**[ Proyecto de desarrollo**

**para gestión aduanera II ]**

**(DAS) Documento Arquitectura de Software**

**Versión 1.0**

**Identificación de Documento**

| **Identificación** |  |
| --- | --- |
| **Proyecto** |  |
| **Versión** |  |

| **Documento mantenido por** |  |
| --- | --- |
| **Fecha de última revisión** |  |
| **Fecha de próxima revisión** |  |

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

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 22/05/2025 | 1.0 | Redacción e implementación de todo el apartado 1. Introducción, 3 Estilos y patrones arquitectónicos, y todos los propósitos de la 4 Modelo 4+1 y vistas arquitectónicas, además de los actores de la Vista escenario. | Benjamín Neira |
| 24/05/2025 | 1.1 | Adición de los requisitos de calidad; propósito, atributos de calidad, reglas y criterios de evaluación de calidad, y principios de diseño aplicados; principios de diseño, diseño centrado en el usuario. | Benjamin Luengo |
| 25/05/2025 | 1.2 | Implementación del diagrama de despliegue en la vista física y su descripción | Octavio Chávez |
|  |  |  |  |

**Tabla de Contenidos**

[**1. INTRODUCCIÓN 4**](#_heading=h.976ip82az2ns)

[1.1. Contexto del Problema 4](#_heading=h.4cjmgpw4a5mo)

[1.2. Propósito 4](#_heading=h.m10yhnlcnnww)

[1.3. Ámbito 5](#_heading=h.cezoy9j3w50i)

[1.4. Definiciones, acrónimos y abreviaciones 5](#_heading=h.nu84xyv9ar1j)

[1.5. Resumen ejecutivo 5](#_heading=h.bqm9ds580rqi)

[1.6. Arquitectura del sistema 6](#_heading=h.ulunc9he166u)

[**2. VISIÓN DEL SISTEMA 7**](#_heading=h.osb7xdhbcv9n)

[2.1 Descripción general del sistema 7](#_heading=h.c8rt5cr0nadq)

[2.2 Objetivos del sistema 7](#_heading=h.2curxhfskho4)

[2.3 Principales funcionalidades esperadas 8](#_heading=h.gxqje3t0lsk0)

[2.4 Supuestos y dependencias 8](#_heading=h.3bf84nlwiibt)

[**3. ESTILOS Y PATRONES ARQUITECTÓNICOS 8**](#_heading=h.iu7uwzyxfaw3)

[3.1. Estilo arquitectónico adoptado 8](#_heading=h.r4rui3dblyde)

[3.2. Justificación del estilo según el contexto del sistema 9](#_heading=h.9wzl12bik6b4)

[3.3. Patrones de diseño aplicados 9](#_heading=h.wsf6oeyp6uzn)

[**4. MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS 10**](#_heading=h.yo7pbgnihdrp)

[4.1. VISTA DE ESCENARIO 10](#_heading=h.h61lzgvlozdm)

[4.1.1. Propósito 10](#_heading=h.r6delly8ppnq)

[4.1.2. Actores 10](#_heading=h.mll0piptl9vo)

[4.1.3. Diagrama general de casos de uso 11](#_heading=h.qqavobl4v328)

[4.1.4. Diagrama de casos de uso específicos 11](#_heading=h.sl40acs2juuh)

[4.1.5. Lista de casos de uso 12](#_heading=h.rgoqtoci45em)

[4.1.6. Especificación de casos de uso 13](#_heading=h.zf1q5g7mmlcz)

[4.2. VISTA LÓGICA 14](#_heading=h.qw9xwfotr6h)

[4.2.1. Propósito 14](#_heading=h.ph8jikj38kvg)

[4.2.2. Diagrama de clases 14](#_heading=h.deuoxl1o6y8h)

[4.2.3. Descripción diagrama de clases 14](#_heading=h.bu3g98ti4z69)

[4.3. VISTA DE IMPLEMENTACIÓN/DESARROLLO 15](#_heading=h.jpzhpsbg436d)

[4.3.1. Propósito 15](#_heading=h.f3ng34t1xef7)

[4.3.2. Diagrama de componente 15](#_heading=h.16537hcbvvcv)

[4.3.3. Descripción diagrama de componente: 15](#_heading=h.2q3frf1y8eif)

[4.3.4. Diagrama de paquete 16](#_heading=h.k8maihqlf3ub)

[4.3.5. Descripción diagrama de paquete 16](#_heading=h.yfhutuhujz6i)

[4.4. VISTA DE PROCESOS 17](#_heading=h.37d0vqzhixp4)

[4.4.1. Propósito 17](#_heading=h.vjsbafu7w1g6)

[4.4.2. Diagrama de actividad 17](#_heading=h.tw7hsh3b7vyy)

[4.4.3. Descripción diagrama de actividad 18](#_heading=h.o7vncd8g3sj5)

[4.5. VISTA FÍSICA (salida vehículo o entrada vehículo) 19](#_heading=h.u1gbm3enezfg)

[4.5.1. Propósito 19](#_heading=h.stim2i5gt8at)

[4.5.2. Diagrama de despliegue 19](#_heading=h.4dvuviwsd2yk)

[4.5.3. Descripción diagrama de despliegue 20](#_heading=h.69dxcreozkz)

[**5. REQUISITOS DE CALIDAD 20**](#_heading=h.f8x7inflp1wf)

[5.1. Propósito 20](#_heading=h.da1n97u3ee57)

[5.2. Atributos de calidad 20](#_heading=h.fed3k3yrm8ly)

[5.3. Reglas y criterios de evaluación de calidad 21](#_heading=h.do8iq6dxmcfn)

[**6. PRINCIPIOS DE DISEÑO APLICADOS 21**](#_heading=h.j9311or2ies7)

[6.1. Propósito 21](#_heading=h.vd378b15jx07)

[6.2. Principios de diseño 21](#_heading=h.yzbm5m7y26b)

[6.3. Diseño centrado en el usuario (UX/UI, prototipos, experiencia de usuario) 22](#_heading=h.3td53wms2fgi)

[**7. PROTOTIPO 22**](#_heading=h.nuttqpazl2y)

[7.1. Propósito 22](#_heading=h.ruisd0ivudmm)

[7.2. Mockups (imágenes con una breve descripción) 22](#_heading=h.xq4mvgbhq2rb)

[7.3. Justificar herramientas de prototipado 22](#_heading=h.e9gjjuqcd4fb)

[**8. EVALUACIÓN DE CALIDAD HEURÍSTICA DE NIELSEN 22**](#_heading=h.4rsn2ggon4l8)

[8.1. Propósito 22](#_heading=h.pvwpazr6d9h)

[8.2. Lista de verificación 22](#_heading=h.hziiaay9jfnn)

[8.3. Análisis y métricas de resultados 22](#_heading=h.guxdvwk6olta)

[**9. CONTROL DE VERSIONES 23**](#_heading=h.bb32wpxzg522)

[9.1. Propósito 23](#_heading=h.o68cwafqwgl)

[9.2. Control de versión utilizado (justificar el tipo de control de versión utilizad (fecha, semántica o secuencial) 23](#_heading=h.ipd7j5o6dghh)

[9.3. Justificar herramientas de versionamiento 23](#_heading=h.z663wvvyuqff)

[**10. CONCLUSIONES 23**](#_heading=h.6ppldfhzdi3x)

[**11. BIBLIOGRAFÍA 23**](#_heading=h.ujnwo3ai0faw)

# INTRODUCCIÓN

## Contexto del Problema

El servicio nacional de aduanas cumple un rol fundamental en el control de fronteras y en la facilitación del comercio exterior. Sin embargo, a pesar de la gran importancia que tiene esta institución, su funcionamiento actual en los pasos fronterizos terrestres presenta serias limitaciones que afectan directamente la eficiencia del sistema y la experiencia de los usuarios.

