**[Proyecto de Aduanas]**

**(DAS) Documento Arquitectura de Software**

**Versión 1.0**

**Identificación de Documento**

| **Identificación** | Equipo 3 |
| --- | --- |
| **Proyecto** | Proyecto Aduana AR-CH |
| **Versión** | V.1 |

| **Documento mantenido por** |  |
| --- | --- |
| **Fecha de última revisión** | 27/05/2025 |
| **Fecha de próxima revisión** | - |

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

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 23/05/2025 | V.1 | Primera inserción de los datos necesarios (Modelo 4+1) | Luciano Beneventi, Felipe Rubio y Demesio Monsalves |
| 24/05/2025 | V.2 | Segunda inserción de datos necesarios (Introducción, Visión del sistema, Requisitos de calidad, Principio de diseño) | Luciano Beneventi y Felipe Rubio |
| 25/05/2025 | V.3 | Tercera inserción de datos necesarios y revisión | Demesio Monsalves |
| 26/05/2025 | V.4 | Ampliar punto 3 | Luciano Beneventi |
| 27/05/2025 | V.5 | Incluir identificación de documento | Luciano Beneventi |
| 21/06/2025 | V.6 | Incluir información sobre el prototipo con sus respectivos mockups, la evaluación heurística de Nielsen con su respectiva tabla y el control de versiones | Demesio Monsalves |
| 22/07/2025 | V.7 | Realizar el análisis y métricas de los resultados de la evaluación heurística del proyecto | Luciano Beneventi |

**Tabla de Contenidos**

[**1. INTRODUCCIÓN 3**](#_heading=h.uybu821ken12)

[1.1. Contexto del Problema 4](#_heading=h.fcdng6v2d3pi)

[1.2. Propósito 4](#_heading=h.t0px4f4p2qmh)

[1.3. Ámbito 4](#_heading=h.7fsrvs2slyw9)

[1.4. Definiciones, acrónimos y abreviaciones 4](#_heading=h.g03z6tegj5sx)

[1.5. Resumen ejecutivo 5](#_heading=h.5kq7ycxm5zi9)

[1.6. Arquitectura del sistema 6](#_heading=h.wly2gldjefs6)

[**2. VISIÓN DEL SISTEMA 7**](#_heading=h.4egerclc7ze4)

[2.1. Descripción general del sistema 7](#_heading=h.hci4rtgwyrqk)

[2.2. Objetivos del sistema 7](#_heading=h.bps8nrcf5qvo)

[2.3. Principales funcionalidades esperadas 7](#_heading=h.h3rmqamecv55)

[2.4. Supuestos y dependencias 7](#_heading=h.qgx2ufxdxtkk)

[**3. ESTILOS Y PATRONES ARQUITECTÓNICOS 8**](#_heading=h.g5hi62aop9gk)

[3.1. Estilo arquitectónico adoptado 8](#_heading=h.we1fldngva8c)

[3.2. Justificación del estilo según el contexto del sistema 8](#_heading=h.lnebaz8nbe6v)

[3.3. Patrones de diseño aplicados 8](#_heading=h.jia7unl6dk6k)

[**4. MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS 9**](#_heading=h.3bieieof3fdr)

[4.1. VISTA DE ESCENARIO 9](#_heading=h.4k593kxj59c)

[4.2. VISTA LÓGICA 13](#_heading=h.ra02acobdklf)

[4.3. VISTA DE IMPLEMENTACIÓN/DESARROLLO 15](#_heading=h.ve7nrf568ldu)

[4.4. VISTA DE PROCESOS 17](#_heading=h.hdqnmmctssfp)

[4.5. VISTA FÍSICA 19](#_heading=h.rsd0y0teh9ro)

[**5. REQUISITOS DE CALIDAD 20**](#_heading=h.828w7yhhuqxw)

[5.1. Propósito 20](#_heading=h.lwu0raxy4e9k)

[5.2. Atributos de calidad 20](#_heading=h.r5tz3lelylsp)

[*5.3. Reglas y criterios de evaluación de calidad 21*](#_heading=h.qwrobnijc7jm)

[**6. PRINCIPIOS DE DISEÑO APLICADOS 22**](#_heading=h.ok845remhcht)

[*6.1. Propósito 22*](#_heading=h.5ix6fxolghz9)

[*6.2. Diseño centrado en el usuario (UX/UI, prototipos, experiencia de usuario) 23*](#_heading=h.tql07wlbwokg)

[**7. PROTOTIPO 24**](#_heading=h.j20o6k9y7oax)

[7.1. Propósito: 24](#_heading=h.gngafkd7wdxp)

[7.2. Mockups (imágenes con una breve descripción) 24](#_heading=h.h0u08t9exjqp)

[7.3. Justificar herramientas de prototipado 32](#_heading=h.6edlh3rn5q0t)

[**8. EVALUACIÓN DE CALIDAD HEURÍSTICA DE NIELSEN 33**](#_heading=h.oc2znjorez57)

[8.1. Propósito: 33](#_heading=h.cr7r3rd8axxl)

[8.2. Lista de verificación 33](#_heading=h.dp5ojqnknrc8)

[8.3. Análisis y métricas de resultados 35](#_heading=h.pb6m59be0phb)

[**9. CONTROL DE VERSIONES 36**](#_heading=h.ulutdrvbvhk7)

[9.1. Propósito: 36](#_heading=h.f0q8a59knr9j)

[9.2. Control de versión utilizado 36](#_heading=h.3msa8j9ny1wu)

[9.3. Justificar herramientas de versionamiento 36](#_heading=h.yqcyf94xxkfg)

[**10. CONCLUSIONES 37**](#_heading=h.56pnerjz9w7t)

[**11. BIBLIOGRAFÍA 38**](#_heading=h.p1dfb2gyt3aa)

# INTRODUCCIÓN

## Contexto del Problema

La aduana está enfrentando graves problemas de retrasos para realizar los trámites aduaneros necesarios para que las personas puedan transitar, provocando retrasos de entre 8 a 20 horas de espera que han perjudicado la reputación de la aduana y causado una creciente incomodidad por parte de los viajeros.

## Propósito

El propósito nuestro durante este proyecto es modernizar los sistemas de las aduanas para así agilizar todos los procesos necesarios para el tránsito entre países con Argentina, de esa forma buscamos mejorar la experiencia de los usuarios y reducir los tiempos de espera.

## Ámbito

El sistema que se busca implementar tiene como objetivo agilizar los procesos que se realizan en la aduana, a fin reducir los tiempos que tardan dichos procesos, para lograr este objetivo el sistema debe de contar con las funcionalidades necesarias para automatizar la generación de los documentos que hacen falta para poder transitar por la aduana, como documentos para el tránsito de menores, alimentos, mascotas y vehículos. También debe implementar funcionalidades para poder generar informes que ayuden a las aduanas a tener una mejor visualización sobre la situación actual que transcurre en el lugar. Para esto se busca implementar un sitio WEB donde estén disponibles diferentes apartados que permitan el acceso tanto a personas comunes como a los mismos funcionarios de la aduana para poder realizar los documentos de manera más rápida.

## Definiciones, acrónimos y abreviaciones

| **ACRÓNIMO** | **DESCRIPCIÓN** |
| --- | --- |
| *MVC* | Modelo vista controlador, patrón de diseño que busca separar los componentes en estos tres apartados para hacer más cómodo el desarrollo. |
| *Aes-256* | Método de cifrado de datos |
| *WCAG* | Estándar de accesibilidad orientado a el contenido digital |
| *Lighthouse* | Herramienta que se usa para mejorar la calidad de páginas WEB |
| *DCU* | Programas y aplicaciones utilizados para evaluar las interfaces de usuario. |

## Resumen ejecutivo

Ante la problemática de atrasos en la realización de los trámites necesarios para poder transitar por la aduana Chile-Argentina, que ha repercutido en una peor experiencia para los viajeros y una mala percepción sobre el paso por la aduana.

Solución Propuesta:

Desarrollar un sistema de tránsito de aduana que permita agilizar los procesos necesarios para este.

* Automatización de generación de los documentos necesarios para el tránsito por aduana.
* Interoperabilidad con otros sistemas aduaneros limítrofes y con otros agentes que interactúan con el trámite de tránsito de aduanas.
* Interfaz intuitiva que busque un fácil acceso a los usuarios a los trámites que requieren.
* Generación de informes con información para la aduana sobre el tránsito que ocurre ahí.
* Implementación de medidas de seguridad y confidencialidad sobre los datos personales de los involucrados.

Beneficios esperado:

* Agilizar los procesos para el transporte por aduanas, reduciendo los tiempos de espera.
* Mejorar la experiencia de usuario al no exponerlos a tediosos trámites.
* Mejorar la funcionalidad de la aduana, implementando funcionalidades que otorgan mayor información sobre el tránsito que ocurre y reduciendo la carga a los funcionarios.

Alcance:

Este sistema debe trabajar dentro del paso “Los Libertadores” de manera inicial y debe incluir en los procesos a otras entidades relacionadas con los trámites para los documentos necesarios, como el SAG o PDI. Además que las funcionalidades no deben ser más que las necesarias para no complejizar el sistema.

## Arquitectura del sistema

Para el correcto desarrollo de este sistema hemos de realizar medidas que aseguren un entendimiento en todo el equipo antes de siquiera comenzar a desarrollar el código, esto a fin de minimizar problemas del entendimiento de las necesidades que deben cumplirse para que el software se alinea a lo que se busca solucionar. Para ello esquematizamos distintas partes que corresponden al sistema, como su funcionamiento esperado, distribución en el entorno y más a fin de que todo el equipo pueda estar bien a lo que se debe trabajar.

**Vista de escenario**

* Ilustrar las interacciones entre los actores y los distintos apartados del sistema. Esto mediante el uso de un diagrama de casos de uso.

**Vista Lógica**

* Describir las clases y componentes del sistema y cómo se relacionan e interactúan entre sí. Para ello desarrollaremos un diagrama de clases para ilustrar.

**Vista de Implementación/Desarrollo**

* Ejemplificar cómo será la organización del proyecto, mostrando su organización en módulos con un diagrama de componentes y su organización entre diferentes modelos y sus dependencias a través de un diagrama de paquetes.

**Vista de proceso**

* Donde se vislumbra la secuencia de acciones del sistema y el manejo del control, esto mediante un diagrama de actividades.

**Vista física**

* Donde se ilustra la infraestructura en la que se encontrará el software y como este se ejecuta en el entorno real, esto se muestra utilizando un diagrama de despliegue.

En resumidas cuentas, a través de esta arquitectura multivista se da una visión más completa sobre lo que se busca desarrollar y cómo esperamos que funcione, así asegurando la eficiencia y seguridad del proyecto.

# VISIÓN DEL SISTEMA

## Descripción general del sistema

Este sistema tiene como objetivo modernizar y digitalizar los procesos que actualmente se realizan en la aduana del paso fronterizo “Los Libertadores”. La idea es agilizar el tránsito de personas, vehículos, alimentos, mascotas y otros elementos que normalmente requieren distintos documentos. Para eso, se propone una plataforma web que permita tanto a los usuarios comunes como a los funcionarios generar y gestionar la documentación necesaria de forma rápida, segura y alineada con la normativa vigente. También se contempla que el sistema se conecte con servicios externos como el SAG o la PDI para validar automáticamente información relevante.

## Objetivos del sistema

* Reducir los tiempos de espera que actualmente afectan el paso por la aduana.
* Automatizar la creación y validación de los documentos requeridos.
* Mejorar la experiencia de los usuarios con una interfaz clara, sencilla y accesible.
* Entregar herramientas a los funcionarios para facilitar su trabajo y tomar mejores decisiones -gracias a reportes generados por el sistema.
* Garantizar que los datos estén protegidos, cumpliendo con estándares de seguridad que aseguren su confidencialidad, integridad y disponibilidad.

