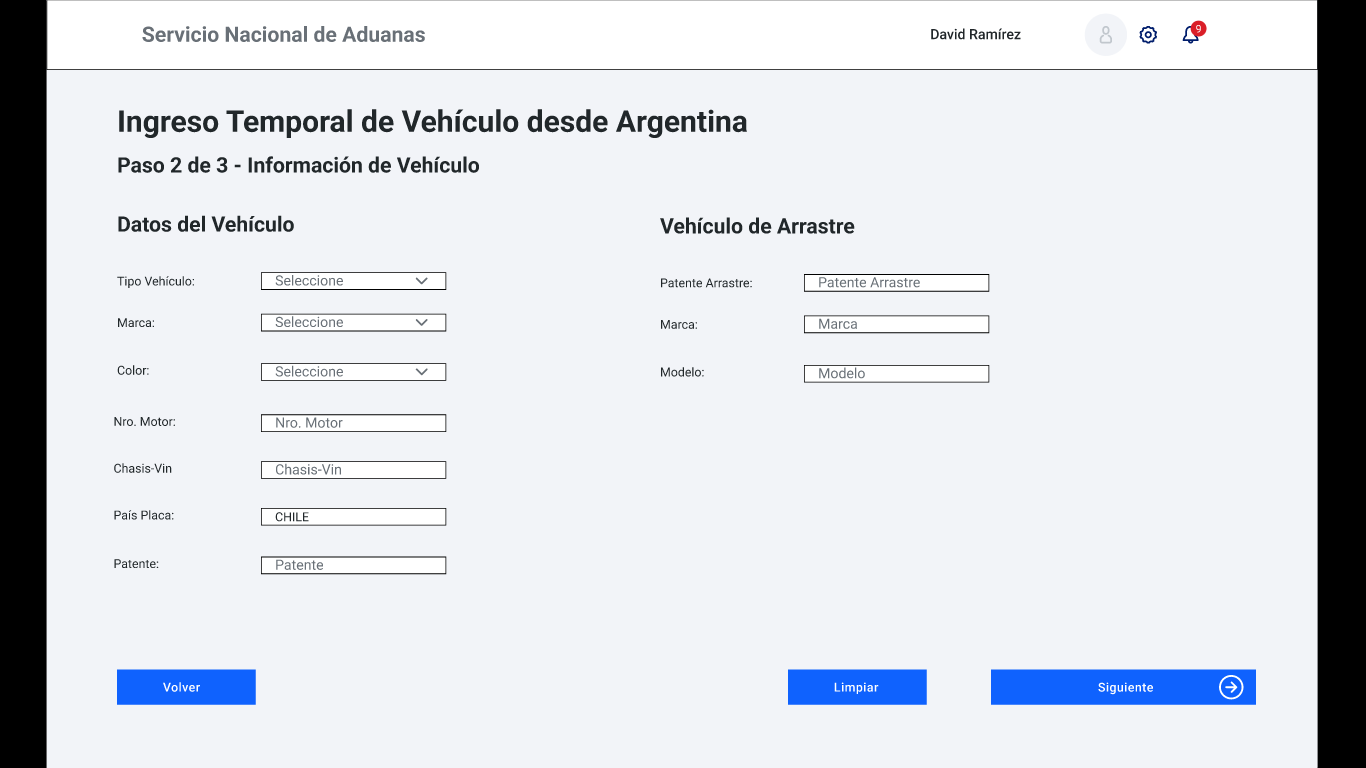
## **7.2.** **Mockups**

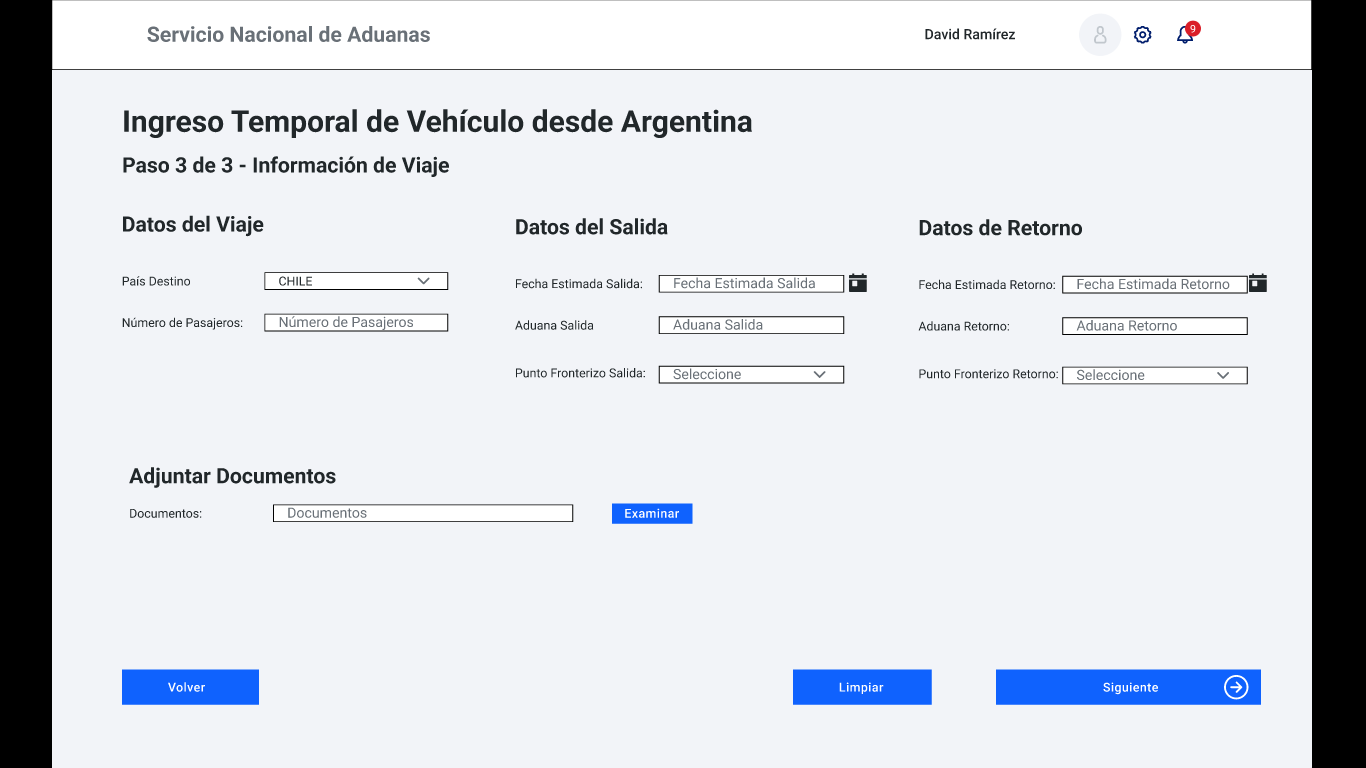
### **7.2.1 Recorrido que debe seguir un turista para registrar una solicitud de ingreso vehicular**