Uno de los problemas más visibles ocurre en el paso fronterizo Los Libertadores, donde durante temporadas de alta demanda, como el verano, se han registrado tiempos de espera de más de 8 horas para cruzar desde, o hacia Argentina. Esta situación no solo es perjudicial para los ciudadanos y turistas, sino que también representa un cuello de botella para el comercio, el turismo y para el control migratorio. Las causas son muchas: la tramitación manual de documentos, la falta de interoperabilidad entre sistemas de aduanas, PDI y SAG, la escasa y pobre digitalización previa al cruce, y la limitada capacidad de infraestructura tecnológica para alta concurrencia.

En este contexto, se hace evidente la necesidad de modernizar y mejorar el sistema aduanero terrestre, a través de una solución tecnológica integral que permita digitalizar y automatizar los trámites aduaneros, optimizando los procesos fronterizos y mejorando la coordinación entre instituciones tanto del estado chileno, como con el estado argentino.

## Propósito

Este proyecto tiene como objetivo, diseñar y desarrollar un sistema de software enfocado en modernizar y optimizar los procesos del control aduanero terrestre en Chile, especialmente pensado para el paso Los Libertadores. La solución tecnológica que se propuso, busca resolver los principales problemas operativos actuales, mediante la implementación de una plataforma digital que facilite el registro, validación y emisión de documentación de aduanas, integrando además funcionalidades de interoperabilidad entre las distintas instituciones que participan, como aduanas, SAG y PDI.

El sistema proporcionará un entorno seguro, accesible y escalable, que permitirá tanto a los ciudadanos como a los funcionarios, realizar trámites previos al cruce fronterizo desde cualquier lugar con acceso a internet. De esta manera, se pretende reducir significativamente los tiempos de espera, especialmente en temporadas de alto flujo de vehículos, y al mismo tiempo tener un mejor mecanismo de fiscalización.

Además, el sistema facilitará la generación de reportes en tiempo real, mejorando la planificación y toma de decisiones por parte de las autoridades. Se incorporarán medidas robustas de seguridad, cumpliendo las diversas normativas chilenas sobre protección de datos personales, garantizando así la confianza tanto para los ciudadanos como para los funcionarios del estado.

## Ámbito

El sistema estará implementado y operará dentro del ámbito del paso fronterizo terrestre chileno Los Libertadores. Dentro de este contexto, se interactúa diariamente con múltiples instituciones del estado como: el servicio nacional de aduanas, que fiscalizar el paso de mercancías y vehículos; el SAG que controla el ingreso de productos de origen vegetal y animal; y la PDI que supervisa el flujo migratorio y verifica la documentación correspondiente.

La plataforma digital que se propone, funcionará como un sistema de gestión que conectará estas instituciones a través de un software digital común interoperable y coordinado. Permitirá a los ciudadanos realizar trámites de declaración, solicitud de permisos notariales para menores, o preinscripción de formularios de salida. Además, los funcionarios en frontera podrán verificar y registrar digitalmente el ingreso y la salida de personas y vehículos, reduciendo errores y tiempos a comparación del papeleo físico.

## Definiciones, acrónimos y abreviaciones

| **ACRONIMO** | **DESCRIPCION** |
| --- | --- |
| *SAG* | Servicio Agrícola y Ganadero. Encargado del control sanitario de productos agrícolas o de origen animal. |
| *PDI* | Policía de Investigaciones de Chile. Supervisa el flujo migratorio y verifica documentación personal. |
| *API* | Application Programming Interface. Conjunto de funciones que permiten la comunicación entre sistemas. |
| *MVC* | Modelo-Vista-Controlador. Patrón de diseño que separa la lógica de negocio, la presentación y el control. |

## Resumen ejecutivo

El presente documento de arquitectura de software, describe una propuesta tecnológica que está orientada a resolver las principales dificultades detectadas en el funcionamiento del paso fronterizo terrestre chileno Los Libertadores. El servicio nacional de Aduanas, enfrenta actualmente largas esperas, procesos manuales ineficientes, baja interoperabilidad institucional, falta de trazabilidad digital y escasa capacidad de adaptación ante aumento de flujo de personas, como en temporada de verano.

Este proyecto, plantea el desarrollo de un sistema de gestión digital para la aduana, cuyo propósito es mejorar y automatizar la tramitación de documentos obligatorios, facilitar el control en frontera y optimizar la coordinación entre las distintas instituciones involucradas, como el SAG, PDI y las aduanas de países vecinos, en este caso Argentina. A través de una plataforma web segura, accesible e intuitiva, se busca reducir significativamente los tiempos de espera y modernizar los servicios públicos asociados al cruce fronterizo.

El sistema propuesto contempla funcionalidades como la pre inscripción de documentos, la generación de formularios aduaneros en línea, el control digital de salidas e ingresos de vehículos, la validación de declaraciones juradas, la interoperabilidad entre sistemas nacionales e internacionales, y la generación de reportes estadísticos para la toma de decisiones. Todo esto se implementará bajo los estándares de seguridad, rendimiento y disponibilidad, cumpliendo con las normativas nacionales de seguridad y protección de datos personales. Este documento define no sólo el contexto, ámbito y objetivos del sistema, sino también su arquitectura técnica, proporcionando una visión estructural y funcional que guiará con buen paso su desarrollo y futura implementación

## Arquitectura del sistema

El sistema propuesto, se basa en el modelo de arquitectura 4+1 de Philippe Kruchten, el cual permite describir el diseño del software desde distintas perspectivas, facilitando su comprensión por parte de distintos actores del proyecto. Este modelo de arquitectura, posee cinco vistas, que fueron adaptadas al contexto del sistema del paso fronterizo Los Libertadores.

Vista de escenarios: Esta vista se centra en los principales casos de uso que debe cubrir el sistema. Estos escenarios permiten definir claramente los requerimientos funcionales y validar el comportamiento que se espera del sistema.

Vista lógica: Corresponde a la organización funcional del sistema, dividiéndolo en módulos como gestión de usuario, formularios, interoperabilidad, control fronterizo, reportes y auditoría. Esta vista permite representar la funcionalidad interna y cómo las partes del sistema interactúan para cumplir los requerimientos definidos.

Vista de desarrollo: Describe cómo se estructuran y desarrollan los componentes del sistema. Se propone una arquitectura modular, con tecnologías modernas tanto para el backend como para el frontend, organizando el código en repositorios, todo esto con el fin de facilitar el mantenimiento y el futuro del software.

Vista de procesos: Aborda la ejecución dinámica del sistema, considera procesos como validaciones simultáneas, generaciones de reportes automáticos, consulta de usuarios, entre otros. Esta vista asegura que el sistema soporte la carga durante temporadas de alto flujo, manteniendo tanto rendimiento, como disponibilidad.

Vista física: Representa la infraestructura en la que se desplegará el sistema. La arquitectura física garantizará alta disponibilidad, escalabilidad y seguridad, permitiendo al sistema operar correctamente incluso bajo condiciones críticas de uso

# VISIÓN DEL SISTEMA

## 2.1 Descripción general del sistema

La aduana de Chile se ocupa de fiscalizar, gestionar y controlar el ingreso y salida de mercancías, afirmando el cumplimento de las leyes tributarias, comerciales y sanitarias. Igualmente, recauda impuestos aduaneros, enfrenta el contrabando y protege el país del ingreso de productos ilegales o peligrosos.