## Principales funcionalidades esperadas

* Registro de usuarios y acceso seguro mediante autenticación.
* Ingreso, edición y validación de datos personales y de vehículos.
* Firma digital de declaraciones juradas.
* Generación automática de los acuerdos necesarios para cruzar la aduana.
* Posibilidad de exportar o imprimir los documentos generados.
* Consulta en tiempo real del estado de los trámites realizados.
* Herramientas para que los funcionarios puedan administrar pasos, validar información y gestionar aduanas.
* Integración con servicios externos como SAG y PDI para validar antecedentes o requisitos.
* Generación de reportes administrativos que apoyen la gestión aduanera.

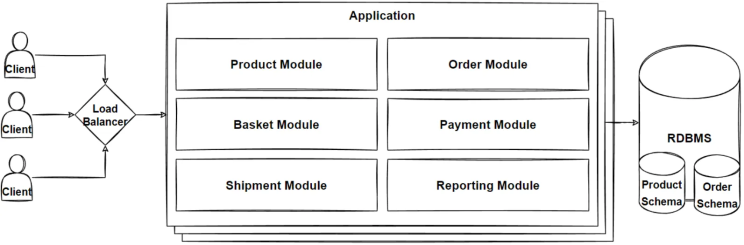
## Supuestos y dependencias

* Se espera que los usuarios tengan acceso a internet y un dispositivo compatible para usar la plataforma.
* El sistema asume que entidades como SAG y PDI tienen disponibles APIs u otros mecanismos para validar información.
* La primera versión del sistema será implementada únicamente en el paso fronterizo “Los Libertadores”.
* Es clave que las instituciones involucradas participen activamente en la validación, mantenimiento y soporte del sistema.
* Se necesita contar con una infraestructura segura y estable (servidores web, base de datos, soporte técnico) para que el sistema funcione correctamente.

# ESTILOS Y PATRONES ARQUITECTÓNICOS

## **Estilo arquitectónico adoptado**

El estilo arquitectónico elegido corresponde al sistema monolítico modular,donde se dividen las funciones en módulos separados pero aún se mantiene en una sola unidad, así cada módulo es responsable de cada una de las funcionalidades.



De esta forma se puede mantener una escalabilidad mejor que en el sistema monolítico clásico y a su vez de no tener que distribuir las funciones apartadas que complican el intercambio de información entre estos.

## Justificación del estilo según el contexto del sistema

Para un sistema de entrada de vehículos en una aduana, una arquitectura monolítica puede ser adecuada inicialmente por su relativa simplicidad en el desarrollo y despliegue. Dado que las funcionalidades principales (registro de vehículos, validación de documentos, etc.) están estrechamente relacionadas y pueden requerir un acceso frecuente a los mismos datos, un monolito simplifica la gestión de la base de datos y la comunicación interna entre los componentes. Esto puede facilitar un desarrollo más rápido en las etapas iniciales del proyecto.

## Patrones de diseño aplicados

El patrón de diseño que decidimos implementar para este sistema es el MVC, pues permite una división clara entre las funcionalidades y la vista, además de la familiarización que cuenta el equipo con este tipo de patrón.

# MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS

## VISTA DE ESCENARIO

**Propósito**

El propósito de esta vista radica en dar un primer vistazo a como se espera que los diferentes actores interactúen con el sistema y las tareas que se busca que el sistema realice. De esta forma esta vista sirve como punto de partida para poder realizar el resto del diseño buscando que las tareas establecidas se puedan realizar correctamente. Este caso sirve para mostrar las tareas que se pueden realizar en este sistema de aduanas y quienes pueden hacerlo.

**Actores**

**-Visitante**: Actor que crea una cuenta nueva, no puede interactuar más allá de ver las opciones pero no puede realizarlas.

**-Persona**: Actor con cuenta ya creada que puede interactuar con el sistema y usar sus funciones.

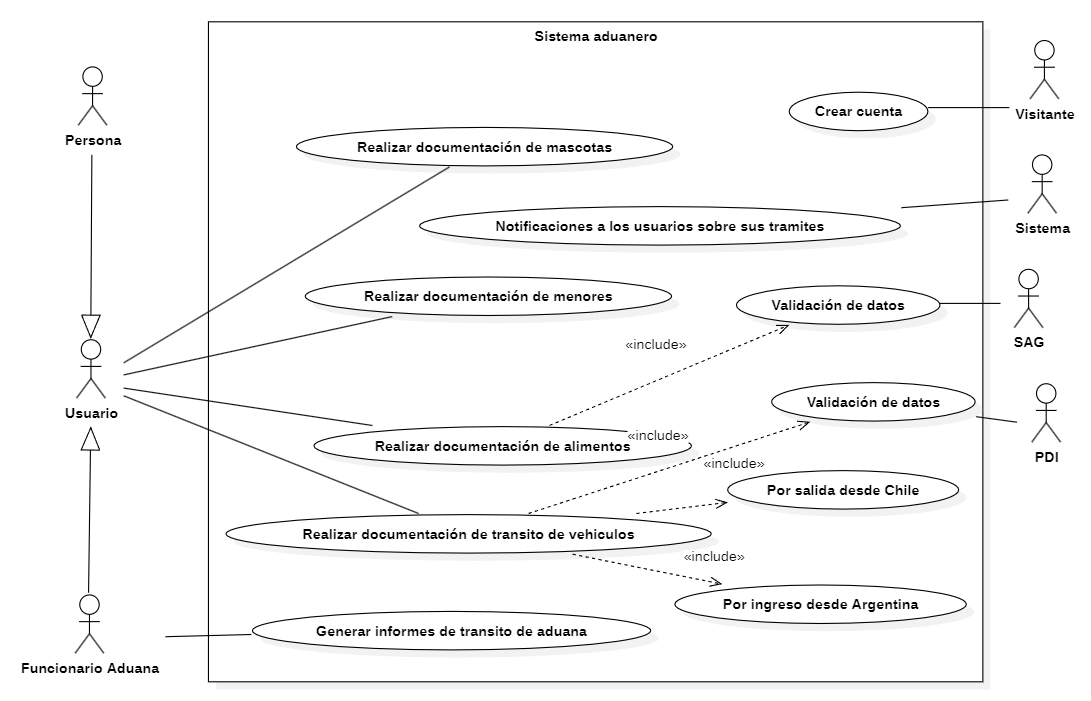
**-Usuario**: Actor que es quien puede realizar e interactuar con el sistema sus funciones, pero presta sus privilegios a otros actores que se busca que puedan realizar dichas acciones.

**-Funcionario Aduana**: Actor que corresponde al trabajador de la aduana, además de poder rellenar los documentos al igual que la Persona, cuenta con responsabilidades propias.

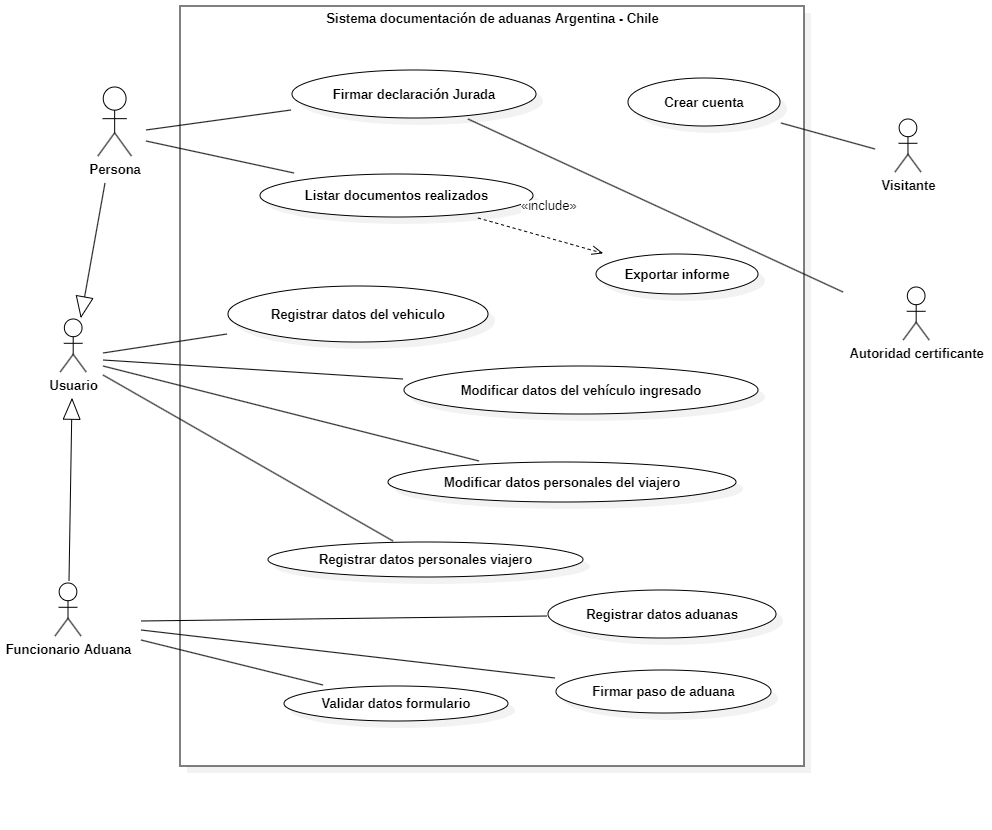
**-PDI**: Actor externo que interactúa con el sistema para validar que los datos usados no están relacionados a un crimen registrado.

**-SAG**: Actor que valida que los alimentos registrados no incumplan las normativas establecidas sobre el transporte de alimentos fuera del país.

**-Sistema**: El sistema mismo que se encarga de mandar notificaciones a las personas sobre el estado actual de sus trámites de forma automática.

**Diagrama general de casos de uso:**

**Diagrama de casos de uso específicos:**



**Lista de casos de uso**

| **Código** | **Nombre** | **Actores** |
| --- | --- | --- |
| CU-001-001 | Crear cuenta | Visitante |
| CU-002-001 | Registrar datos de vehiculo | Persona/Funcionario |
| CU-002-002 | Modificar dato de vehículo | Persona/Funcionario |
| CU-003-001 | Registrar datos de la persona | Persona/Funcionario |
| CU-003-002 | Modificar datos de la persona | Persona/Funcionario |
| CU-004-001 | Firmar declaración jurada | Persona/Autorizador/Autoridad |
| CU-005-001 | Listar documentos | Persona |
| CU-005-002 | Exportar informes realizados | Persona |
| CU-006-001 | Registrar la aduana | Funcionario |
| CU-007-001 | Firmar paso de Aduana | Funcionario |
| CU-008-001 | Validar datos del informe | Funcionario |

**Especificación de casos de uso:**

| **Caso de Uso** | Registrar datos del vehículo | **Identificador:** CU-002-001 |
| --- | --- | --- |
| **Actores** | Persona, Funcionario de aduana | |
| **Tipo** | Primario | |
| **Referencias** | R.F 2: Ingreso de información y validación de vehículos. | |
| **Precondición** | La persona debe estar registrada en el sistema para poder ingresar los datos de su vehículo. | |
| **Postcondición** | El vehículo queda registrado en el acuerdo. | |
| **Descripción** | La persona o el funcionario al realizar un acuerdo en línea llega a la sección de ingreso de vehículos, ingresa los datos de este, como marca, patente, tipo de vehículo y más, tras registrarlo se queda ingresado en el acuerdo. | |
| **Resumen** | El sistema permite ingresar los datos del vehículo con el que se realizará el viaje. | |