****

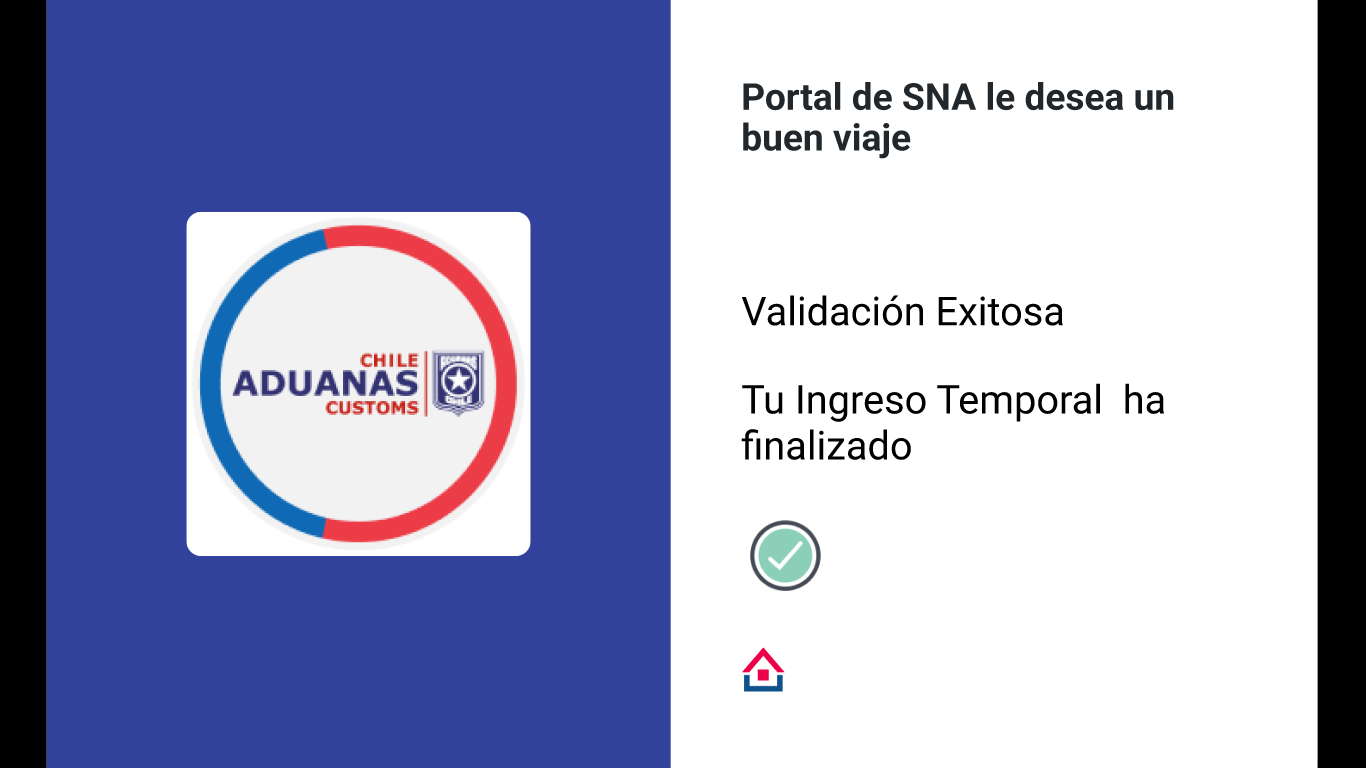
****

****

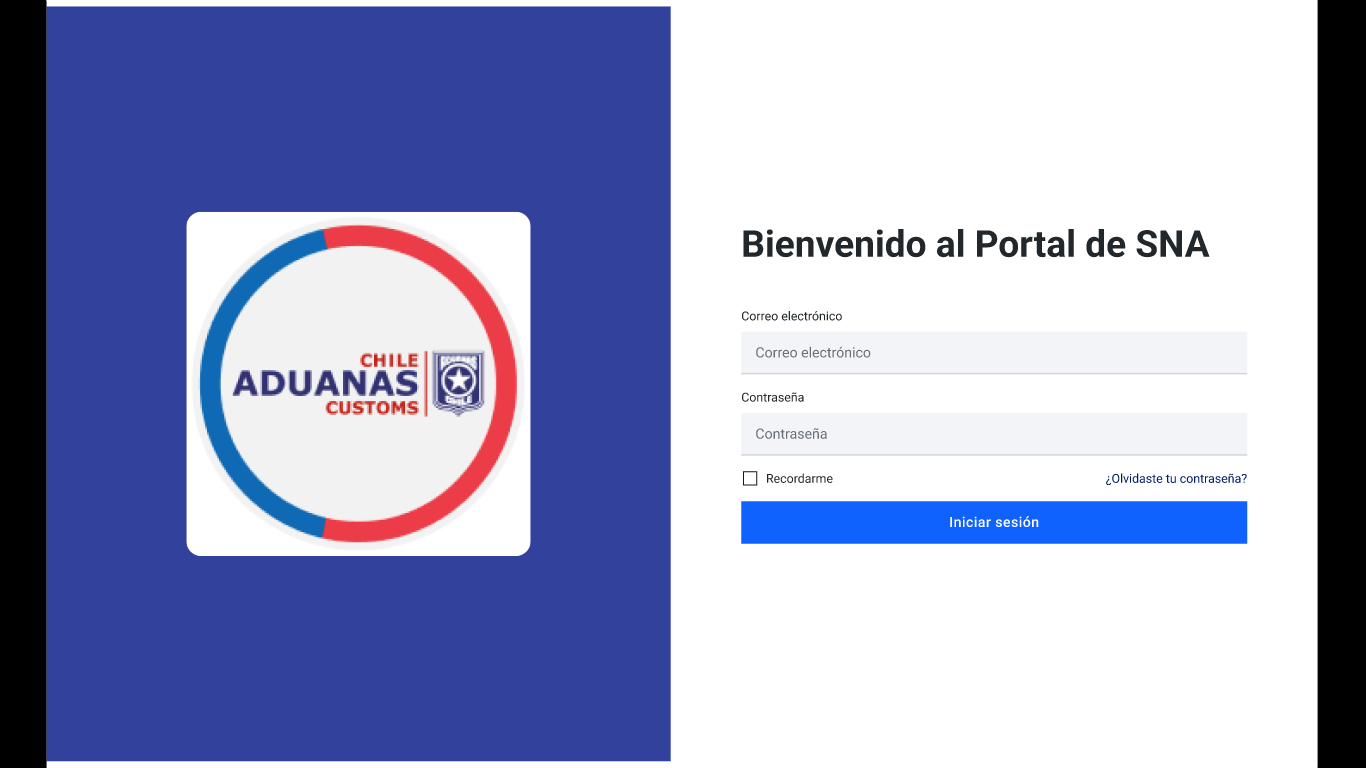
****

****

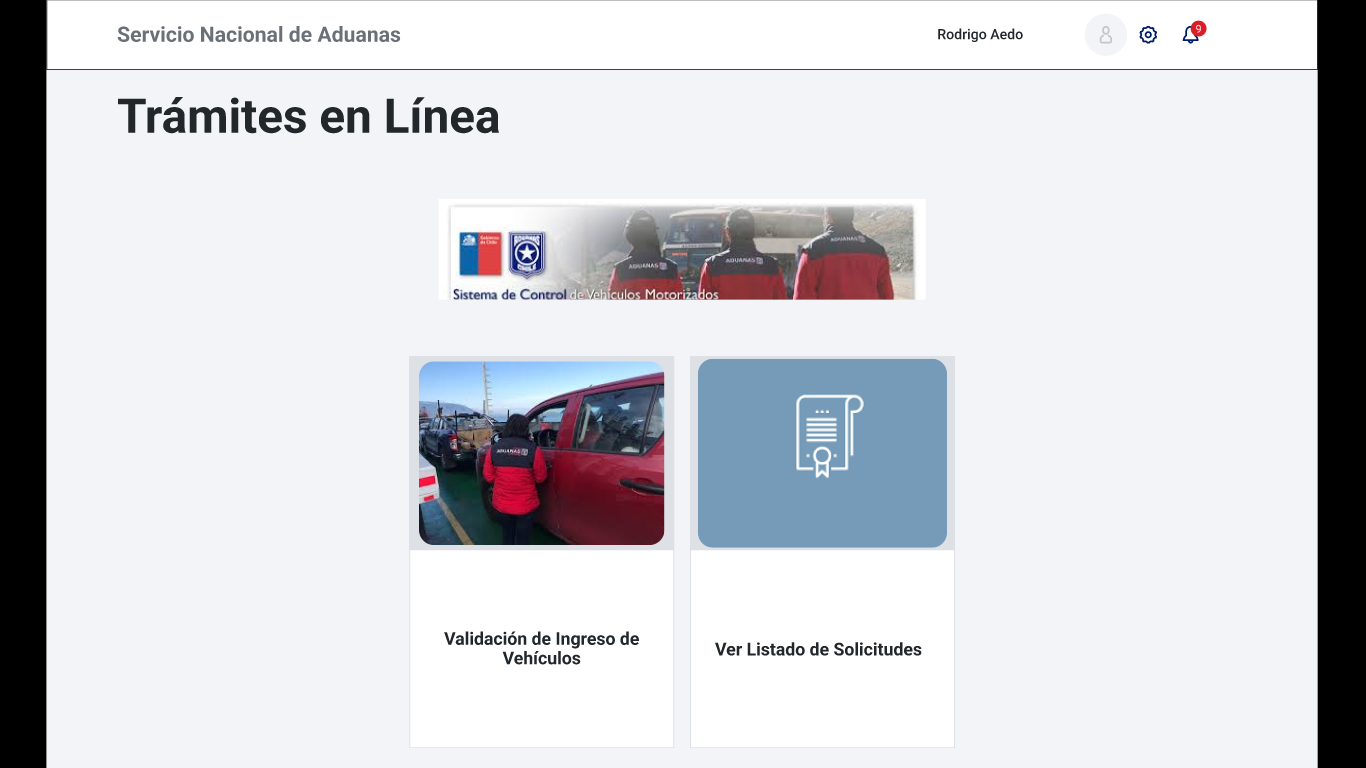