## 2.2 Objetivos del sistema

Se propone el desarrollo de un sistema de software para la Aduana de Chile con el objetivo de modernizar y optimizar los procesos de fiscalización y facilitación del comercio exterior. Esta iniciativa busca fortalecer la recaudación fiscal, impulsar el desarrollo económico, mejorar la competitividad del país en los mercados internacionales y brindar un servicio más eficiente y transparente a los ciudadanos y empresas.

## 2.3 Principales funcionalidades esperadas

* Generar autorizaciones de salida para menores.
* Completar y emitir formularios de salida y admisión de vehículos.
* Gestionar declaración jurada de ingreso de alimentos y mascotas.
* Gestionar registro de ingreso y egreso de personas en pasos fronterizos.
* Generar reportes estadísticos de flujo de personas en PDF o Excel.

## 2.4 Supuestos y dependencias

* Se asume que el sistema requiere de un servidor físico para el almacenamiento y gestión de datos.
* Se asume que el personal de Aduanas contará con dispositivos (computadoras o tablets) adecuados para operar el sistema.
* Se asume que los usuarios recibirán la capacitación necesaria antes de la puesta en marcha del sistema.
* El sistema depende de la integración con la base de datos de la policía de investigaciones (PDI) para validar el estatus migratorio de los viajeros.
* La entrega del sistema depende de la aprobación del presupuesto por parte del Ministerio de Hacienda.
* El despliegue del sistema en puntos fronterizos depende de la infraestructura eléctrica y de red esté disponible y en funcionamiento.

# ESTILOS Y PATRONES ARQUITECTÓNICOS

## Estilo arquitectónico adoptado

En este proyecto, el estilo arquitectónico que se usará será en base a microservicios, que es una forma de diseñar y estructurar un software dividiéndolo en servicios pequeños, independientes y especializados, que trabajan de manera coordinada. Cada microservicio se encarga de una funcionalidad específica del sistema y se comunica con los demás mediante mecanismos como APIs.

A diferencia de las arquitecturas monolíticas tradicionales, donde todas las funcionalidades están integradas en una única aplicación, en los microservicios cada componente puede desarrollarse, desplegarse y mantenerse de forma independiente. Esto permite una mayor flexibilidad, facilita el trabajo en equipo, mejora al sistema en cuanto a fallos, y permite una escalabilidad más ágil a lo largo del tiempo. Cada microservicio puede utilizar tecnologías, bases de datos o lenguajes distintos si es necesario, siempre que se mantenga una comunicación clara con el resto del sistema.

## Justificación del estilo según el contexto del sistema

La decisión de adoptar un estilo arquitectónico basado en microservicios, es tanto por necesidades funcionales y operativas del proyecto, como también por las ventajas que este enfoque tiene en términos de escalabilidad, mantenimiento, disponibilidad y flexibilidad.

El caso de negocio tiene múltiples procesos distintos que están conectados entre sí, que tienen baja dependencia entre sí, lo que los hace ideales para ser modelados como servicios independientes dentro de una arquitectura de microservicios. Esto mejoraría significativamente la escalabilidad, ya que cada servicio puede crecer de forma independiente según la demanda. También favorece el despliegue continuo, permitiendo actualizar o reparar partes del sistema sin afectar el funcionamiento general, algo fundamental en contextos como este, donde la disponibilidad del servicio debe mantenerse incluso en temporadas de alto flujo como el verano.

Además, la arquitectura de microservicios mejora la resiliencia del sistema ante fallos. Si uno de los servicios presenta una falla, no necesariamente afecta al resto del sistema, lo que es en simples palabras una mayor estabilidad del software. También, facilita el trabajo distribuido de desarrollo, ya que distintos equipos pueden trabajar simultáneamente en distintos servicios, lo que provocaría una entrega más rápida, ágil y ordenada.

## Patrones de diseño aplicados

Para el desarrollo del sistema, se utilizará el patrón de diseño MVC (Modelo-Vista-Controlador) como base para la organización interna de cada microservicio. Este patrón tiene la capacidad de separar de forma clara la lógica de negocio, la interfaz de usuario y el control de flujo entre ambas.

El modelo se encarga de representar y gestionar los datos del sistema, así como las reglas de negocio asociadas. La vista es responsable de presentar la información al usuario, es decir, la interfaz gráfica. Finalmente, el controlador actúa como intermediario entre la vista y el modelo, ya que recibe las solicitudes del usuario, procesa la lógica necesaria y actualiza la vista o el modelo según corresponda.

Este modelo, es conveniente porque permite una mayor organización del código, facilita el mantenimiento y la escalabilidad, y permite que diferentes desarrolladores trabajen de forma simultánea en distintas capas del sistema. Esta separación de responsabilidades también hace más fácil aplicar cambios en la interfaz sin afectar la lógica del negocio. En un sistema complejo como el de aduanas, con múltiples módulos y tipos de usuario, MVC es una solución efectiva.

# MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS

## VISTA DE ESCENARIO

### Propósito

El propósito es describir cómo los diferentes actores interactúan con el sistema en situaciones concretas. Esta vista permite representar los principales casos de uso del sistema, verificando que las funcionalidades cumplan con las necesidades de los usuarios

### Actores

**Ciudadano/turista**: Cualquier persona que desee salir o ingresar al país a través del paso fronterizo Los Libertadores. Interactúa con el sistema para registrar documentación previa, completar formularios, realizar declaraciones juradas y consultar el estado de sus trámites.

**Funcionario de aduanas**: Encargado de revisar la documentación de los ciudadanos, validar sus formularios digitales, registrar tanto ingresos como salidas, y generar reportes relacionados al control fronterizo.

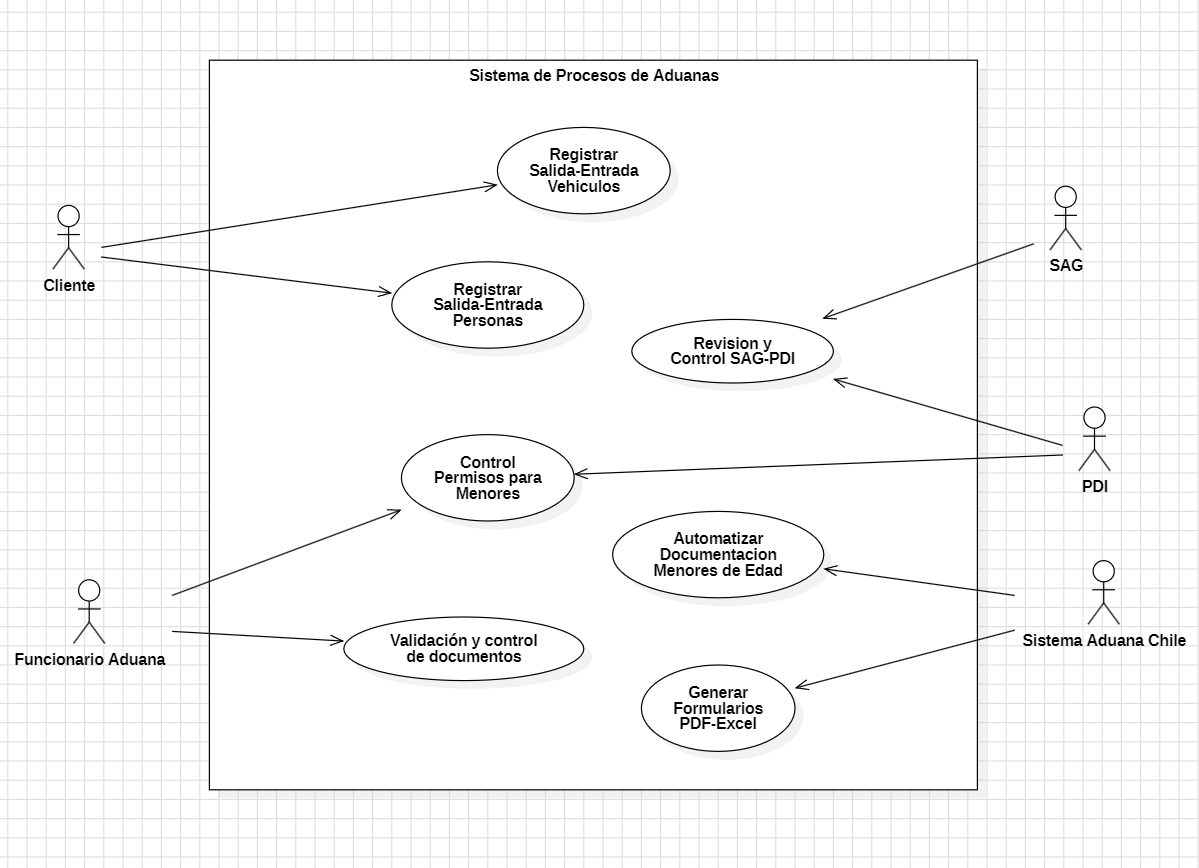
**Funcionario del SAG**: Responsable de verificar las declaraciones juradas de productos agrícolas o de origen animal. Interactúa con el sistema para validar documentación, registrar observaciones y controlar el cumplimiento de las normas sanitarias.