**CURSO NORMAL**

| **Nro.** | **Ejecutor** | **Paso o Actividad** |
| --- | --- | --- |
| N°1 | Persona o Funcionario de Aduana | Ingresar a crear un acuerdo de transporte |
| N°2 | Persona o Funcionario de Aduana | Ingresar el tipo de vehículo con el que se transporta. |
| N°3 | Persona o Funcionario de Aduana | Ingresar el chasis/vin del vehículo |
| N°4 | Persona o Funcionario de Aduana | Ingresar la marca del vehículo |
| N°5 | Persona o Funcionario de Aduana | Ingresar la patente del vehículo |
| N°6 | Persona o Funcionario de Aduana | Asignar SI/NO si cuenta con un vehículo de arrastre. |
| Comenzar a crear acuerdo, ingresar tipo de vehículo, ingresar chasis o vin, ingresar marca, ingresar patente y decir si cuenta con vehículo de arrastre. | | |

**CURSO ALTERNATIVO**

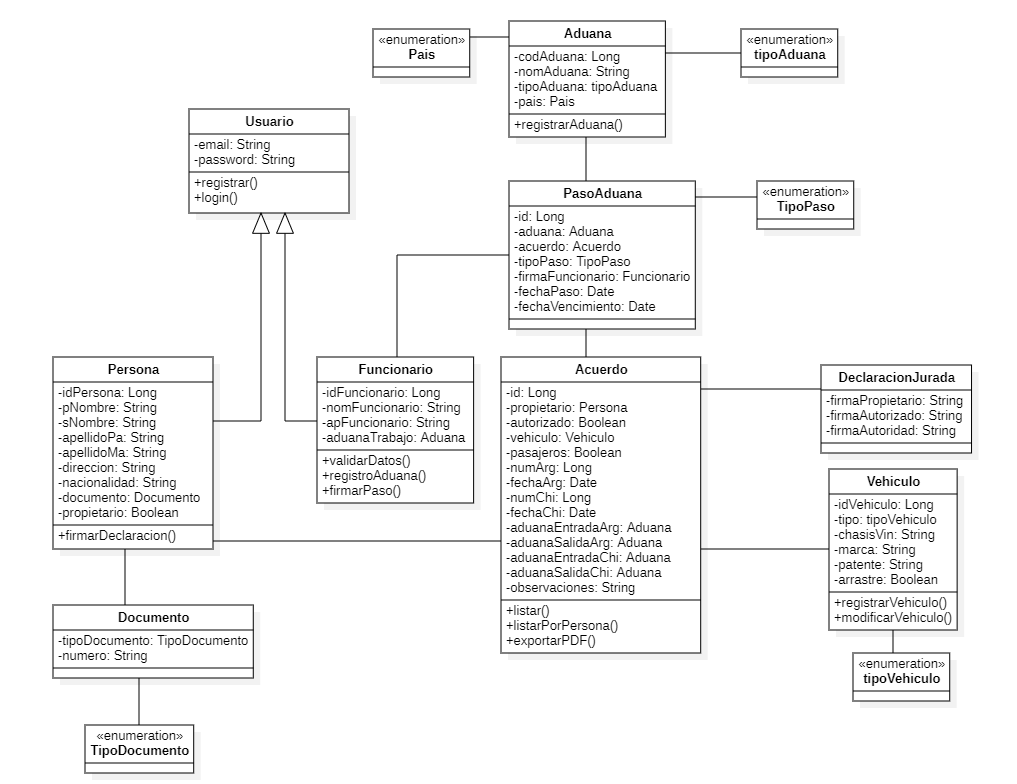
| **Nro.** | **Descripción de acciones alternas** |
| --- | --- |
| N°1 | CN °6: Si se dice que si hay un vehículo de arrastre, se debe hacer un segundo acuerdo por ese vehículo, |
| N°2 | CN °3, 4 y 5: Si no se rellenan estos puntos obligatorios, no se podrá generar el acuerdo. |
| N°1: Cuando se llena el apartado de vehículo de arrastre, si se selecciona NO el acuerdo seguirá con normalidad, si se selecciona SÍ será necesario un segundo acuerdo con dicho vehículo. N°2: Si se rellenan estos puntos obligatorios de forma correcta el acuerdo podrá generarse, en caso contrario el acuerdo será denegado inmediatamente | |

## VISTA LÓGICA

**Propósito:**

El propósito de la vista lógica es brindar una visión sobre cómo trabaja de manera lógica el sistema, implementando las clases, atributos, funciones y relaciones entre estas para representar cómo trabajan dentro de un sistema, esto a fin de facilitar el diseño y la futura implementación.

**Diagrama de clases**



**Descripción diagrama de clases**

Clases principales:  
  
**Usuario**: Clase genérica que cuenta con métodos y atributos compartidos por Persona y Funcionario.

**Persona**: Corresponde a los individuos que transitan por la aduana y cuenta con métodos necesarios para realizar el acuerdo a su vez que los datos personales necesarios para el acuerdo. Cuentan con una relación con la clase **Documento,** que corresponde a los documentos de identificación, como Cédula de identidad, D.N.I o Pasaporte.

**Funcionario**: Corresponde al trabajador de la aduana, quien es responsable de validar que los datos ingresados por la persona sean correctos a su vez de firmar cuando la persona pasa por una aduana.

**Aduana:** Corresponde a las aduanas registradas con su información correspondiente.

**PasoAduana:** Se encarga de registrar los tránsitos de personas con las respectivas aduanas en las que pasa, tanto de entrada como de salida y de ambos países.

**DeclaracionJurada:** Corresponde a la declaración jurada que es necesaria para el acuerdo, debe ser firmada por dos a tres personas, la persona que viaja, el dueño del vehículo en caso que este sea arrendado o prestado (Este campo no es obligatorio, solo si se cumple lo establecido) y una autoridad certificante.

**Vehículo:** Corresponde al vehículo con el que se realizará el viaje por la aduana, en él se registran los datos de dicho vehículo.

**Acuerdo:** La entidad principal del sistema, correspondiente al acuerdo que se debe realizar para poder realizar el viaje, en este se recopila toda la información de las otras clases vinculadas para generar un informe final.  
  
Enumeraciones:

**TipoDocumento**: Sirve para delimitar cuáles son los tipos de documentos permitidos(Cédula de identidad, Pasaporte y D.N.I).

**Pais**: Es para asignarle a la aduana a cual país corresponde(Argentina y Chile).

**TipoAduana**: Sirve para designar el tipo de aduana que corresponde el registro, si es una común o si es un punto de Resguardo/Avanzada.

**TipoPaso:** Sirve para asignar si el paso que realizó el viajero es de entrada o salida.

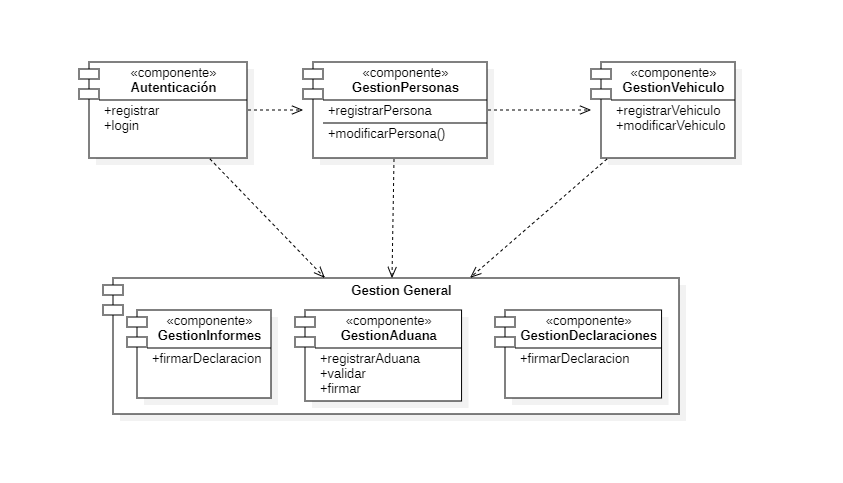
**TipoVehiculo:** Sirve para asignar al vehículo registrado a cual tipo de vehículo corresponde de una lista de los tipos permitidos de vehículo (Auto, Motocicleta, Station Wagon, Jeep, Motor Home, Camioneta y Otro)

## VISTA DE IMPLEMENTACIÓN/DESARROLLO

**Propósito**

En el contexto del modelo 4+1, la **vista de implementación/desarrollo** tiene como propósito general modelar la **organización estática del software desde la perspectiva del desarrollo**. Esto se logra a través de diagramas como el de Componentes, que ilustra la estructura del código en módulos y sus dependencias, y el de Paquetes, que organiza los elementos del modelo en grupos lógicos y muestra las dependencias entre ellos. En conjunto, estos diagramas facilitan la comprensión, gestión, construcción y evolución del software.

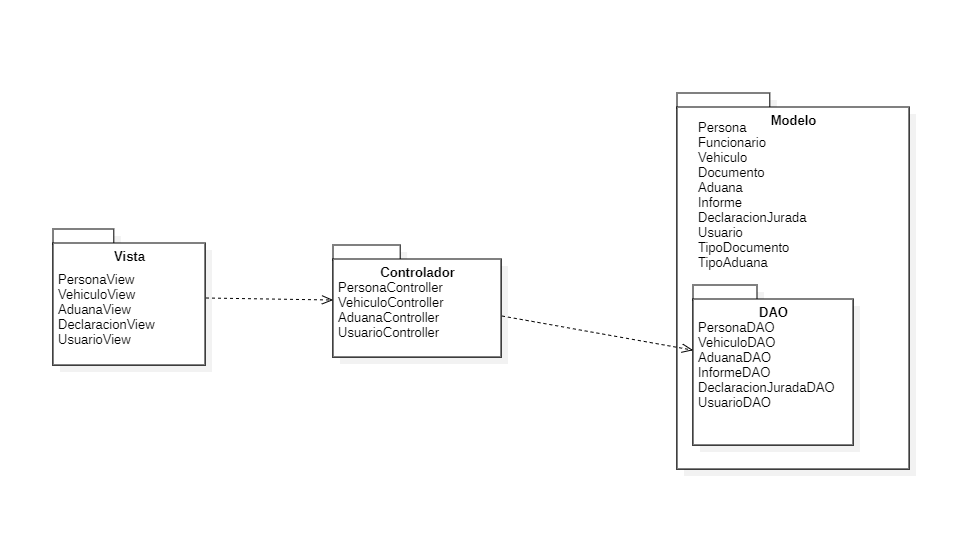
**Diagrama de componente**



**Descripción diagrama de componente**

El diagrama describe la arquitectura de un sistema mostrando sus componentes principales, las responsabilidades de cada uno (a través de las operaciones listadas) y las relaciones de dependencia entre ellos.