****

****

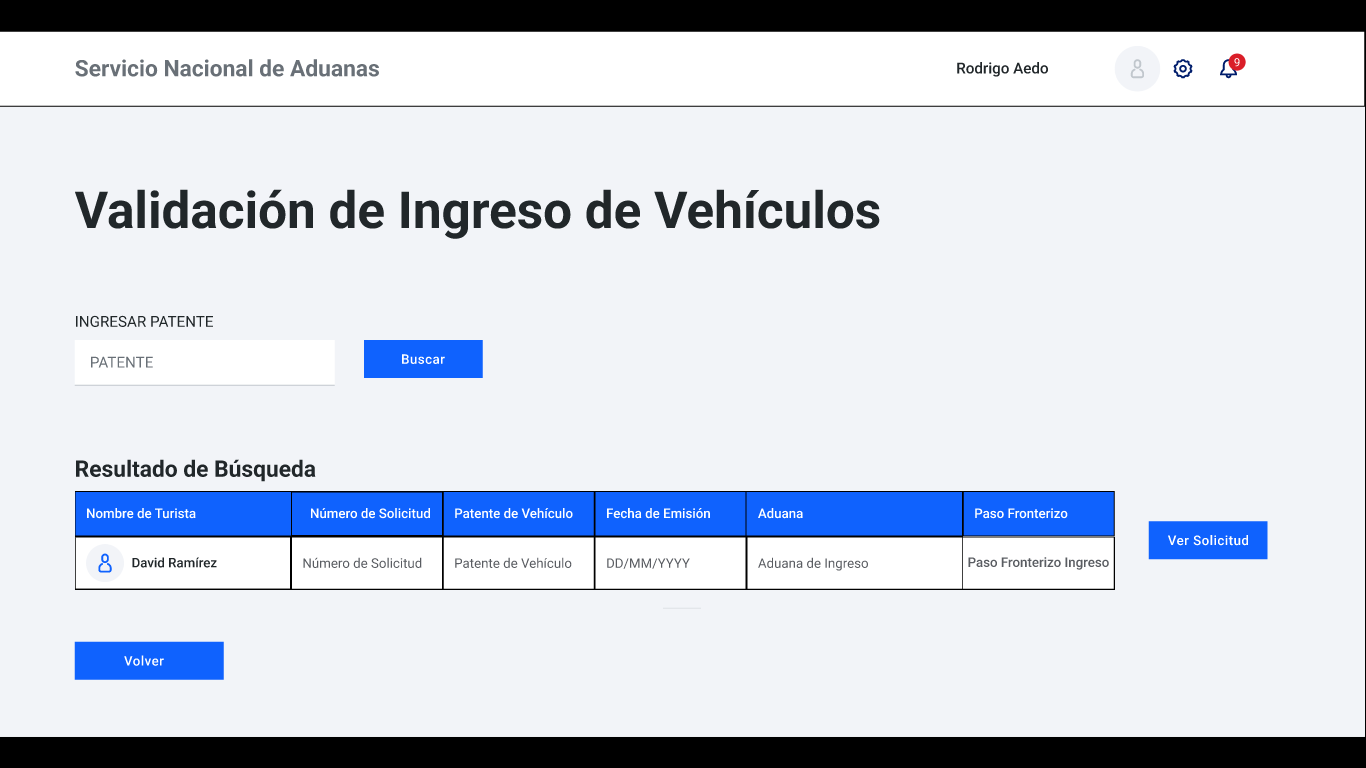
### **7.2.2 Proceso que realiza un agente aduanero para revisar dichas solicitudes y proceder a