**Funcionario de la PDI**: Encargado del control migratorio y de seguridad en frontera. Usa el sistema para consultar antecedentes de los ciudadanos, validar documentación digital y registrar acciones realizadas durante el proceso de control.

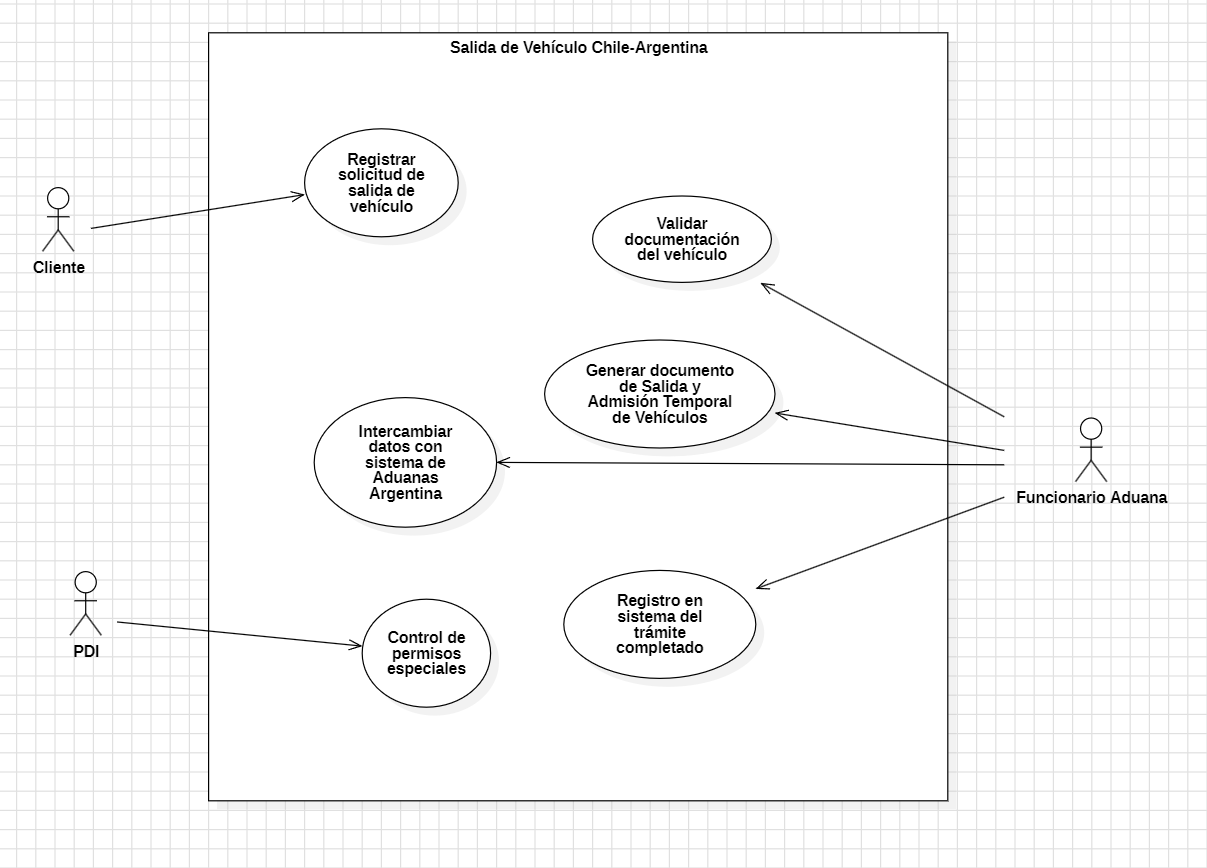
**Administrador del sistema**: Encargado de la gestión de usuarios, configuración de accesos, monitoreo del sistema, control de logs y respaldo de la información. Asegura la seguridad y funcionalidad del sistema en todo momento.

**Sistema de aduana argentina:** Actor que intercambia datos con el sistema nacional para validar formularios, identificar personas y facilitar el cruce bilateral en tiempo real entre ambos países.

### Diagrama general de casos de uso



### Diagrama de casos de uso específicos



### Lista de casos de uso

| **Código** | **Nombre** | **Actores** |
| --- | --- | --- |
| CU-001-001 | Preinscribir salida temporal de vehículo | Ciudadano |
| CU-001-002 | Generar formulario en PDF para salida de vehículo | Sistema |
| CU-001-003 | Validar formulario de salida en frontera | Funcionario de aduanas |
| CU-001-004 | Consultar estado del trámite del vehículo | Ciudadano |
| CU-001-005 | Registrar egreso del vehículo | Funcionario de aduanas |
| CU-001-006 | Declarar ingreso de alimentos o mascotas | Ciudadano |
| CU-001-007 | Validar declaración jurada en frontera | Funcionario SAG/PDI |
| CU-001-008 | Gestionar usuarios y roles | Administrador |
| CU-001-009 | Generar reportes estadísticos de egresos | Administrador |
| CU-001-010 | Interoperar datos con sistemas argentinos | Sistema |

### Especificación de casos de uso

| **Caso de Uso** | Preinscribir salida temporal de vehículo | **Identificador:**  CU-001-001 |
| --- | --- | --- |
| **Actores** | Ciudadano (primario), Sistema (secundario) | |
| **Tipo** | Primario | |
| **Referencias** | Completar y emitir formularios de salida de vehículos. | |
| **Precondición** | El ciudadano debe estar registrado y autenticado en el sistema. | |
| **Postcondición** | El sistema almacena la solicitud de salida del vehículo con los datos validados. | |
| **Descripción** | Este caso de uso permite a un ciudadano iniciar el proceso de salida temporal de su vehículo, ingresando en línea los datos requeridos para preinscribir el formulario oficial antes de presentarse en la frontera | |
| **Resumen** | El ciudadano accede al sistema, inicia sesión, completa los datos del conductor y del vehículo, y envía la solicitud. El sistema valida los datos y confirma el registro, dejándolo listo para generar el formulario. | |