**Diagrama de paquete**



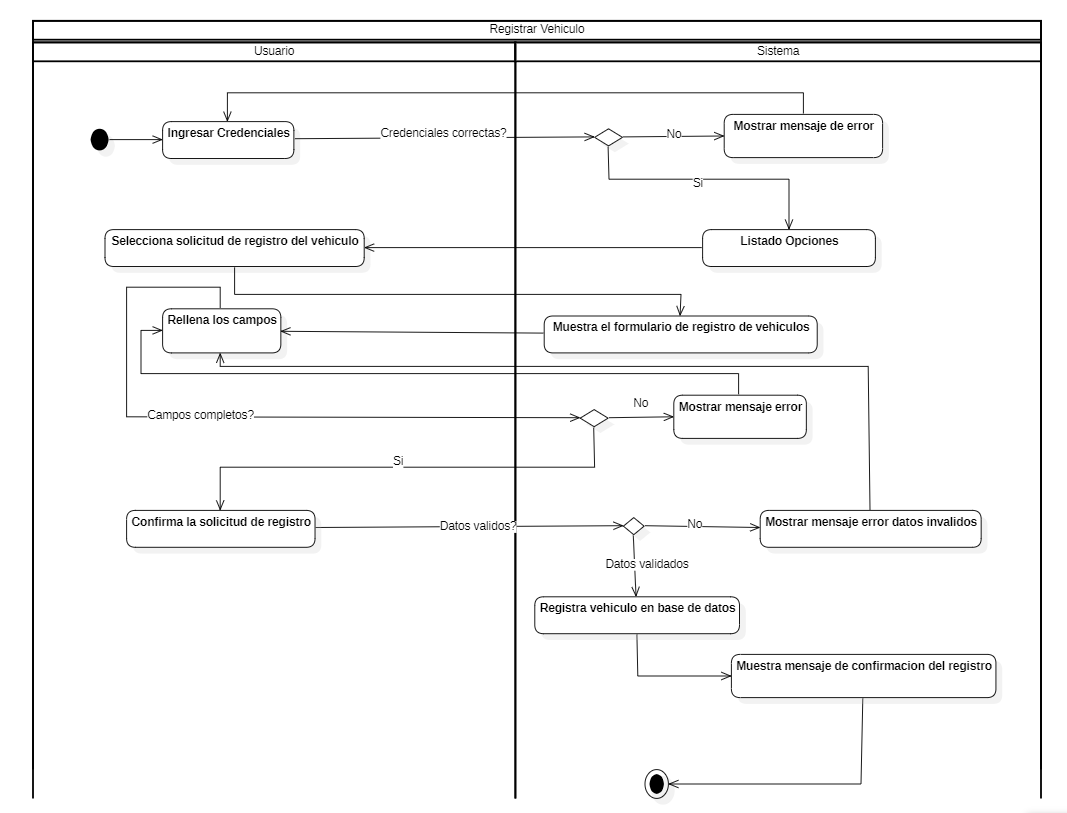
**Descripción diagrama de paquete**

Este diagrama representa una aplicación estructurada bajo el patrón MVC, separando responsabilidades entre la interfaz de usuario (Vista), la lógica de control (Controlador) y la lógica de datos (Modelo + DAO). Esto favorece la mantenibilidad, reutilización y prueba de la aplicación.

## VISTA DE PROCESOS

**Propósito**

La vista de procesos se enfoca en los aspectos dinámicos del sistema. En este caso orientado al diagrama de actividades su **principal función es describir el flujo de control** y **las secuencias de acciones dentro del sistema**, mostrando cómo se ejecutan las operaciones, cómo se coordinan las tareas y cómo se gestionan los eventos y estados en el tiempo.

**Diagrama de actividad**

**Descripción diagrama de actividad**

Este diagrama representa el flujo de actividades y la secuencia de acciones necesarias para que un **usuario registre un vehículo en el sistema de aduanas**. El proceso comienza cuando el usuario accede al sistema e ingresa sus credenciales. Si la autenticación es exitosa, el sistema despliega un listado con las opciones disponibles, entre las cuales el usuario selecciona la opción para registrar un vehículo.

A continuación, el sistema presenta un formulario que debe ser completado por el usuario. Luego, el sistema verifica que todos los campos hayan sido rellenados correctamente y que la información proporcionada sea válida.

Si los datos ingresados son correctos, el sistema procede a registrar el vehículo en la base de datos. En caso de que se detecten errores o información incompleta, se muestran mensajes de advertencia o corrección para que el usuario pueda subsanar las observaciones.

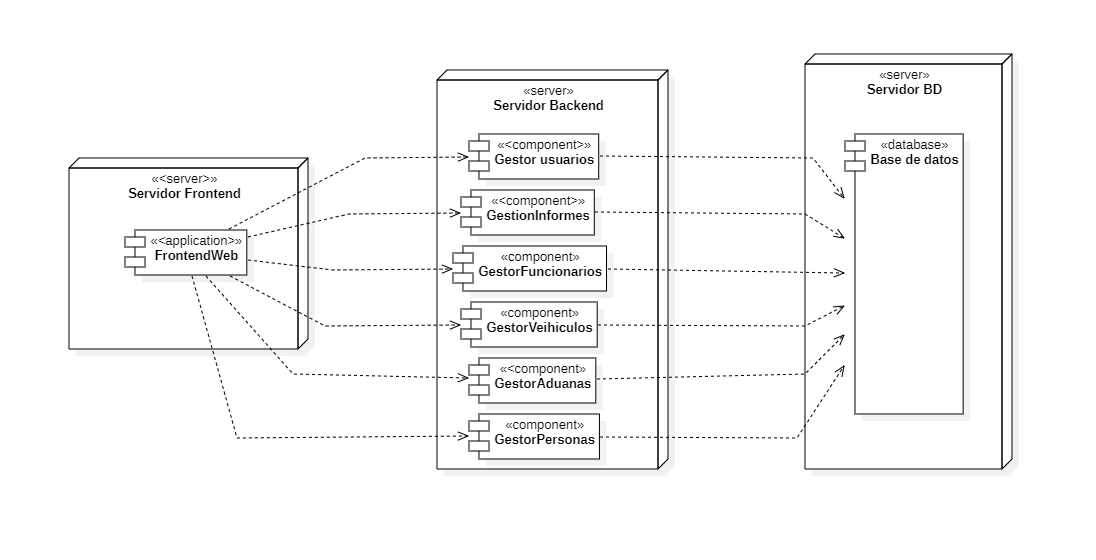
El proceso concluye exitosamente cuando el sistema muestra un mensaje de confirmación al usuario, indicando que el registro del vehículo ha sido completado, cerrando así esta actividad.

## VISTA FÍSICA

**Propósito**

En el contexto del modelo 4+1 la vista física o de despliegue tiene como principal propósito mostrar cómo los componentes de software se mapean y se distribuyen sobre el hardware físico del sistema en pocas palabras detalla la topología de la infraestructura el cómo se instala y ejecuta la aplicación en ese entorno.

**Diagrama de despliegue**



**Descripción diagrama de despliegue**

En el diagrama de despliegue se representa la arquitectura física del sistema aduanero, mostrando cómo se organizan sus partes y como se ejecuta el software. Tenemos **3 capas**: el **servidor frontend** donde aloja la aplicación web con la que el usuario final interactúa. El **servidor backend** el cual contiene la lógica de negocio, dividida en varios gestores( como usuarios, vehículos, aduanas) los cuales procesan las solicitudes del front end.Finalmente el **servidor BD** alberga la base de datos, donde se almacena toda la información del sistema.

# REQUISITOS DE CALIDAD

## Propósito

Los requisitos de calidad o requisitos no funcionales, tienen como propósito el definir las características y el comportamiento deseado de un sistema más allá de su funcionalidad básica, tiene como objetivo asegurar la satisfacción de un usuario, el guiar el diseño arquitectónico, proporcionar criterios medibles de evaluación y reducir riesgos identificando y abordando las preocupaciones al inicio del proyecto así evitando problemas mayores y costosos en etapas posteriores o ya con el sistema en producción.

## **Atributos de calidad**

| **ATRIBUTO DE CALIDAD** | **DESCRIPCIÓN** | **JUSTIFICACIÓN** |
| --- | --- | --- |
| Seguridad | Encriptar datos en la base de datos | Proteger los datos sensibles o información personal de los usuarios asegurando que permanezcan ilegibles para los atacantes |
| Seguridad | Autenticación multifactor | Añade una capa adicional de seguridad más allá de la contraseña lo que reduce considerablemente el riesgo de accesos no autorizados |
| Usabilidad | Facilitar la navegación entre funciones | Una interfaz bien organizada mejora la experiencia de usuario, reduce errores y permite que los usuarios completen tareas más rápido y con menor frustración. |
| Usabilidad | Proveer ayudas contextuales para los usuarios | Las ayudas contextuales, como mensajes informativos, iconos de ayuda o tutoriales interactivos, permiten a los usuarios entender mejor cómo usar el sistema sin necesidad de entrenamiento externo, facilitando la autonomía. |
| Usabilidad | Ser accesible para usuarios con discapacidades distintas | Diseñar con la accesibilidad en mente ( como el uso de lectores de pantalla, navegación por teclado, contraste de colores por ej) nos permite que el sistema sea utilizado por todos los usuarios así cumpliendo con principios de inclusión y normativas de accesibilidad web WCAG. |
| Disponibilidad | La capacidad del sistema para estar operativo y accesible para los usuarios cuando sea necesario estableciendo un tiempo de inactividad máxima mensual | En un sistema de aduanas, la disponibilidad es crítica. cualquier interrupción, por mínima que sea puede paralizar el flujo de bienes de personas en la frontera, generando significativas pérdidas económicas. |
| Confiabilidad | Recuperación ante fallos en un tiempo definido | Minimiza el impacto de interrupciones en un servicio como la aduana es esencial, permitiendo un restablecimiento rápido y evitando consecuencias graves, |
| Rendimiento | Rapidez con la que procesa solicitudes el sistema | Asegura agilidad en el control aduanero, evitando demoras, congestión y frustración. |

## Reglas y criterios de evaluación de calidad

| **Atributo de calidad evaluado** | **Criterio de evaluación** | **Herramientas o métodos** |
| --- | --- | --- |
| Seguridad | Datos sensibles encriptados con Aes-256 | Pruebas de penetración |
| Usabilidad | Facilidad de navegación con ayudas contextuales y tareas clave completadas con un 15% de reducción en tiempo con pruebas de usuario | Pruebas de usabilidad con usuarios, herramientas de auditoría de accesibilidad como Lighthouse. |
| Disponibilidad | 99.9% de tiempo de actividad en el mes medido como % de tiempo operativo | Monitoreo continuo de uptime como new relic, sistemas de alerta 24/7 |
| Rendimiento | Consultas/Solicitudes procesadas en menos de 5 segundos | Prueba de carga/estrés como con jmeter, monitoreo de aplicaciones para latencia |
| Confiabilidad | Recuperación ante fallos en máximo 90 minutos, en el 90% de los casos | pruebas de recuperación ante desastres(Simulaciones de fallos), automatización de despliegues |

# PRINCIPIOS DE DISEÑO APLICADOS

## Propósito

Tiene por objetivo mejorar la calidad del código al hacerlo más legible, mantenible, flexible y extensible. además fomenta la reusabilidad al estar facilitando que componentes o módulos puedan ser utilizados en diferentes partes del sistema o en otros proyectos. nos ayuda a reducir la complejidad organizando el código de manera lógica y coherente. aumenta la adaptabilidad permitiendo que el sistema sea más fácil de modificar y evolucionar ante cambios o nuevos requerimientos. previene problemas comunes al evitar errores de diseño que podrían llevar a sistemas rígidos y difíciles de manejar.