validarlas, rechazarlas o marcarlas como pendientes**

****

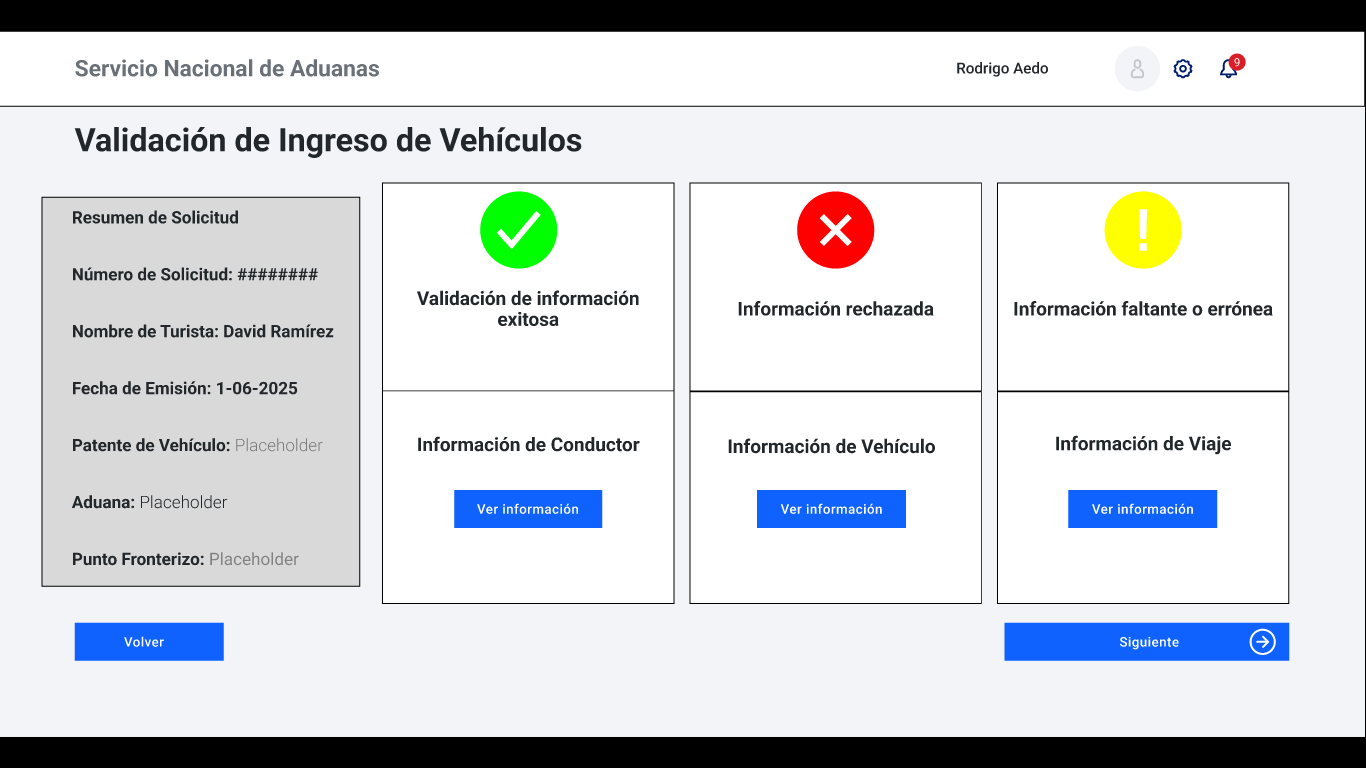
Agente Aduanero debe haber iniciado sesión en el sistema