**CURSO NORMAL**

| **Nro.** | **Ejecutor** | **Paso o Actividad** |
| --- | --- | --- |
| 1 | Ciudadano | Inicia sesión en el sistema |
| 2 | Sistema | Verifica credenciales y el acceso |
| 3 | Ciudadano | Selecciona opción “Salida temporal de vehículo” |
|  | | |

**CURSO ALTERNATIVO**

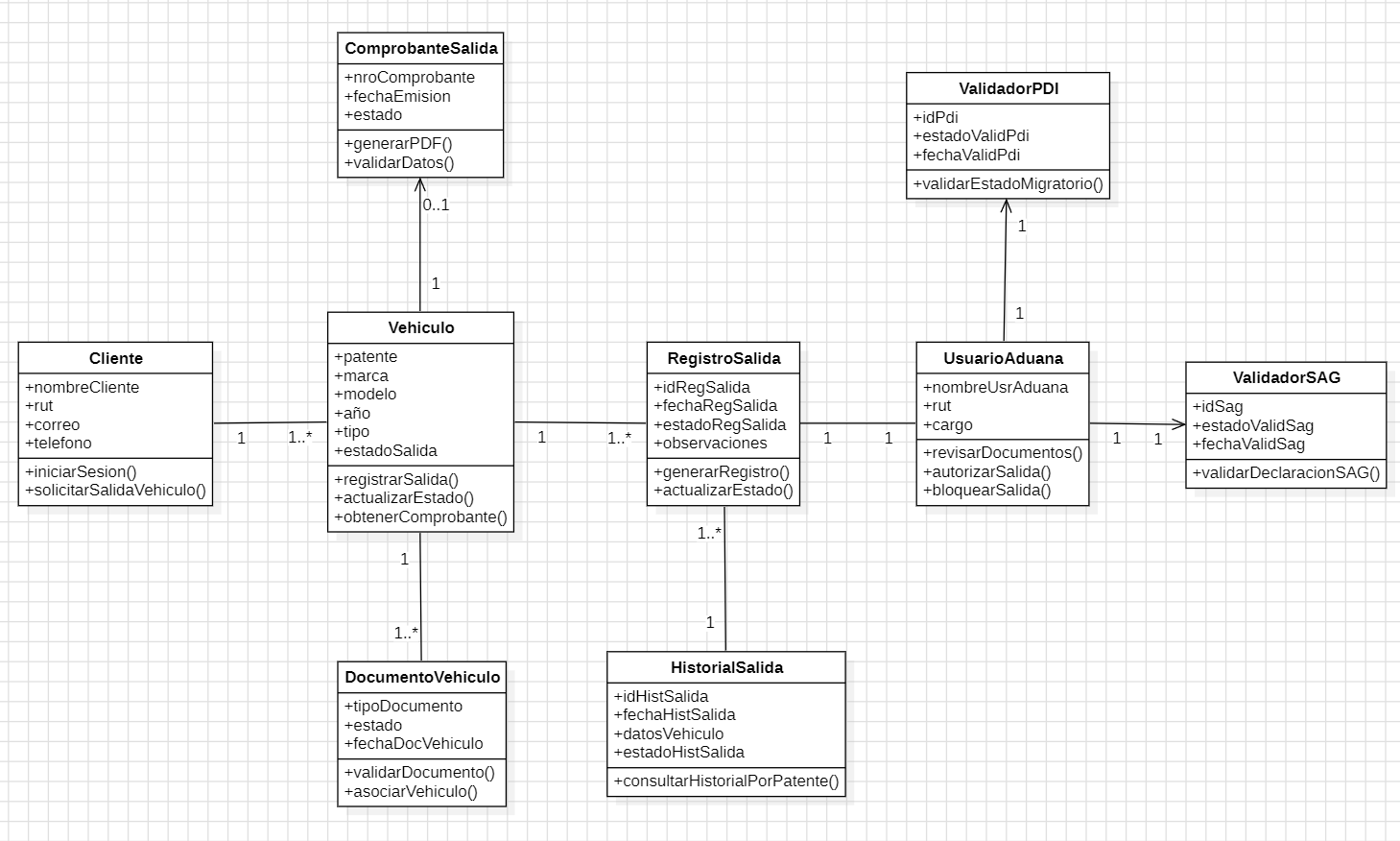
| **Nro.** | **Descripción de acciones alternas** |
| --- | --- |
| 1 | Si el ciudadano no tiene una cuenta registrada, el sistema le ofrece la opción de registrarse antes de iniciar sesión. |
| 2 | Si las credenciales ingresadas son incorrectas, el sistema muestra un mensaje de error y permite reintentar o recuperar la contraseña. |
| 3 | Si el ciudadano intenta acceder al módulo de salida temporal sin estar habilitado, el sistema muestra un mensaje de restricción e indica los pasos a seguir. |
|  | |

## VISTA LÓGICA

### Propósito

El propósito de esta vista lógica es definir la estructura funcional del sistema que gestiona la salida temporal de vehículos por el paso Los Libertadores. Permite organizar el sistema en módulos claramente diferenciados. Esta vista facilita la identificación de las responsabilidades funcionales del sistema, asegurando una separación clara entre la lógica de negocio, el manejo de los datos y la interfaz para el usuario

### Diagrama de clases



### Descripción diagrama de clases

El diagrama de clases modela la estructura del sistema enfocada en la salida de vehículos de Chile a Argentina. Este diagrama define las entidades principales, sus atributos y métodos relevantes, y las relaciones entre ellas, proporcionando una visión clara de las responsabilidades en el proceso de salida.

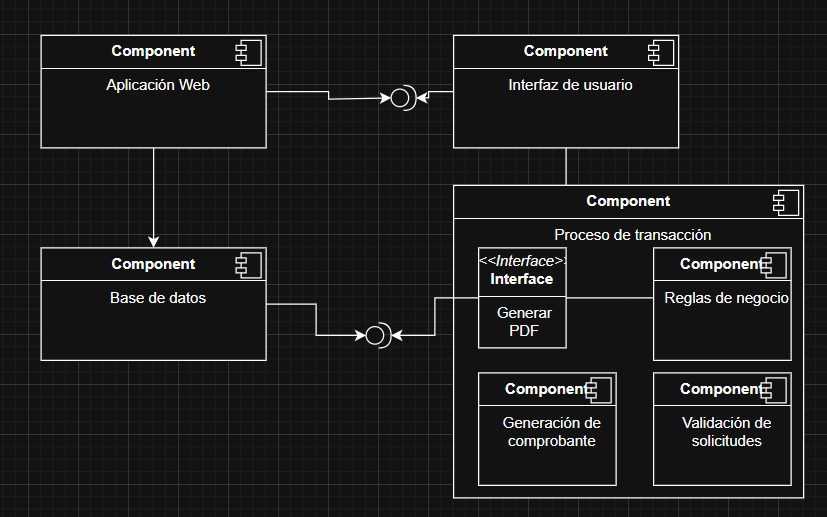
Este diagrama servirá para estructurar la base de datos, definir relaciones lógicas en la implementación y garantizar la consistencia de las entidades involucradas en la salida de vehículos.

## VISTA DE IMPLEMENTACIÓN/DESARROLLO

### Propósito

El propósito de esta vista, es mostrar cómo se implementará el sistema a nivel técnico, para gestionar el proceso de salida temporal de vehículos. Esto incluye la estructura del código, la organización por capas, y cómo se utilizarán tecnologías como backend, frontend, y APIs REST para la comunicación entre servicios. Esta vista es clave para el equipo de desarrollo, ya que les permite coordinarse y mantener un sistema limpio, modular y fácil de mantener.