**Principios de diseño**

| **PRINCIPIO** | **DESCRIPCIÓN** | **APLICACIÓN EN EL SISTEMA** |
| --- | --- | --- |
| Cohesión | Cada módulo o clase tiene una única responsabilidad bien definida. | Los servicios están diseñados para realizar tareas específicas y no múltiples funciones |
| Abstracción | Simplificar la complejidad, enfocándose en lo esencial y ocultando detalles irrelevantes. | Se abstraen entidades como Usuario, Vehículo,Aduana, creando clases que representan sus propiedades y comportamientos claves sin exponer su implementación interna, ya que los usuarios interactúan sólo con la capa visible del frontend. |
| Acoplamiento | En pocas palabras el acoplamiento es el que tan dependientes son los módulos entre sí. | para promover un bajo acoplamiento en el sistema minimizando las dependencias directas entre módulos, usando interfaces y el patrón MVC para desacoplar vistas, controladores y modelos. |
| Descomposición y modularidad | Busca dividir el sistema en componentes lógicos más pequeños, independientes y manejables. | El sistema se estructurará de manera interna con módulos funcionales y responsabilidades bien definidas |
| Encapsulación | Ocultar detalles internos de la implementación de una clase o módulo. exponiendo solo interfaces. | Se aplica en clases como Usuario, vehículo, Aduana las cuales encapsulan sus atributos y métodos internos, exponiendo solo lo necesario a través de métodos públicos. protegiendo la integridad de los datos. |

## **Diseño centrado en el usuario** (UX/UI, prototipos, experiencia de usuario)

El desarrollo del sistema de aduanas se guiará por un enfoque de **diseño centrado en el usuario (DCU)** para asegurar que la solución final no solo sea funcionalmente robusta, sino que también intuitiva, eficiente y satisfactoria para todos sus actores. Para lograr esto se aplicaran los siguientes aspectos claves:

**Investigación y comprensión del usuario**: se identificarán las necesidades y puntos débiles de actores viajeros y funcionarios de aduana (Incluyendo la PDI, SAG) al realizar o procesar trámites, así guiando un diseño efectivo.

**Prototipado:** se crearán prototipos del sitio web para visualizar y probar flujos de interacción clave antes del desarrollo completo del sistema.

**Pruebas de usabilidad:** los prototipos serán probados con viajeros y funcionarios de la aduana para evaluar la facilidad en tareas como registrar los datos del vehículo y su respectiva interfaz de validación de los documentos.

**Iteración y mejora continua**: El feedback obtenido de las pruebas de usabilidad nos guiará a rediseños iterativos. El objetivo es optimizar la claridad y eficiencia de los procesos contribuyendo a la agilización general.

**Accesibilidad**: El diseño de la interfaz web se adherirá a las pautas WCAG para asegurar que el sistema sea usable por todas las personas, incluyendo aquellas con diversas capacidades.

**Experiencia de usuario UX:** Se priorizaron los flujos de trámite intuitivos, retroalimentación clara y oportuna al usuario(respecto al estado de su trámite) y un diseño visual que transmita eficiencia y seguridad, mejorando la percepción general del servicio aduanero.

# **7.** **PROTOTIPO**

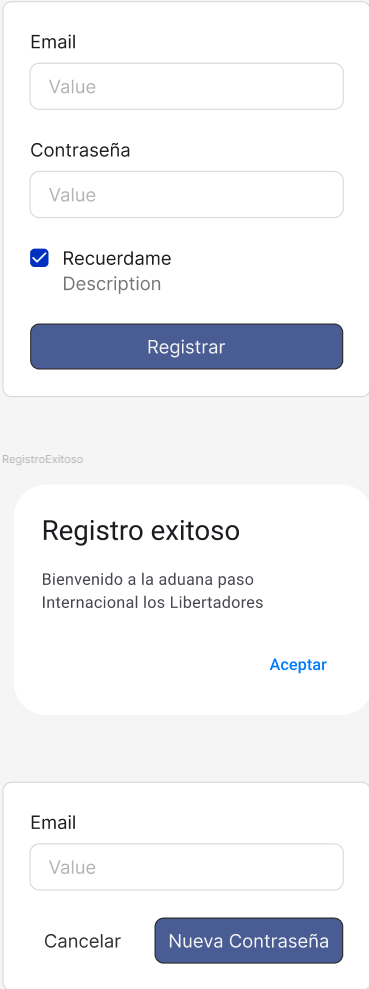
## 7.1. Propósito:

Este prototipo tiene como objetivo principal mostrar cómo se verá y funcionará el sitio web antes de desarrollarlo. Esto permite tener una idea clara del diseño, los colores, los botones, y la forma en que se moverá el usuario entre las distintas pantallas. Gracias a esto, es posible detectar errores o mejorar la experiencia antes de empezar a programar, lo que ahorra tiempo y recursos.

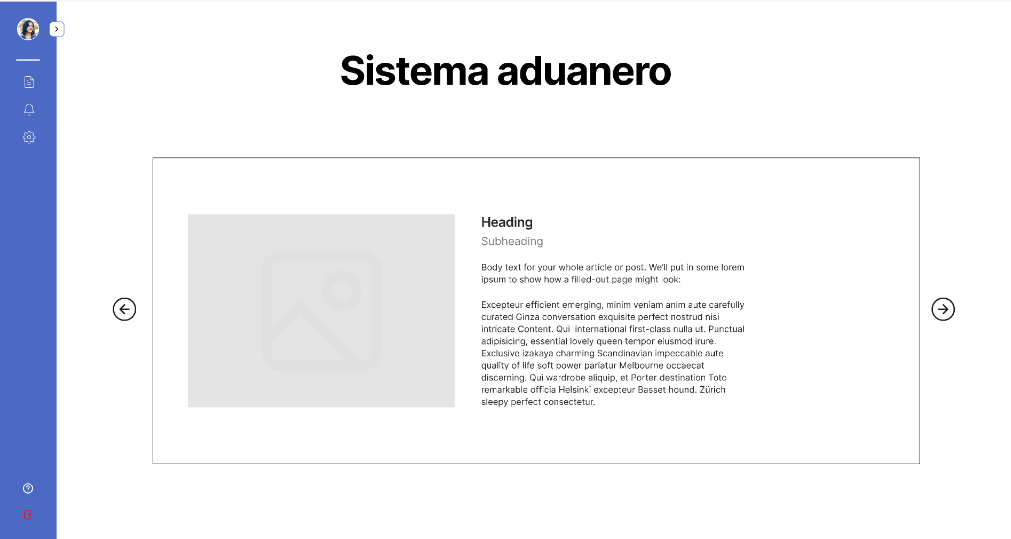
## 7.2. Mockups (imágenes con una breve descripción)



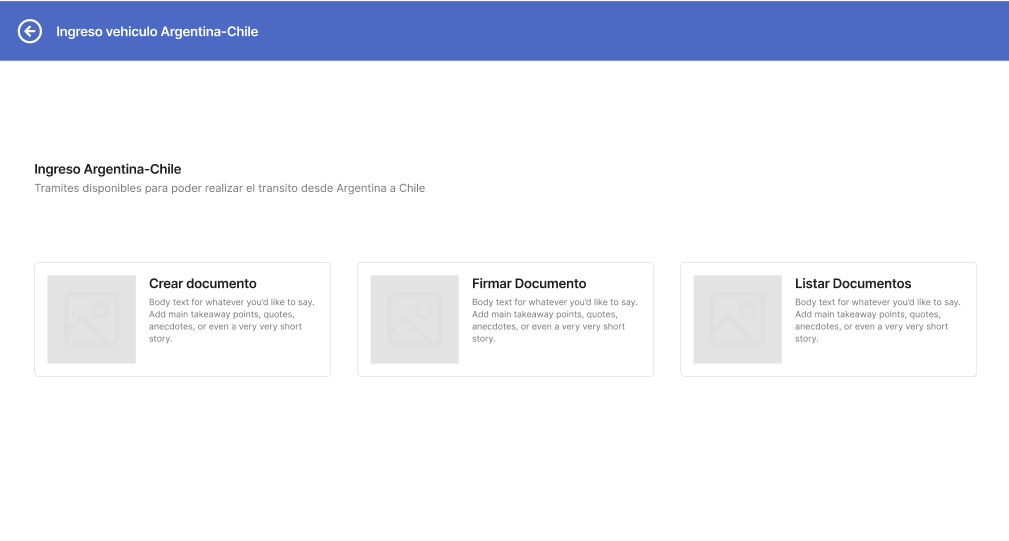
**Vista de login**: Donde vemos que esta la opcion de iniciar sesión o registrarse como nuevo usuario



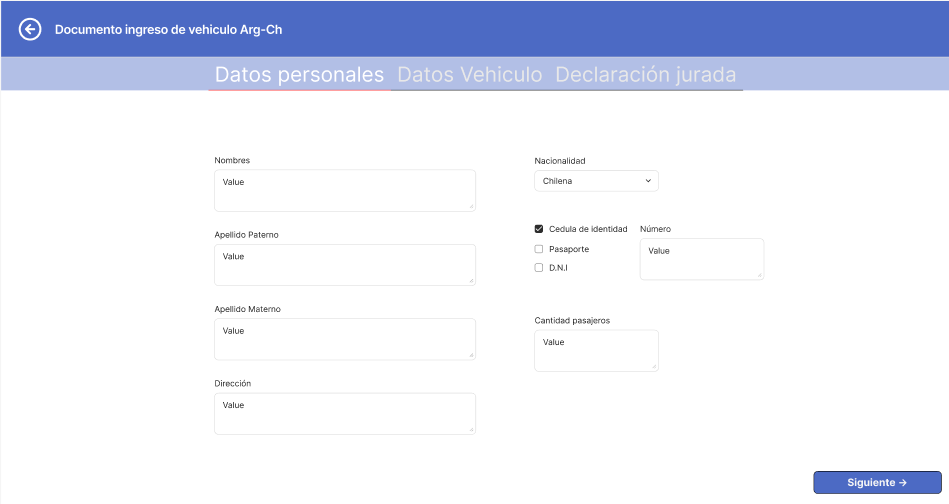
Vista donde se muestran las opciones de registrar con su respectivo registro exitoso y el de recuperar contraseña



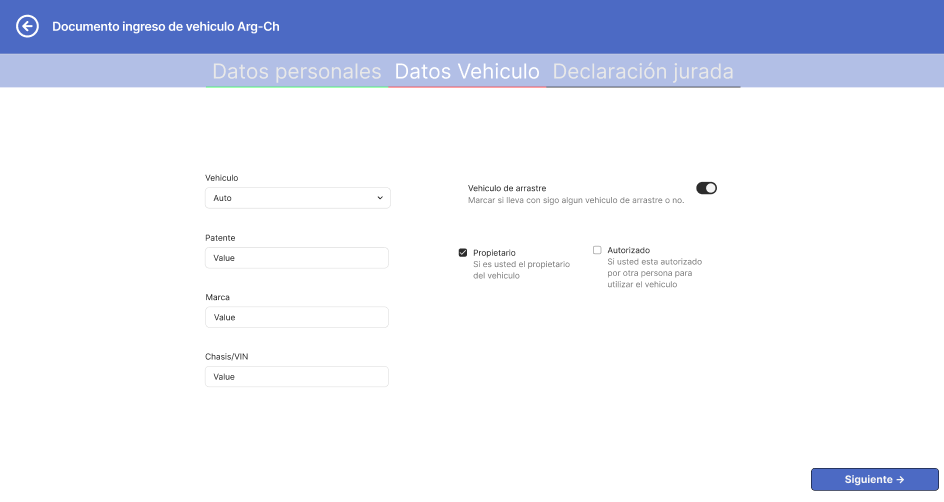
Vista principal del usuario donde podrá acceder a sus órdenes o notificaciones, mostrando noticias relacionadas o recomendaciones para el viaje.



Vista donde se podrá elegir que se desea realizar como usuario en la sección de ingreso de vehículos desde Argentina - Chile.