****

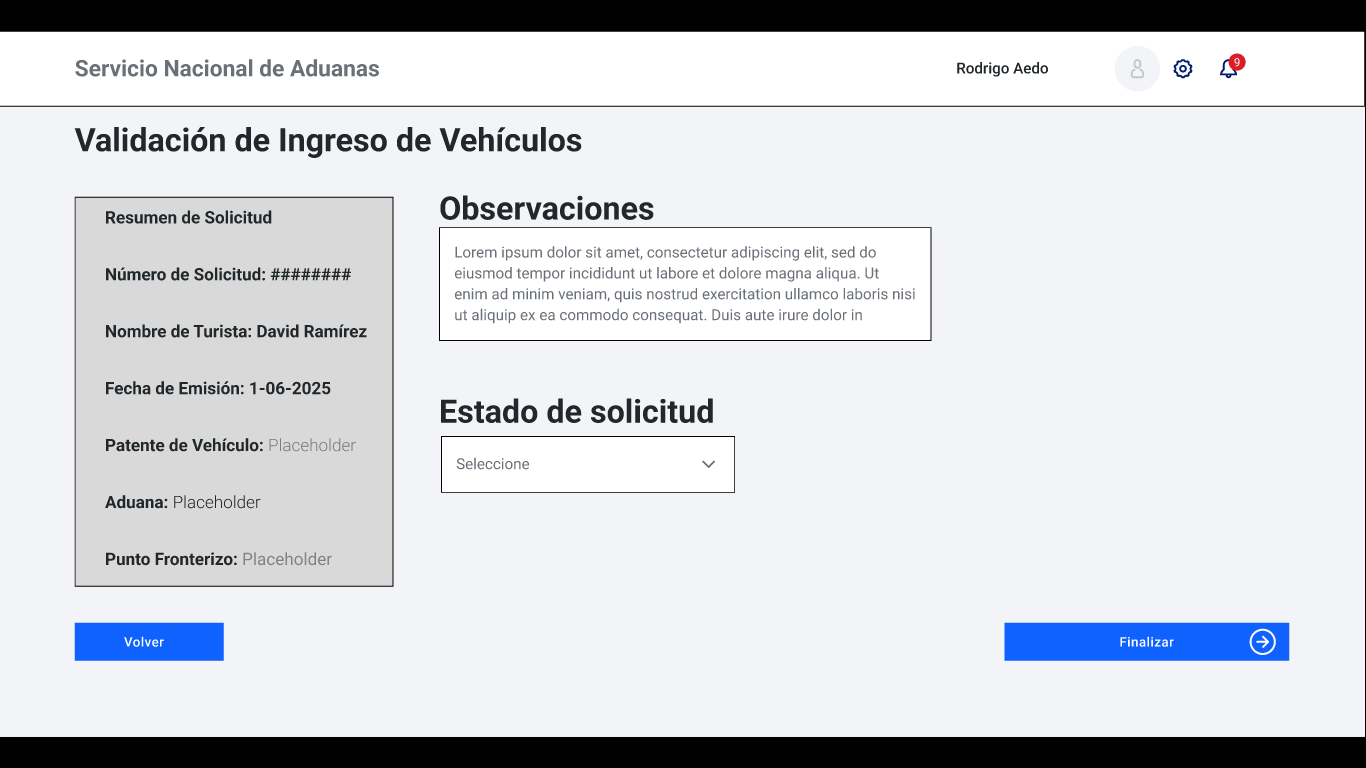
El Agente selecciona la opción en el sistema para acceder a la validación de entrada de vehículos

****

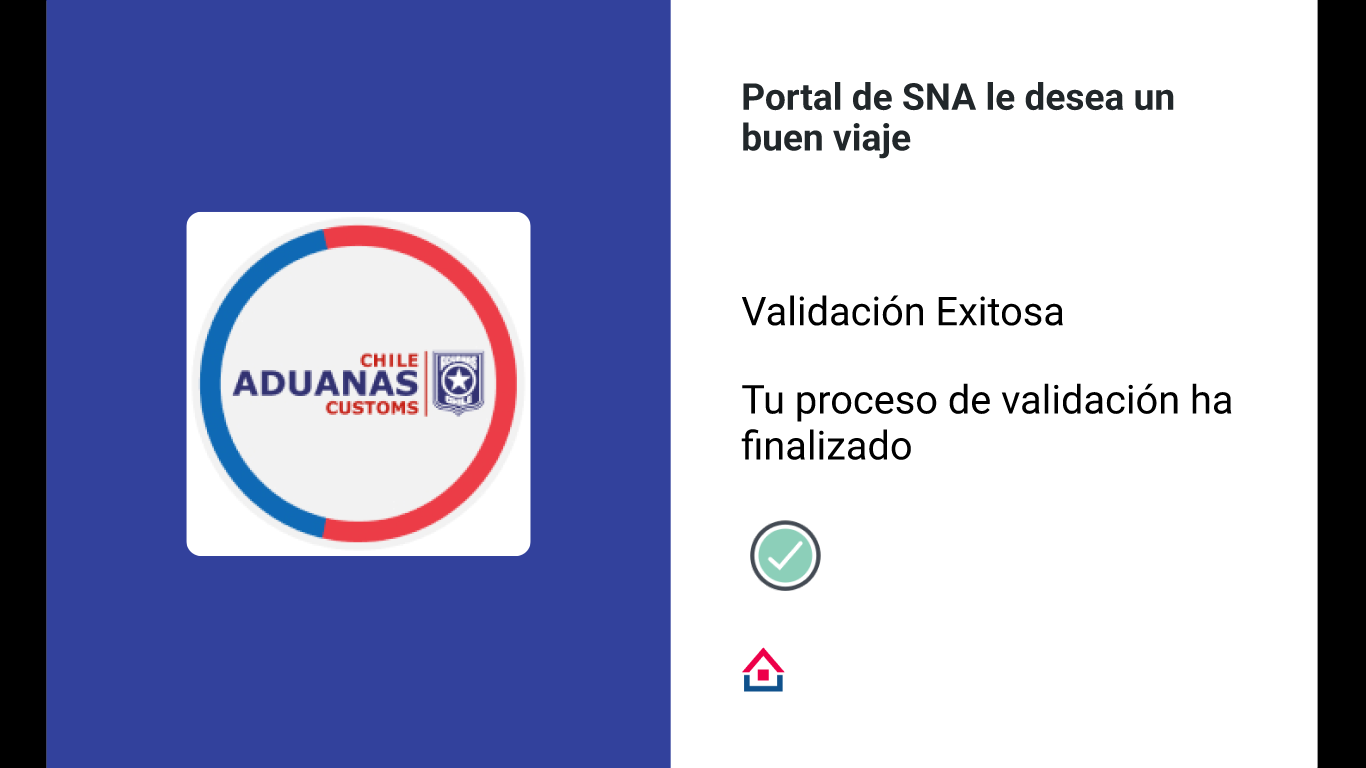
El sistema solicita la patente del vehículo. El Agente digita la patente del vehículo a inspeccionar. El Sistema busca y muestra el formulario de ingreso de vehículos correspondiente.