### Diagrama de componente



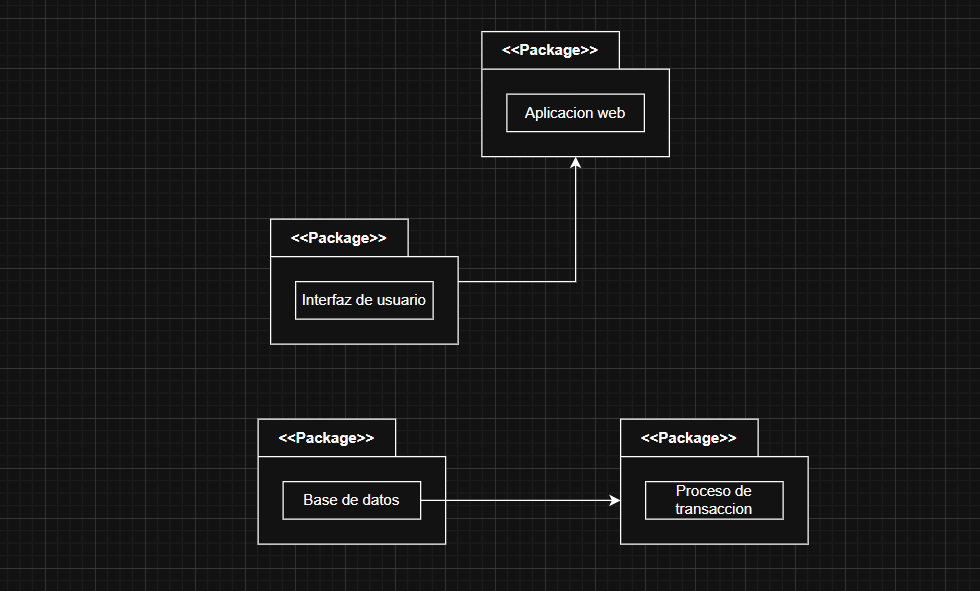
### Descripción diagrama de componente:

Este diagrama UML representa la estructura modular del sistema de salida temporal de vehículos, utilizado en el contexto aduanero, con componentes separados según sus responsabilidades.

Componentes:

En el componentes que gestiona la interacción inicial con el usuario, se conecta directamente con la base de datos para registrar y recuperar información y canaliza las solicitudes hacia los demás componentes. Después esta interfaz de usuario que presenta la información al usuario final, se comunica con el componente proceso de transacción para solicitar operaciones como la generación de documentos. Después, el proceso de transacción que es el núcleo de la lógica de negocio del sistema, proporciona la interfaz generar PDF, la cual es usada por la interfaz de usuario para emitir comprobantes oficiales y por ultimo esta la base de datos que es el componente encargado del almacenamiento persistente de la información y está conectado con Aplicación Web para el acceso general y con proceso de transacción para consultas específicas durante la generación de documentos.

### Diagrama de paquete



### Descripción diagrama de paquete

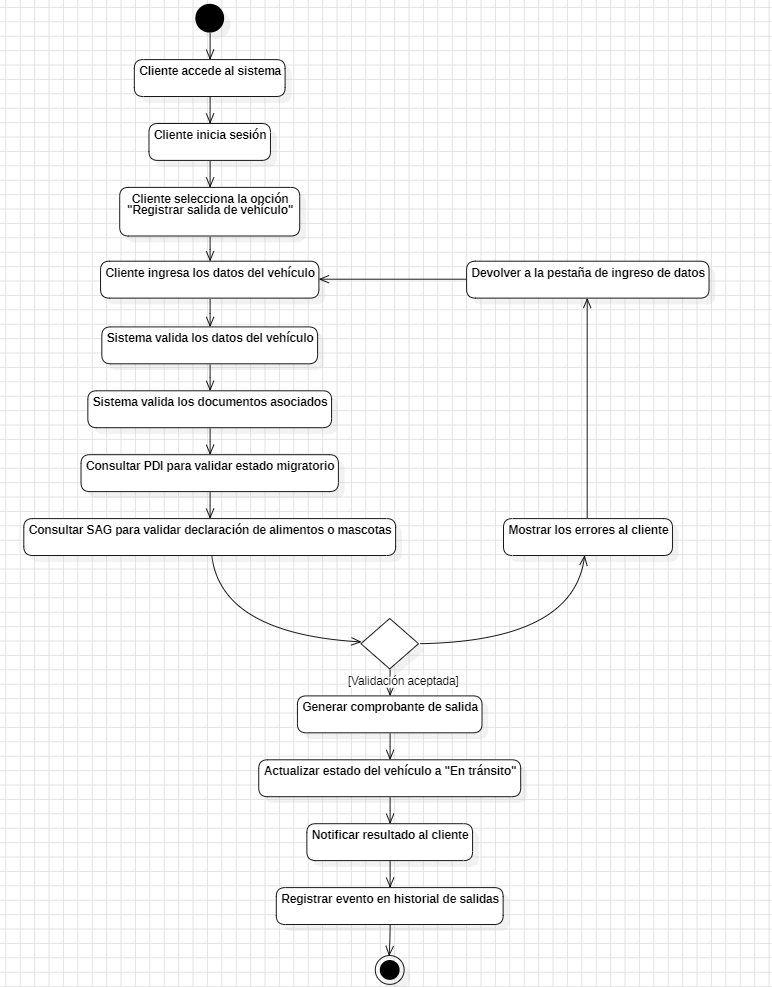
Muestra la estructura modular del sistema de salida temporal de vehículos para Aduanas, organizando sus partes en paquetes según su función. El paquete “Aplicacion web” coordina las operaciones del sistema y depende de “interfaz de usuario”, que gestiona la interacción con el ciudadano. “proceso de transacción” contiene la lógica de negocio, como validaciones y generación de comprobantes, y se conecta tanto con la interfaz como con la “Base de datos”, que almacena la información. Las dependencias entre paquetes, indicadas por flechas punteadas permiten visualizar cómo se comunican los módulos, facilitando un diseño claro, escalable y mantenible.

## VISTA DE PROCESOS

### Propósito

El propósito de la vista de procesos, es representar los flujos operativos internos del sistema de salida de vehículos, incluyendo todos los pasos que sigue el usuario en el sistema. Esta vista permite modelar la concurrencia, la interacción con otros procesos y asegurar que las operaciones sean eficientes y bien coordinadas, incluso bajo presión en temporadas de alto flujo.

### Diagrama de actividad



### Descripción diagrama de actividad

El diagrama de actividades describe el flujo dinámico de pasos y decisiones involucrados en el proceso de salida de vehículos de Chile a Argentina. Este modelo permite visualizar la interacción de los actores y el sistema para completar el proceso. Este comienza cuando el cliente accede al sistema, selecciona la opción de registrar la salida, e ingresa los datos del vehículo. A partir de aquí, el sistema realiza validaciones automáticas, si alguna de estas validaciones falla el sistema notifica al cliente los errores y lo devuelve al paso de ingreso de datos del vehículo, permitiendo la corrección de errores y reintento del trámite. Si todas las validaciones son correctas, el sistema genera un comprobante de salida, actualiza el estado del vehículo a "En tránsito", notifica el resultado al cliente y registra el evento en el historial.