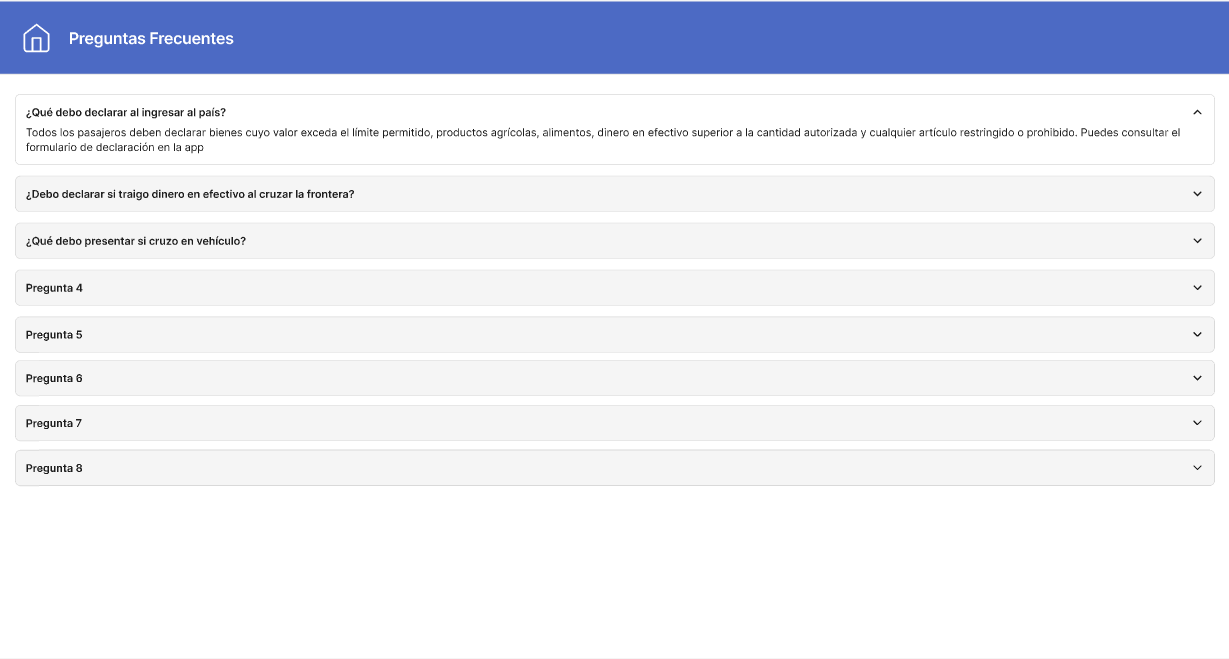
Vista donde se completaran los datos personales del viajero para crear el formulario.



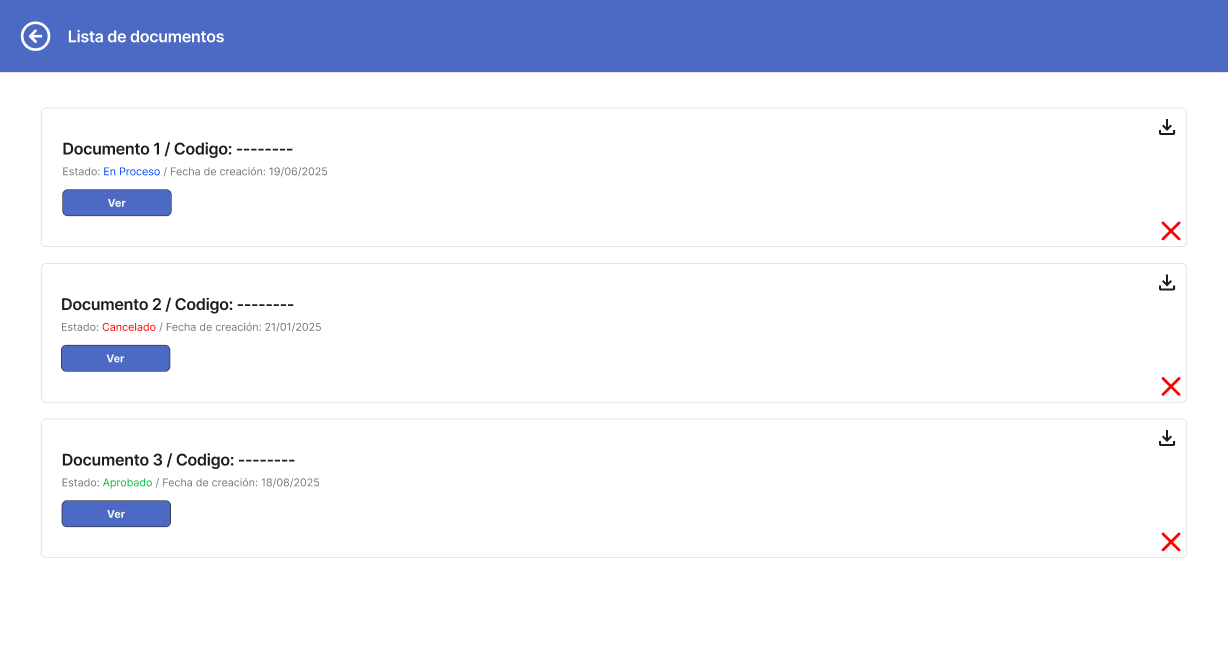
Vista donde se continúan completando los datos solicitados para el registro del vehículo



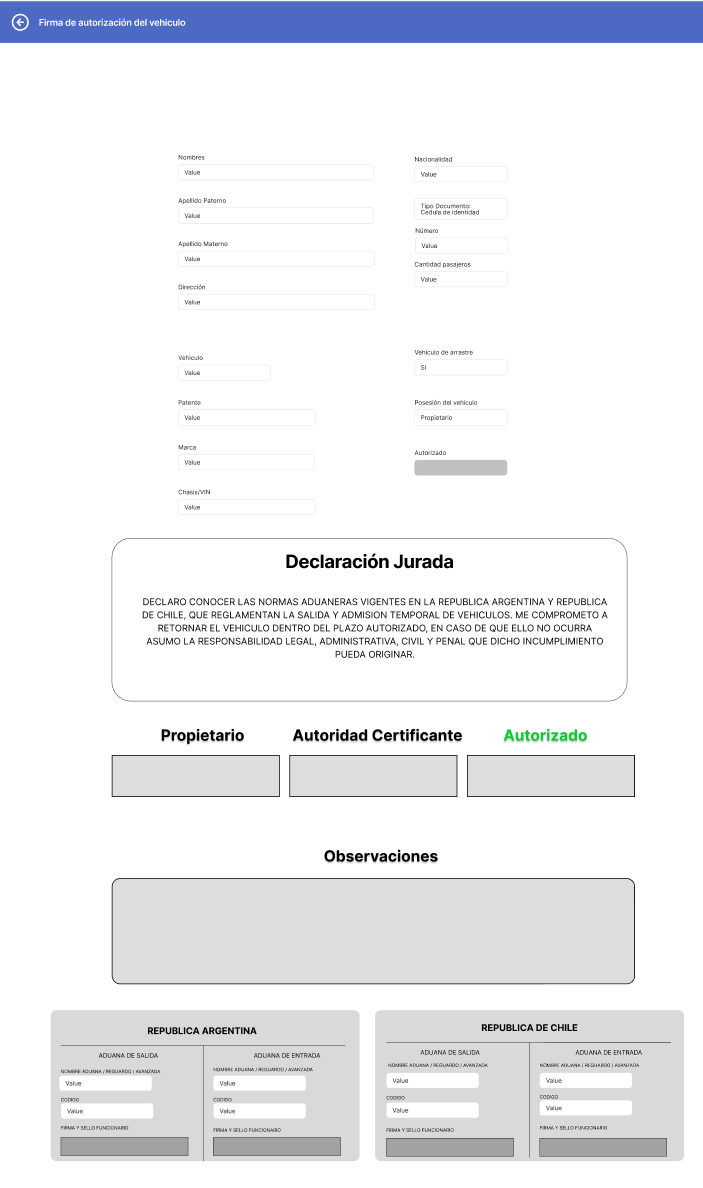
Vista para firmar la declaración jurada y así completar el formulario antes de enviarlo.



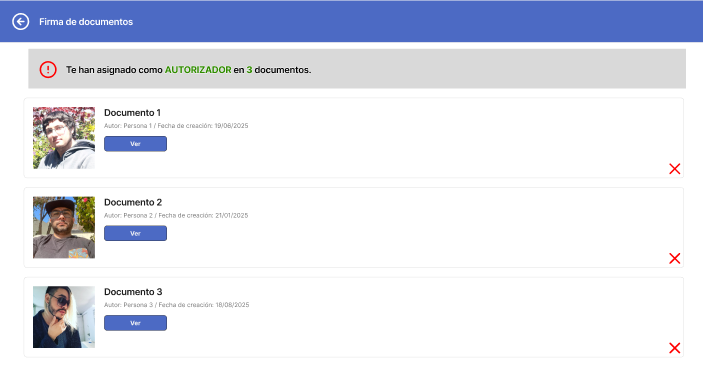
Vista donde se mostrarán las preguntas frecuentes y dudas de los usuarios, con sus respectivas respuestas.



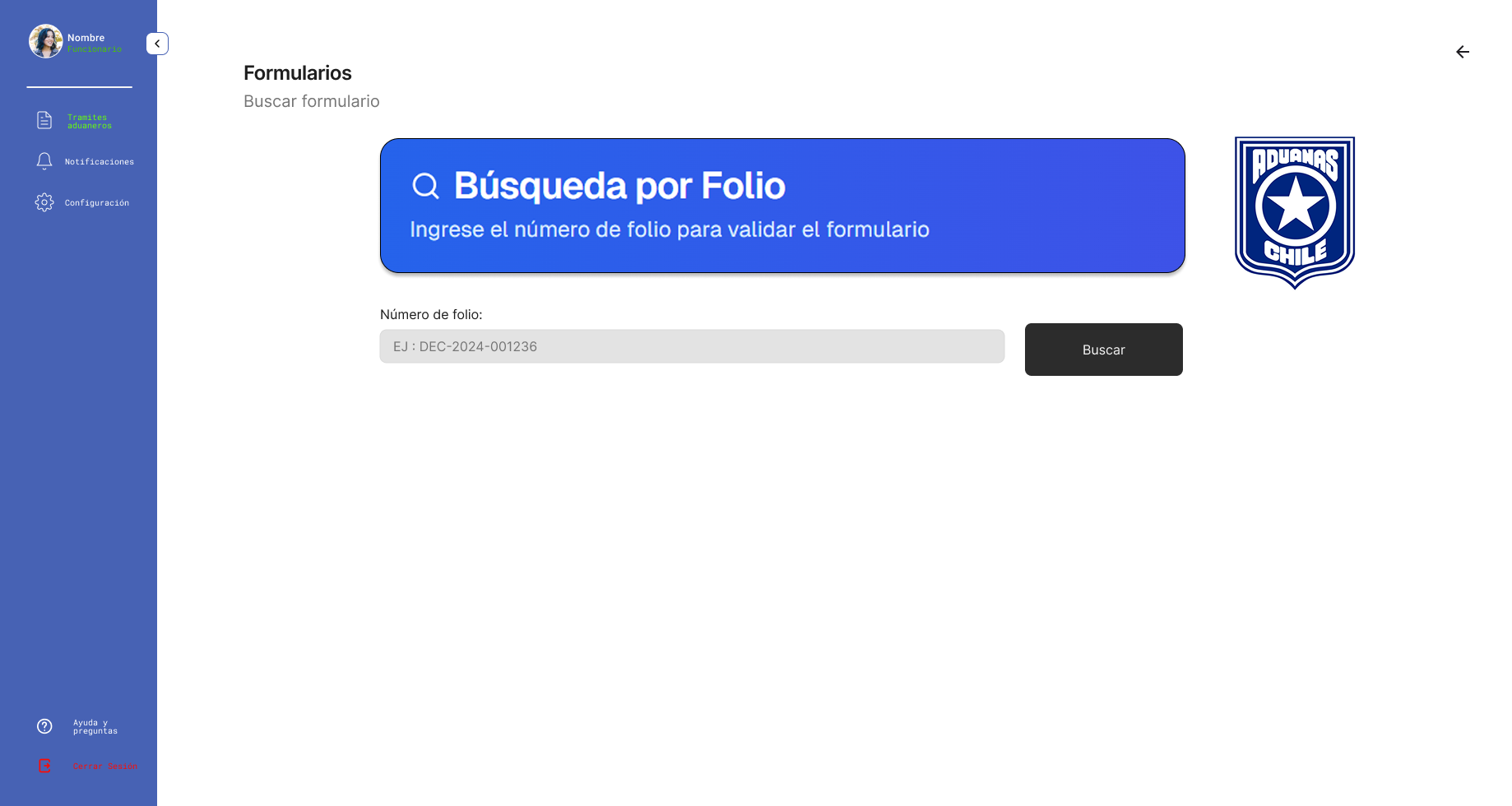
Vista de usuario donde se verán las órdenes enviadas por el, con opciones para cancelarla, descargarla (PDF), ver los detalles de esta y obtener el código que será usado en otras tareas.