****

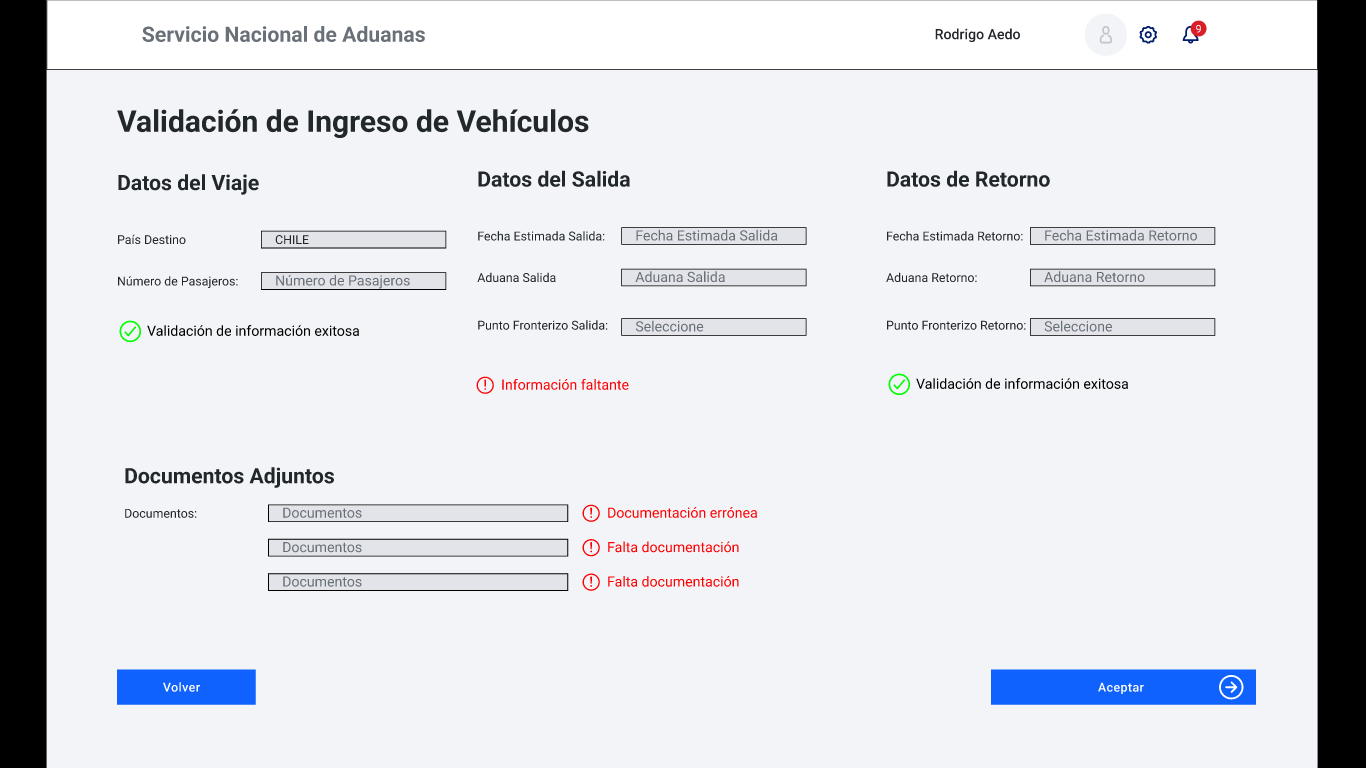
El Agente Aduanero valida los datos del vehículo y del dueño/conductor, que ya debería estar autorizada por PDI. El Agente constata que toda la información y documentación es correcta