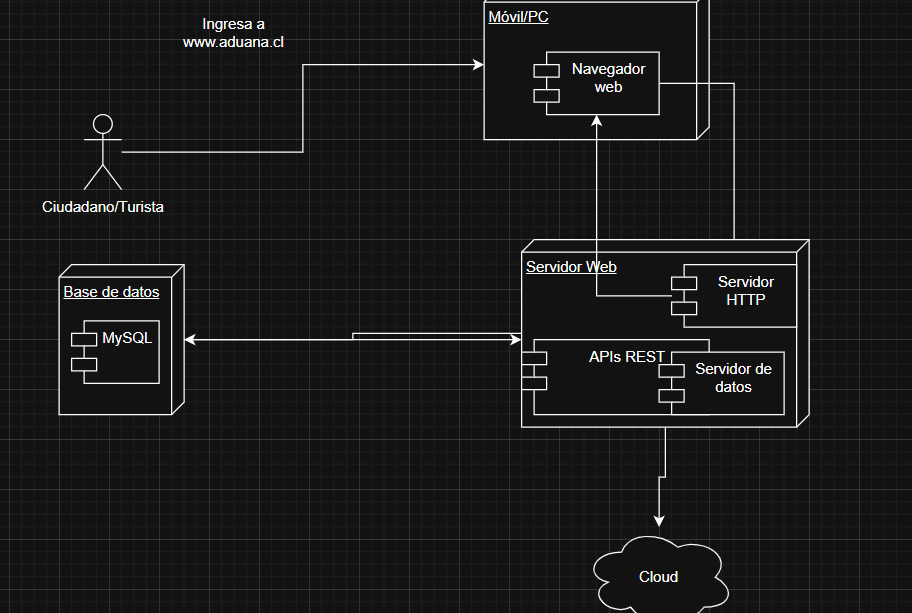
Este diagrama nos permite entender la lógica secuencial del proceso para diseñar pruebas funcionales y asegurar una experiencia de usuario fluida y sin fallas.

## VISTA FÍSICA (salida vehículo o entrada vehículo)

### Propósito

Describir la infraestructura tecnológica que soportará el sistema específico de salida de vehículos. Aquí se detalla cómo estará desplegado el sistema en servidores cloud, con base de datos segura, cifrado de comunicaciones, y respaldo automático de la información. Como este sistema estará en uso constantemente, esta vista permite garantizar su disponibilidad, escalabilidad y seguridad física y lógica.

### Diagrama de despliegue



### Descripción diagrama de despliegue

Representa la implementación del sistema de salida temporal de vehículos por el paso Los Libertadores. En él, el usuario (ciudadano o turista) accede desde su dispositivo móvil utilizando un navegador web. Este navegador se comunica con el servidor web a través de solicitudes HTTP. El servidor web contiene las APIs REST que gestionan la lógica del sistema y se encargan de procesar las solicitudes, conectando a su vez con una base de datos MySQL donde se almacena toda la información relevante del proceso. Esta arquitectura en capas permite una clara separación de responsabilidades entre la interfaz de usuario, la lógica de negocio y el almacenamiento de datos, asegurando que el sistema sea modular, escalable, fácil de mantener y capaz de responder de manera eficiente incluso en condiciones de alta demanda.

# REQUISITOS DE CALIDAD

## Propósito

Asegurar que el sistema cumpla con las expectativas de los usuarios y los requerimientos del negocio, ofreciendo una solución robusta, segura y confiable para el control de salida de vehículos en frontera.

## Atributos de calidad

| **ATRIBUTO DE CALIDAD** | **DESCRIPCIÓN** | **JUSTIFICACIÓN** |
| --- | --- | --- |
| Rendimiento | Tiempo de respuesta rápida en búsquedas, consultas e impresión de documentos. | Minimizar tiempos de espera en frontera, especialmente en alta demanda. |
| Usabilidad | Interfaz amigable, sencilla e intuitiva, diseñada para usuarios con distintos niveles de experiencia. | Facilitar el uso por parte de usuarios ocasionales y frecuentes. |
| Disponibilidad | Acceso al sistema 24/7, especialmente en temporadas de alto tráfico. | Asegurar continuidad operativa durante periodos críticos de tráfico. |
| Seguridad | Autenticación y autorización por roles, cifrado de datos sensibles, trazabilidad de acciones. | Proteger datos de los usuarios y del Estado; evitar fraudes o accesos no autorizados. |
| Interoperabilidad | Capacidad para conectarse con PDI, SAG y Aduana Argentina mediante APIs seguras. | Facilitar procesos de validación y control en frontera. |
| Portabilidad | Funcionamiento en distintos dispositivos y sistemas operativos. | Permitir el acceso desde distintos dispositivos y ubicaciones. |
| Mantenibilidad | Código modular y documentado para futuras mejoras. | Facilitar futuras actualizaciones, cambios y corrección de errores. |
| Escalabilidad | Capacidad para aumentar usuarios o procesos sin degradar el rendimiento. | Soportar flujos crecientes en temporadas de alta demanda. |
| Confiabilidad | Alta disponibilidad del servicio y recuperación rápida ante fallos. | Evitar interrupciones que puedan afectar el flujo normal de pasajeros. |
| Accesibilidad | Cumplir con estándares WCAG nivel AA para usuarios con discapacidad. | Asegurar la inclusión de todos los usuarios. |

## Reglas y criterios de evaluación de calidad

| **ATRIBUTO** | **DESCRIPCIÓN** | **JUSTIFICACIÓN** |
| --- | --- | --- |
| Rendimiento | Respuesta promedio de menos de 2 segundos en 95% de solicitudes. | Pruebas de carga y estrés. |
| Usabilidad | 80% de usuarios completan tareas sin ayuda en 1er intento. | Pruebas de usabilidad, validación con usuarios. |
| Seguridad | Cifrado de datos sensibles, 0 incidentes de fuga de datos. | Análisis de seguridad, pentesting, auditorías. |
| Interoperabilidad | Integración exitosa con PDI, SAG y Argentina en 100% de pruebas. | Pruebas de integración y simulación de datos externos. |
| Escalabilidad | Soporta al menos 200 solicitudes simultáneas sin degradación significativa. | Pruebas de carga simulada. |
| Confiabilidad | Alta disponibilidad del servicio y recuperación rápida ante fallos. | Evitar interrupciones que puedan afectar el flujo normal de pasajeros. |
| Accesibilidad | Uptime mínimo del 99% durante temporada alta. | Monitoreo y métricas de disponibilidad. |

# PRINCIPIOS DE DISEÑO APLICADOS

## Propósito

Guiar el desarrollo del sistema mediante principios sólidos de ingeniería de software para asegurar una solución robusta, escalable y mantenible.

## 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 |
| Bajo Acoplamiento | Reducir las dependencias entre módulos. | Comunicación entre microservicios mediante APIs estandarizadas. |
| Encapsulamiento | Ocultar la implementación interna y exponer sólo lo necesario. | Exposición de servicios a través de interfaces claras, manteniendo la lógica de negocio oculta. |
| Modularidad | Dividir el sistema en partes independientes. | Arquitectura basada en microservicios, permitiendo escalabilidad y mantenibilidad. |
| Escalabilidad | Soportar crecimiento del sistema sin degradar el rendimiento. | Uso de contenedores, balanceo de carga y servicios desacoplados para manejo de alta concurrencia. |
| Reusabilidad | Diseñar componentes reutilizables. | Servicios y módulos desacoplados que pueden integrarse en otros procesos de aduana. |

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

| **PRINCIPIO** | **APLICACIÓN EN EL SISTEMA** |
| --- | --- |
| Simplicidad | Interfaz clara, con botones e instrucciones fáciles de entender. |
| Consistencia | Uso coherente de colores, fuentes, íconos y estructuras en todas las pantallas. |
| Minimización de pasos | Flujos de usuario optimizados, evitando pasos innecesarios en los trámites. |
| Accesibilidad universal | Cumplimiento de estándares WCAG 2.1 nivel AA para usuarios con discapacidades. |
| Soporte visual e iconográfico | Uso de íconos e imágenes representativas para apoyar la comprensión de la interfaz. |
| Prevención de errores | Validaciones en tiempo real, mensajes descriptivos para datos incorrectos. |

# PROTOTIPO

## Propósito

Materializar una idea de manera temprana para evaluar su viabilidad técnica y funcional antes de invertir recursos en su desarrollo final. Esto facilita la identificación de errores, inconsistencias o mejoras necesarias, reduciendo riesgos y costos al realizar ajustes rápidamente y optimizando el proceso de diseño y desarrollo. Además, brinda una representación concreta del producto que puede ser comprendida, probada y retroalimentada de forma clara por los usuarios o clientes.