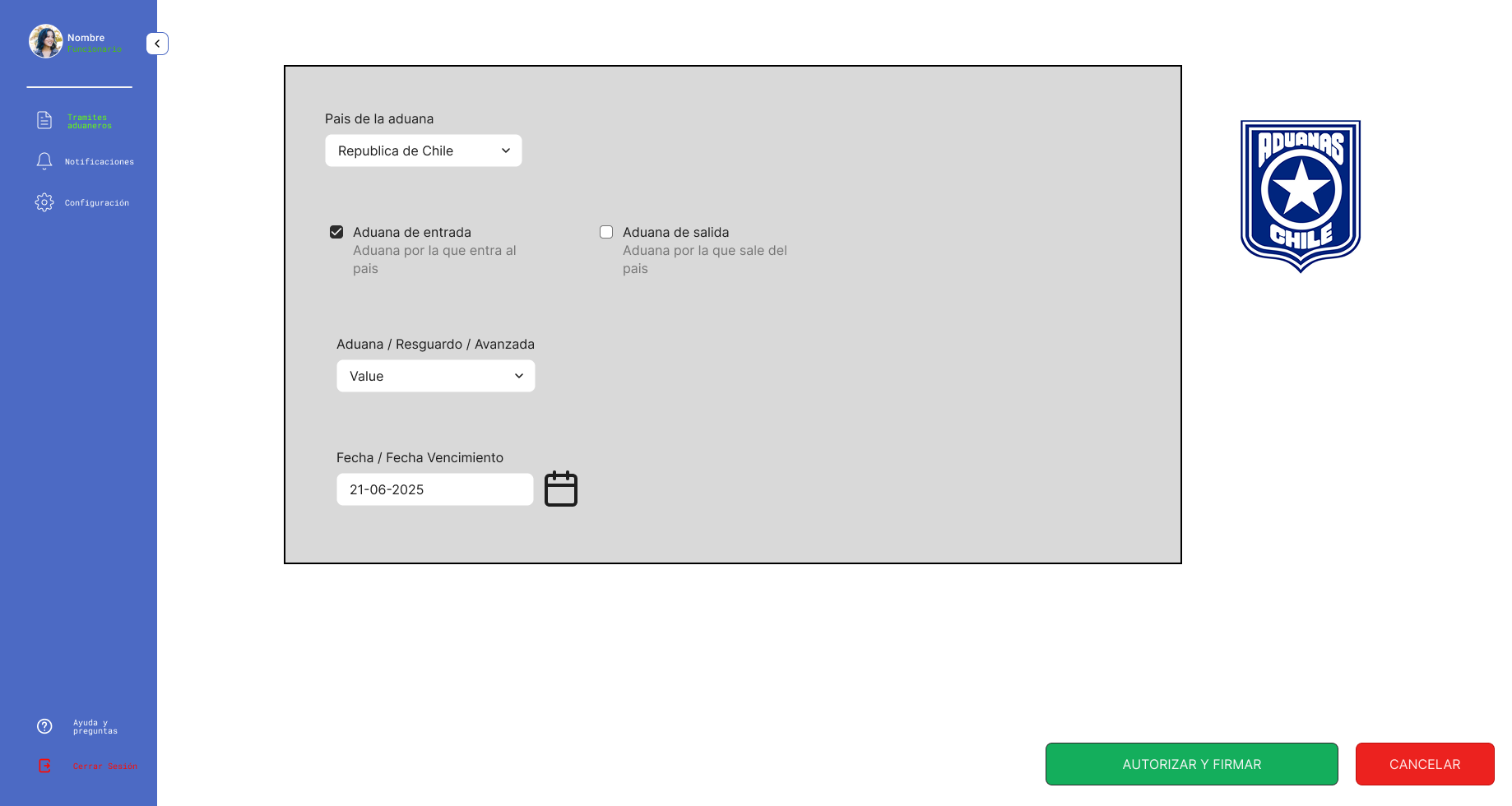
Vista resumen de un formulario realizado, mostrando los datos según el proceso en el que se encuentre.



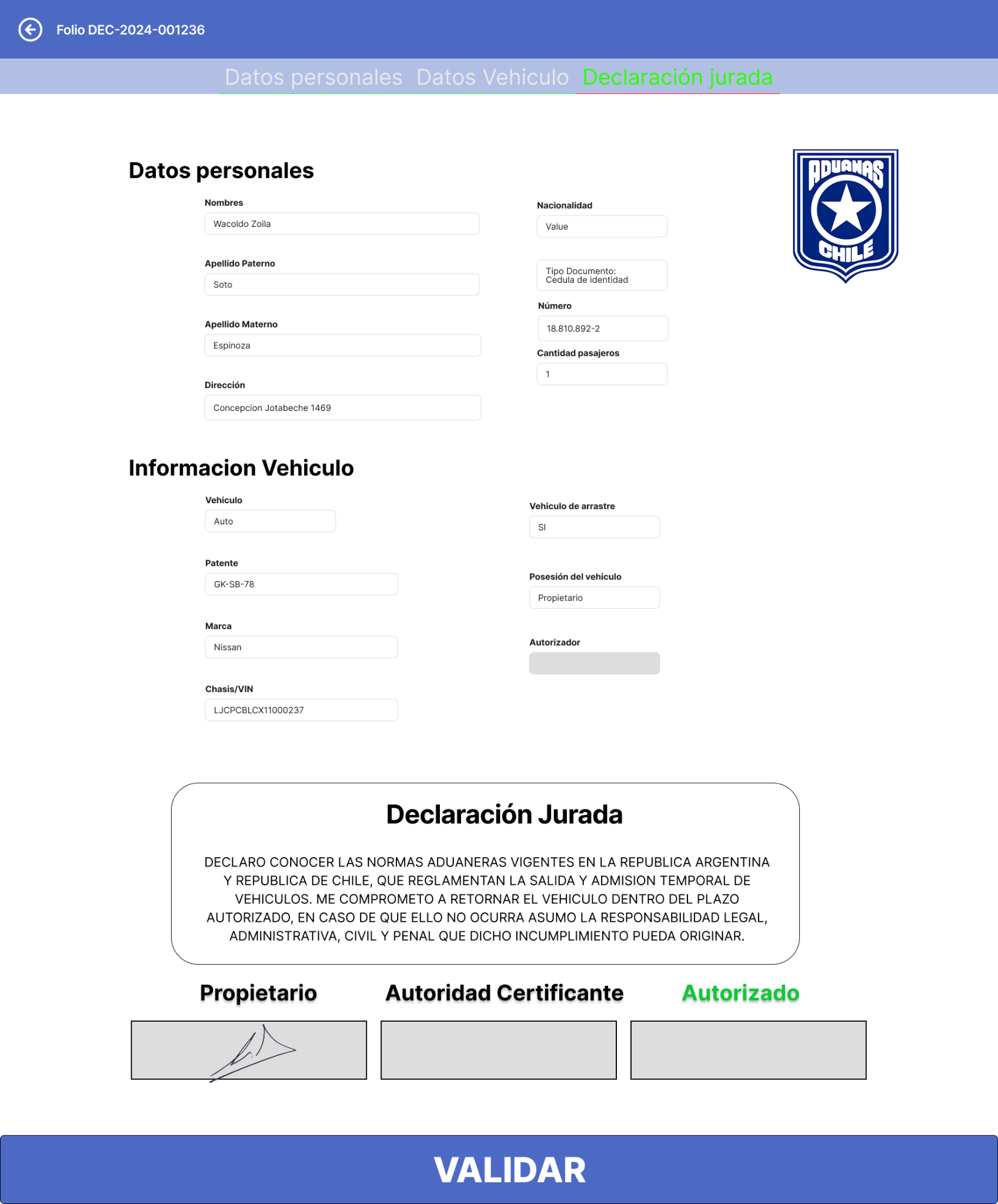
Vista de firmas externas, donde se mostrarán los formularios en los que el usuario fue puesto como propietario del vehículo y es necesaria su firma para validar este documento.



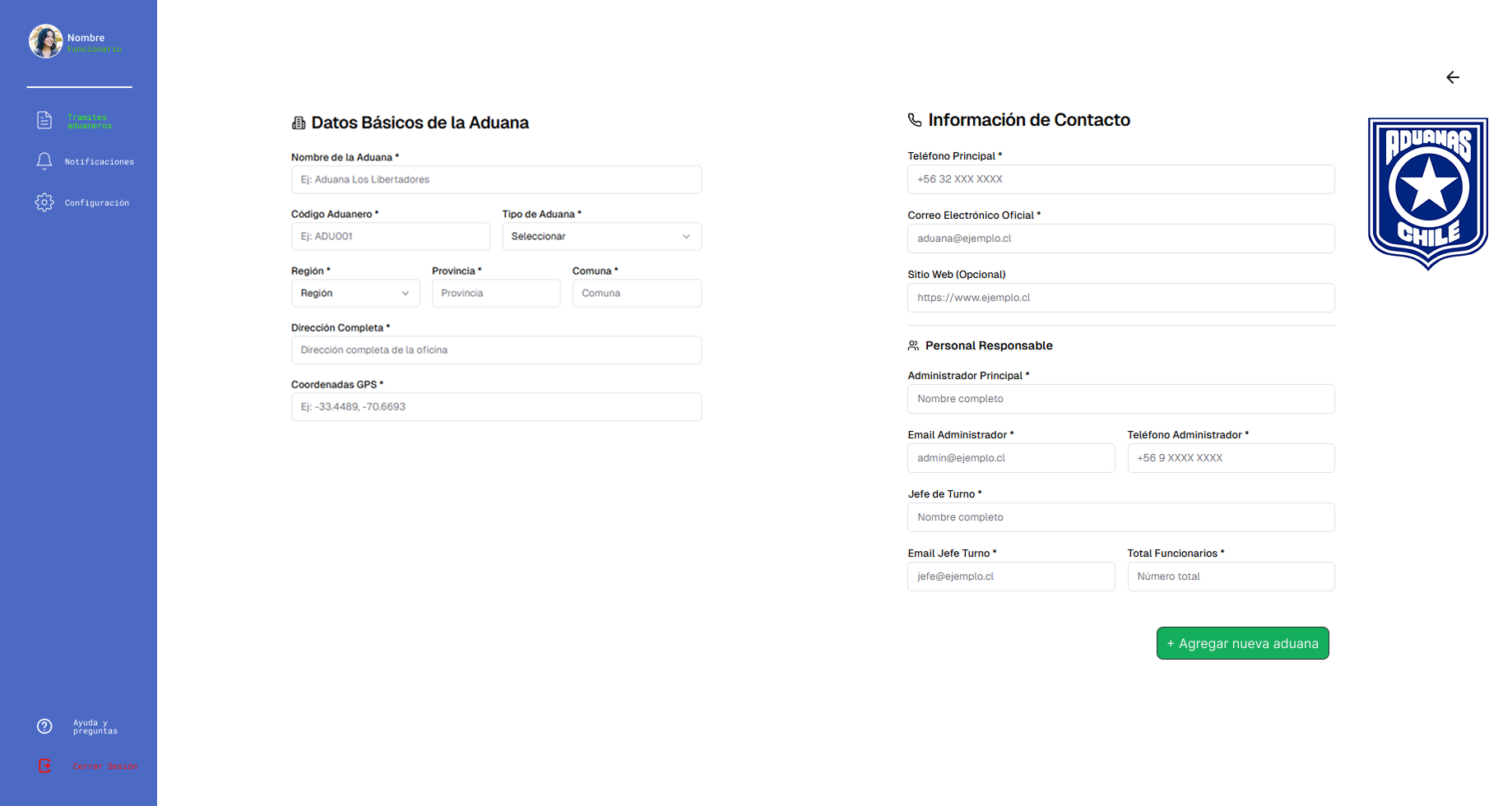
Vista de la sección donde el funcionario de aduana busca mediante el código del formulario este mismo para realizar las tareas sobre este.