****

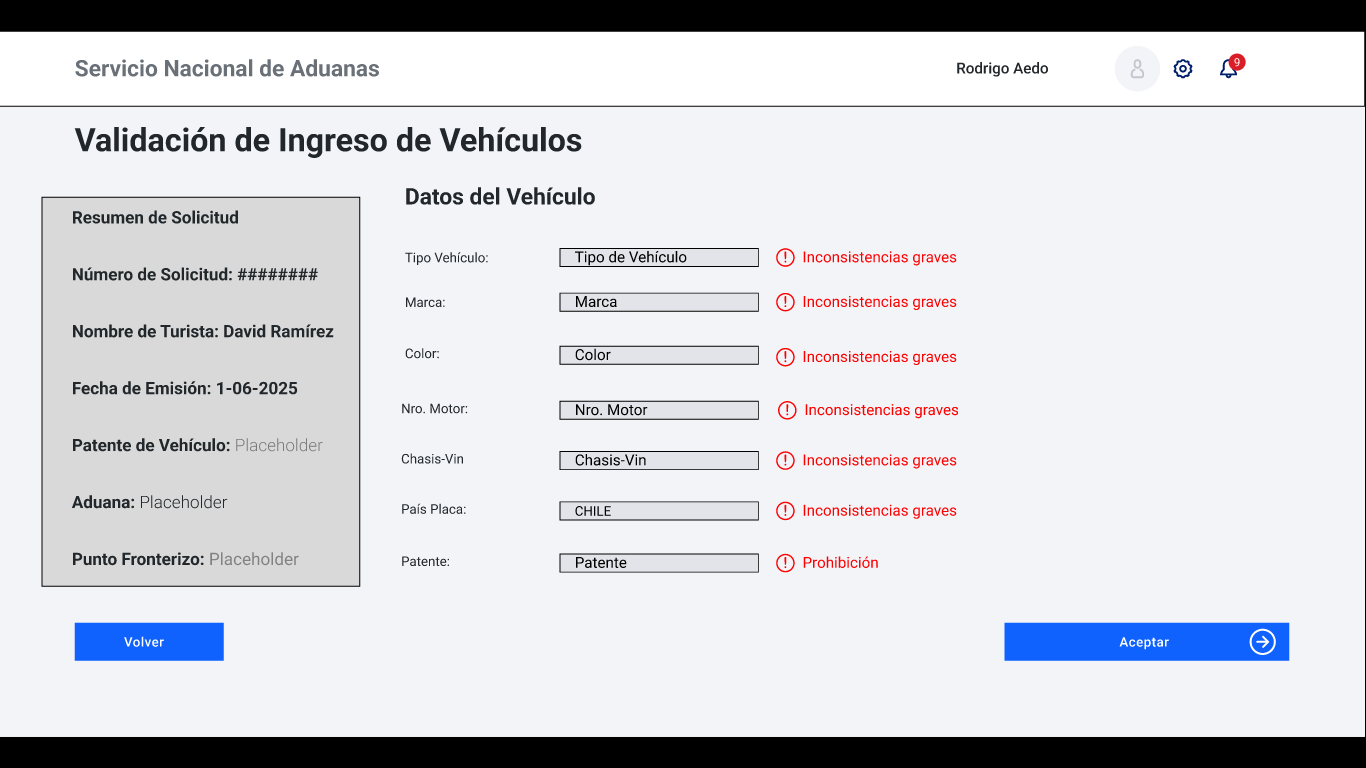
El Agente Aduanero selecciona la opción para cambiar el estado de la solicitud y elige "aceptado". El sistema permite ingresar los motivos en caso de que estado de solicitud sea “rechazado” o “pendiente”

****

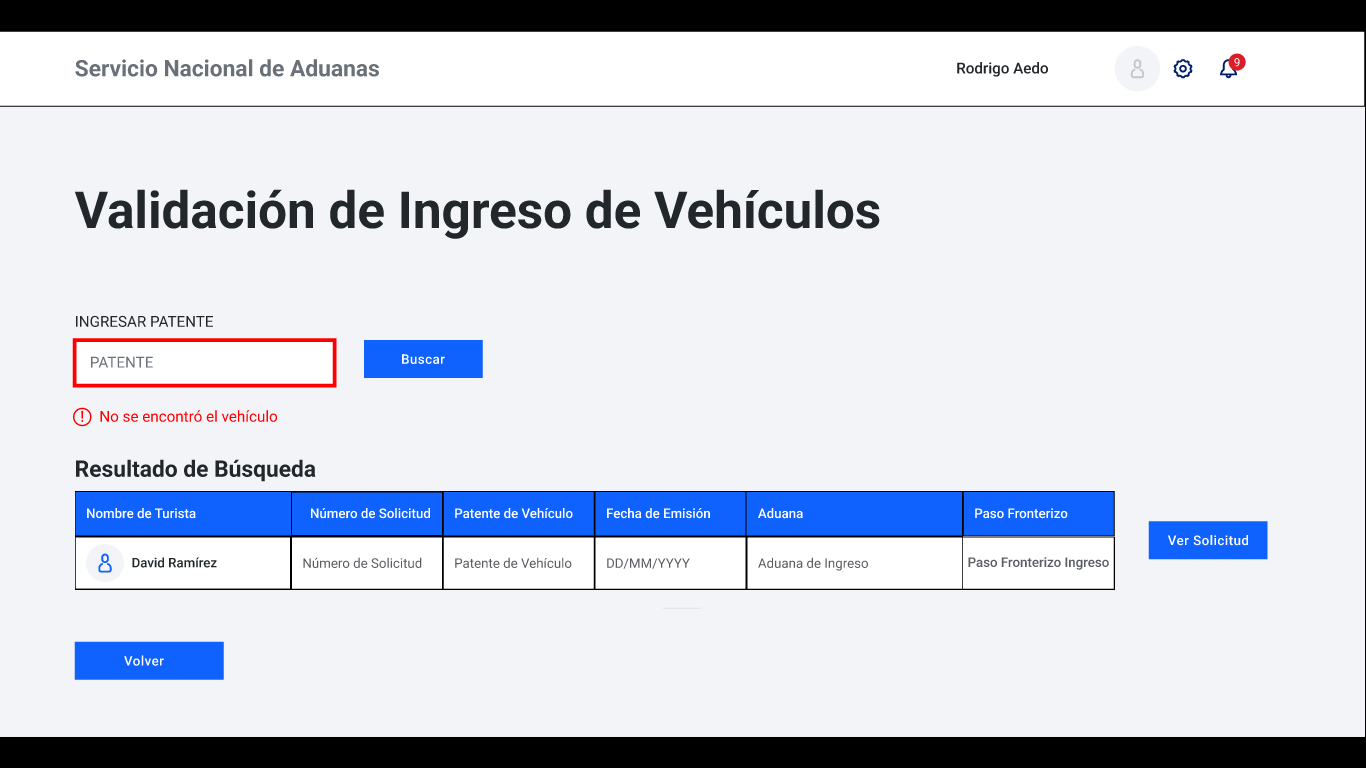
El sistema guarda el nuevo estado y registra la validación exitosa

****

CURSO ALTERNATIVO: Documentación Faltante o Errónea: Durante la validación, el Agente Aduanero detecta que falta algún documento. El agente selecciona la opción para cambiar el estado de la solicitud y elige "pendiente".

****

CURSO ALTERNATIVO: Documentación Faltante o Errónea: Durante la validación, el Agente determina que la información no cumple con los requisitos fundamentales para el ingreso (ej. prohibiciones, inconsistencias graves con PDI, poder notarial inválido). El Agente Aduanero selecciona la opción para cambiar el estado de la solicitud y elige "rechazado".

****

CURSO ALTERNATIVO: Vehículo no Encontrado en el Sistema: El sistema no encuentra ningún formulario de ingreso asociado a la patente. El sistema muestra un mensaje indicando que no se encontró el vehículo.