## Mockups (imágenes con una breve descripción)

## Justificar herramientas de prototipado

Utilizamos Figma como herramienta de prototipado porque nos permitió diseñar y simular de forma visual e interactiva la interfaz del producto antes de su desarrollo, permitiéndonos tener una comprensión clara de la experiencia del usuario. Además al ser una plataforma colaborativa, todos los miembros del equipo pudimos trabajar de manera simultánea en tiempo real y recibir retroalimentación inmediata, lo cual optimizó la comunicación y aceleró el proceso de diseño iterativo.

# EVALUACIÓN DE CALIDAD HEURÍSTICA DE NIELSEN

## Propósito

Identificar problemas de usabilidad en la interfaz de usuario para mejorar la experiencia de este, detectando de manera rápida y sin costo errores en el diseño que comprometen la experiencia del usuario antes de que el producto llegue al usuario final.

## 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)? | **✔** | El sistema posee mensajes indicando en qué paso está el usuario actualmente. | Baja gravedad debido a que el sistema informa claramente al usuario sobre cada paso. |
| 2 | **Correspondencia entre el sistema y el mundo real** | ¿La terminología y flujos se relacionan con el lenguaje y lógica del usuario? | **✔** | El sistema utiliza un lenguaje cotidiano y técnico de bajo nivel. | Baja gravedad debido a que el lenguaje utilizado en el sistema es comprensible para usuarios de cualquier nivel técnico. |
| 3 | **Control y libertad del usuario** | ¿El usuario puede deshacer/repetir acciones fácilmente? | **✔** | El usuario puede volver/borrar en caso de equivocación. | Baja gravedad debido a que el usuario puede deshacer sus acciones o corregir sus errores. |
| 4 | **Consistencia y estándares** | ¿Se mantiene un diseño coherente entre pantallas, botones y mensajes? | **✔** | Hay coherencia entre los distintos diseños y botones. | Baja gravedad debido a que el sistema posee coherencia entre el tamaño, color y diseño en la interfaz. |
| 5 | **Prevención de errores** | ¿El diseño evita que ocurran errores antes de que sucedan? | **✔** | El sistema le informa al usuario cuando el paso a realizar no se puede modificar. | Baja gravedad debido a que el sistema ofrece claridad total en caso de encontrarse con un paso no modificable posteriormente. |
| 6 | **Reconocimiento mejor que recuerdo** | ¿Las opciones y funciones son visibles sin que el usuario deba recordar información? | **✔** | Las funciones son de fácil uso y se pueden recordar fácilmente. | Baja gravedad debido a que el sistema se mantiene simple y fácil de usar con nombres identificativos para cada función. |
| 7 | **Flexibilidad y eficiencia de uso** | ¿Permite atajos o personalización para usuarios avanzados? | ✘ | Es de uso simple y no se pueden hacer atajos. | Gravedad moderada debido a que, por la simplicidad del sistema, los atajos personalizados no sumarían una eficiencia significativa en su uso. |
| 8 | **Diseño estético y minimalista** | ¿La interfaz evita información innecesaria o ruido visual? | **✔** | La interfaz posee figuras simples e iconos que acompañan la sección correspondiente | Baja gravedad debido a que el sistema se mantiene limpio de iconos y botones que no sean estrictamente necesarios o no aportan al diseño. |
| 9 | **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores** | ¿Los mensajes de error son claros, comprensibles y ofrecen solución? | ✘ | El prototipo no cubre mensajes de error. | Gravedad alta debido a que el sistema no muestra ningún mensaje/ventana de error al ocurrir alguno. |
| 10 | **Ayuda y documentación** | ¿Existe ayuda accesible, clara y orientada a la tarea cuando el usuario lo necesita? | **✔** | Existe un apartado de preguntas frecuentes. | Gravedad moderada debido a que hay ayuda para dudas generales, pero falta soporte para preguntas más específicas. |

## Análisis y métricas de resultados

El análisis de usabilidad basado en los 10 principios de Nielsen muestra un cumplimiento general del 80%, destacando sus fortalezas en la claridad del sistema, coherencia visual y prevención de errores. La mayoría de los criterios presentan baja gravedad, lo que indica una experiencia de usuario adecuada y comprensible. Sin embargo, existen áreas de mejora en la flexibilidad para usuarios avanzados, la ausencia de mensajes de error claros y la documentación limitada para resolver dudas específicas.

# CONTROL DE VERSIONES

## Propósito

Llevar un registro detallado de los cambios realizados a lo largo del tiempo, identificando quién, cuándo y qué se cambió exactamente. Esta práctica facilita la colaboración entre múltiples usuarios al permitir trabajar simultáneamente sobre un mismo proyecto sin sobrescribir el trabajo de los demás, teniendo también la posibilidad de revertir cambios a versiones anteriores en caso de errores, pérdidas de información o decisiones equivocadas. Además, mejora el seguimiento del progreso del proyecto, asegurando una mayor trazabilidad, seguridad y control en entornos dinámicos.

## Control de versión utilizado (justificar el tipo de control de versión utilizad (fecha, semántica o secuencial)

Optamos por utilizar el control de versiones semántico en nuestro proyecto porque posee una estructura clara y coherente en la numeración de versiones. Basamos este sistema en el formato MAJOR.MINOR.PATCH, incrementando el número mayor para cambios incompatibles con versiones antiguas, el menor para nuevas funcionalidades compatibles y el número de parche para correcciones menores. Gracias a esto, logramos reducir ambigüedades y facilitar la integración continua y el mantenimiento a lo largo del tiempo.

## Justificar herramientas de versionamiento

Utilizamos GitHub como herramienta de control de versiones en nuestro prototipo porque nos proporcionó mayor simplicidad en la colaboración entre los integrantes del equipo y asegurar una mayor organización y trazabilidad en el desarrollo. Gracias a sus funcionalidades, como las ramas (branches), pudimos trabajar en paralelo sin interferir en el trabajo de otros, revisar contribuciones antes de integrarlas y mantener un flujo de trabajo estructurado.

# **CONCLUSIONES**

El presente documento, definió la arquitectura del sistema de software para la gestión de salida temporal de vehículos, abordando tanto aspectos funcionales, como técnicos. A través del uso del modelo arquitectónico 4+1, una arquitectura basada en microservicios, y el patrón de diseño MVC, se planificó una solución modular, escalable y segura, acorde a los requerimientos del caso de negocio.

La propuesta busca optimizar los procesos aduaneros, mejorar la experiencia del usuario, reducir tiempos de espera, y garantizar la interoperabilidad con sistemas tanto nacionales como internacionales. Además, se consideraron estándares de calidad y normativas del estado chileno, asegurando que el sistema sea sostenible, confiable y adaptable a futuras necesidades.

# BIBLIOGRAFÍA

Sommerville, I. (2011). *Ingeniería del software* (9.ª ed.). Pearson Educación.

Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice* (3rd ed.). Addison-Wesley.