Vista de la sección donde el funcionario registra el paso de un pasajero por una aduana, señalando el país, el movimiento que realiza el pasajero (Entrada o Salida), el tipo de aduana y la fecha de vencimiento o paso.



Validación realizada por el funcionario sobre el formulario que realizó el pasajero, se puede ver desde los datos personales del pasajero, los datos del vehículo y las firmas realizadas en la declaración jurada.



Vista del funcionario donde se pueden registrar los datos de las aduanas, de esta forma solo se usarán las registradas para realizar los registros de paso de los viajeros.

## 7.3. Justificar herramientas de prototipado

El uso de prototipado en Figma es una buena práctica porque permite visualizar y validar de forma anticipada el diseño y funcionamiento de una aplicación o sitio web, sin necesidad de escribir código. Esto facilita una mejor toma de decisiones, ya que tanto el equipo de desarrollo como los clientes o usuarios pueden ver y entender cómo será la experiencia final.

Además, al detectar errores o mejoras en etapas tempranas del proyecto, se evitan fallas costosas y se optimizan los recursos. También mejora la comunicación entre diseñadores, desarrolladores y otras áreas involucradas, asegurando que todos tengan una visión clara y compartida del producto.

Por estas razones, el prototipo no solo agiliza el proceso de diseño, sino que también aumenta la calidad del producto final y reduce el riesgo de fallas en el desarrollo.

# **8.** **EVALUACIÓN DE CALIDAD HEURÍSTICA DE NIELSEN**

## 8.1. Propósito:

El propósito de una evaluación heurística de Nielsen es **detectar problemas de usabilidad en una interfaz digital** usando un conjunto de 10 principios reconocidos. Esta evaluación permite mejorar la experiencia del usuario de forma rápida y efectiva, sin necesidad de pruebas complejas, ayudando a crear sistemas más claros, intuitivos y fáciles de usar.

## 8.2. Lista de verificación

| **Nº** | **Principio de Usabilidad de Nielsen** | **Criterio de Evaluación** | **¿Se cumple? (✔/✘)** | **Observaciones / Evidencia** | **Gravedad del problema** |
| --- | --- | --- | --- | --- | --- |
| 1 | **Visibilidad del estado del sistema** | ¿El sistema informa claramente al usuario de lo que está ocurriendo (cargas, acciones)? | ✔ | Cuando se ejecutan botones con acciones importantes se ejecutan cuadros de diálogos. | No aplica |
| 2 | **Correspondencia entre el sistema y el mundo real** | ¿La terminología y flujos se relacionan con el lenguaje y lógica del usuario? | ✔ | Se utilizan términos conocidos como  declaración. | No aplica |
| 3 | **Control y libertad del usuario** | ¿El usuario puede deshacer/repetir acciones fácilmente? | ✔ | El usuario puede regresar a ventanas anteriores y cancelar operaciones que esté ejecutando, como en el formulario o el menú. | No aplica |
| 4 | **Consistencia y estándares** | ¿Se mantiene un diseño coherente entre pantallas, botones y mensajes? | ✔ | Se utilizan botones y menús coherentes entre sí, los cuales tienen una ubicación similar. | No aplica |
| 5 | **Prevención de errores** | ¿El diseño evita que ocurran errores antes de que sucedan? | ✔ | Se advierte antes de eliminar un formulario o al ejecutar alguna acción de relevancia. | No aplica |
| 6 | **Reconocimiento mejor que recuerdo** | ¿Las opciones y funciones son visibles sin que el usuario deba recordar información? | ✔ | Existen menús desplegables y botones completamente visibles al usuario. | No aplica |
| 7 | **Flexibilidad y eficiencia de uso** | ¿Permite atajos o personalización para usuarios avanzados? | X | Se debería integrar un modo oscuro, modo para daltónicos para lograr mayor comprensión, y evitar errores por parte del usuario | Medio |
| 8 | **Diseño estético y minimalista** | ¿La interfaz evita información innecesaria o ruido visual? | ✔ | Interfaz limpia, con lo necesario para los trámites; sin elementos decorativos distractores. | No aplica |
| 9 | **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores** | ¿Los mensajes de error son claros, comprensibles y ofrecen solución? | ✔ | Los Cuadros de diálogo ofrecen  descripciones | No aplica |
| 10 | **Ayuda y documentación** | ¿Existe ayuda accesible, clara y orientada a la tarea cuando el usuario lo necesita? | ✔ | Hay una sección para responder preguntas frecuentes del uso del sistema; la documentación está en un PDF externo poco accesible. | Medio |

## 8.3. Análisis y métricas de resultados

En base a lo visto en la evaluación heurística de Nielsen vista anteriormente, se obtuvieron los siguientes resultados:

* Criterios cumplidos: 9 de 10
* Criterios no cumplidos: 1 de 10
* Porcentaje de cumplimiento: 90%
* Porcentaje por cumplir: 10%

Observando estos resultados se puede ver que 9 de los criterios cumplen con los estándares actuales del desarrollo de software, evidenciando un cumplimiento consistente, que hace a este un proyecto que cumple con estándares de calidad actuales y dando valor al proyecto en sí, destaca que uno de estos punto de igual forma presenta una observación sobre algunas recomendaciones para su mejora, pero de todas formas el estado actual no compromete el funcionamiento del sistema y es útil en el estado actual en el que está.

Por otro lado se presenta un criterio que no se ha cumplido, “Flexibilidad y eficiencia de uso”, señalando que el sistema actual no ofrece mecanismos que permitan a los usuarios realizar atajos u opciones de personalización que hagan al sistema uno más flexible y eficiente. Este punto queda entonces como un punto a trabajar en futuras iteraciones del proyecto a fin de aumentar el nivel de calidad del proyecto y facilitar el uso de este para todo el publico que hará uso de el programa, esto mediante la implementación de formas de personalización del sistema, como un modo oscuro, designar apartados como favoritos y que estos se muestran más fácilmente, accesibilidad para distintos trastornos de la visión entre otras funcionalidades que vayan apareciendo con el paso de las conversaciones.

Para concluir, mediante la observación de todos estos resultados se puede establecer a este proyecto como uno que cumple en gran medida los estándares de calidad actual del software, presentando una alta usabilidad y denotando el punto flaco que se debe seguir trabajando para crear un sistema completo y de alta calidad.

# **9.** **CONTROL DE VERSIONES**

## 9.1. Propósito:

El propósito del control de versiones con Git es **gestionar y registrar los cambios realizados en el código fuente de un proyecto a lo largo del tiempo**. Esto permite tener un historial completo de modificaciones, facilitar el trabajo en equipo, y revertir cambios en caso de errores o conflictos.

Git permite que varios desarrolladores trabajen simultáneamente en diferentes partes del código sin sobrescribir el trabajo de los demás, gracias al uso de ramas (*branches*). Además, mejora la organización y la calidad del desarrollo al permitir probar nuevas funciones de forma aislada y luego integrarlas de manera controlada.

## 9.2. Control de versión utilizado

El control de versiones que utilizaremos será el semántico ya que proporcionará una mayor información sobre los cambios que ha tenido el software, para así ayudar a los desarrolladores y usuarios a comprender de una mejor manera en que versión está el software.

## 9.3. Justificar herramientas de versionamiento

El uso de herramientas de control de versiones es fundamental en el desarrollo de software moderno, ya que permiten mantener un registro detallado de todos los cambios realizados en un proyecto a lo largo del tiempo. Esto brinda una mayor seguridad y control sobre el trabajo, facilitando la identificación y corrección de errores, así como la recuperación de versiones anteriores en caso de fallos.

Además, estas herramientas mejoran significativamente la colaboración entre varios desarrolladores, permitiendo que cada integrante del equipo trabaje en distintas partes del código sin interferencias, gracias al uso de ramas (*branches*). Esto ayuda a organizar mejor el desarrollo, probar nuevas funcionalidades sin afectar la versión estable del proyecto y luego integrar los cambios de forma ordenada.

Link repositorio: https://github.com/Chiispyn/Ing-software

# 10. CONCLUSIONES

Con este documento buscamos dejar claro cómo debería funcionar el sistema que proponemos para modernizar el paso fronterizo “Los Libertadores”. Desde el principio el objetivo fue resolver el problema real de los tiempos de espera, que afectan tanto a las personas como a los funcionarios que trabajan ahí.

La solución no solo contempla la arquitectura y el diseño general del sistema, sino que también incluye elementos concretos como el prototipo del sitio, la evaluación de usabilidad y el control de versiones. Gracias al prototipo pudimos visualizar cómo sería la experiencia de los usuarios antes de programar, permitiendo ajustar detalles importantes. La evaluación heurística confirmó que el sistema cumple con la mayoría de los principios de usabilidad, dejando en evidencia solo un punto a mejorar en futuras versiones. Y el uso de control de versiones con Git y versionado semántico nos entrega orden y colaboración efectiva durante el desarrollo.

No se busca hacer algo exageradamente complejo. La idea es una solución puntual, que resuelva lo que se necesita sin agregar más de lo necesario. Si esto se implementa bien, no solo mejorará el flujo actual, sino que también puede escalar a otros pasos fronterizos. En resumen, tenemos una base sólida, funcional y con espacio para seguir mejorando.

# 

# 11. BIBLIOGRAFÍA

Para la elaboración de este informe consultamos y nos basamos en las siguientes fuentes clave:

* OpenAI. (2025). *Asistencia en redacción y apoyo técnico mediante ChatGPT*. Disponible en<https://chatgpt.com/>
* Bass, L., Clements, P. & Kazman, R. (2012). *Software Architecture in Practice* (3ª ed.). Addison-Wesley.
* Kruchten, P. (1995). Architectural Blueprints—The “4+1” View Model of Software Architecture. *IEEE Software*, 12(6), 42–50.
* Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
* Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
* IEEE. (1998). *IEEE Recommended Practice for Software Requirements Specifications* (IEEE Std 830-1998).
* Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann.
* W3C. (2018). *Web Content Accessibility Guidelines (WCAG) 2.1*. Recuperado de<https://www.w3.org/TR/WCAG21/>
* https://jarroba.com/modelo-41-vistas-de-kruchten-para-dummies/