### **7.3. Justificar herramientas de prototipado**

Nuestro equipo decidió usar Figma como herramienta de prototipado porque permite diseñar interfaces de forma colaborativa en tiempo real desde cualquier navegador, sin necesidad de instalar software. Figma tiene una comunidad de diseñadores y desarrolladores muy activa, los cuales están constantemente compartiendo componentes reutilizables, esto ayudó a optimizar nuestro tiempo de diseño y a mantener consistencia en componentes e interfaces.

# 8 EVALUACIÓN DE CALIDAD HEURÍSTICA DE NIELSEN

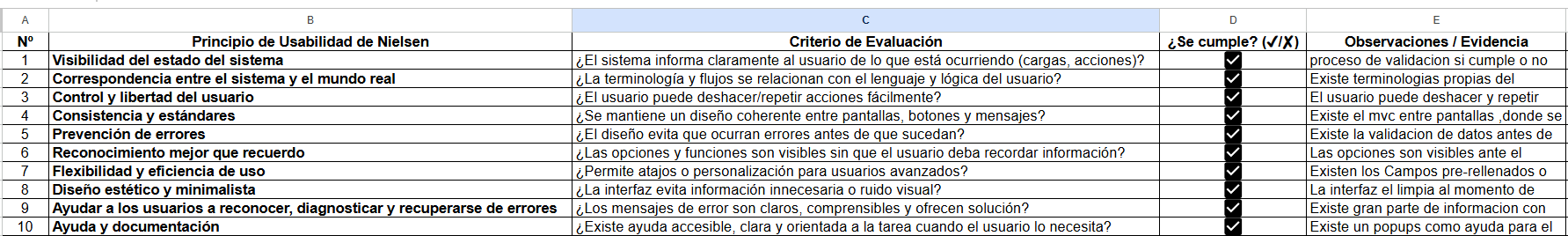
## 8.1. Propósito

**Propósito de aplicar los principios de usabilidad de Nielsen:**

Utilizar los 10 principios de usabilidad de Nielsen permite evaluar y diseñar interfaces orientadas al usuario, asegurando que el sistema sea intuitivo, consistente y fácil de usar. Su propósito es mejorar la experiencia del usuario, reducir errores, facilitar el aprendizaje y garantizar una interacción eficiente y satisfactoria con el sistema.

## 8.2. Lista de verificación

Enlace a Planilla de Calidad de Software de Nielsen: [**Planilla de Calidad de Software de Nielsen**](https://docs.google.com/spreadsheets/d/1O-8Q4J4CPKvc8IgcejJODmZWOSWFLcht/edit?gid=1484671116#gid=1484671116)



## 8.3. Análisis y métricas de resultados

Se aplicaron los 10 principios de usabilidad de Nielsen al diseño del sistema. En general, el proyecto cumple adecuadamente en aspectos como consistencia,prevención de errores y uso de terminología clara para los usuarios.

Sin embargo, se identificaron áreas de mejoras en el control y libertad del usuario, como mensajes de error más claros y ayuda contextual.

# 9 CONTROL DE VERSIONES

## 9.1. Propósito

El propósito de utilizar control de versiones en nuestro sistema de software de ingreso de vehículos es gestionar de forma ordenada y segura los cambios. Si bien este proyecto no contempla la mantención de código fuente, el control de versiones nos permite llevar un historial de las modificaciones y facilitar el trabajo colaborativo, revertir errores, mantener diferentes versiones de la documentación y asegurar la integridad del proyecto durante su evolución.

Enlace a repositorio de Github: [**Repo Github - Sistema de Ingreso de Vehículos para SNA**](https://github.com/rodrigo010101/IngenieriaSoftaware)

## 9.2. Control de versión utilizado

Como equipo tomamos la decisión de utilizar control de versiones semántico, pues permite establecer un esquema claro y predecible para identificar los cambios en un sistema de software. Para las diferentes versiones se está usando el formato MAJOR.MINOR.PATCH, donde Major indica una versión incompatible con las anteriores, Minor introduce nuevas funcionalidades compatibles con la versión actual y Patch aumenta e indica corrección de errores sin alterar las funcionalidades actuales. Al momento de escribir esta documentación, la versión del proyecto es identificada por la secuencia 1.0.0, es decir hemos alcanzado una versión estable y funcional básica. Cuando empezamos la fase inicial hace algún tiempo, la versión era 0.0.0, es decir nos encontrábamos en una fase inicial de desarrollo que no garantiza ni estabilidad ni compatibilidad, donde todavía no se aplicaban formalmente las reglas del negocio.

## 9.3. Justificar herramientas de versionamiento

1. Control de cambios detallado: para un sistema aduanero, la transparencia y la capacidad de rastrear cualquier modificación son vitales para la seguridad y el cumplimiento normativo, es por esto que Git es la herramienta preferida como sistema de control de versiones distribuidas. Cada “commit” o cambio, crea un historial rastreable, con el autor y la fecha de la modificación.
2. Por otra parte, Github nos ofrece una plataforma centralizada de colaboración que almacena todas las versiones de código, que facilita la comunicación y agiliza el flujo de trabajo.
3. De acuerdo a nuestra experiencia en desarrollo de software, estamos acostumbrados a trabajar los proyectos localmente. en nuestras propias máquinas las cuales muchas veces están expuestas a fallas. Durante estas situaciones, Github proporciona una capa crucial de protección contra la pérdida de datos, al actuar como un respaldo remoto de nuestros repositorios de código Git; el código fuente completo y su historial están seguros y accesibles en la nube de GitHub.

# 10 CONCLUSIONES

Establecer reglas claras y criterios de evaluación de calidad es fundamental para garantizar que el sistema cumpla con los objetivos funcionales y no funcionales definidos. Estos parámetros permiten medir el desempeño, la usabilidad, la seguridad y otros atributos clave mediante herramientas y métodos apropiados, asegurando así la entrega de un producto confiable, eficiente y alineado con las expectativas de los usuarios y las partes interesadas. Además, proporcionan una base objetiva para la mejora continua y la toma de decisiones a lo largo del ciclo de vida del sistema.