**Aduanas**

**(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** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Tabla de Contenidos**

[**1.**](#_heading=h.han7sm21gyh5) **INTRODUCCIÓN 4**

[1.1.](#_heading=h.4pzoxo215kcd) Contexto del Problema 4

[1.2.](#_heading=h.xujhgl2wjimu) Propósito 4

[1.3.](#_heading=h.jps3mj37fth6) Ámbito 4

[1.4.](#_heading=h.8qr0ktbbbej7) Definiciones, acrónimos y abreviaciones 4

[1.5.](#_heading=h.d1gqbmbdpzzz) Resumen ejecutivo 4

[1.6.](#_heading=h.lpqlc5w0rhla) Arquitectura del sistema 4

[**2.**](#_heading=h.wm9enypvrrrk) **VISIÓN DEL SISTEMA 4**

[2.1.](#_heading=h.vxung77nkbtn) Descripción general del sistema 5

[2.2.](#_heading=h.q265sox8pay6) Objetivos del sistema 5

[2.3.](#_heading=h.bsud7zvvdges) Principales funcionalidades esperadas 5

[2.4.](#_heading=h.87llcd6813e9) Supuestos y dependencias 5

[**3.**](#_heading=h.2x5139ect0gm) **ESTILOS Y PATRONES ARQUITECTÓNICOS 5**

[3.2.](#_heading=h.t8q12xymgijr) Justificación del estilo según el contexto del sistema 5

[**4.**](#_heading=h.86j02ao05m2h) **MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS 5**

[4.1.](#_heading=h.suftpw239r9f) VISTA DE ESCENARIO 5

[*4.1.1.*](#_heading=h.554msudrl9ff) *Propósito 5*

[*4.1.2.*](#_heading=h.5h1m84j4d030) *Actores 5*

[*4.1.3.*](#_heading=h.ffzvnxog48qx) *Diagrama general de casos de uso 5*

[*4.1.4.*](#_heading=h.zhcrmg3g4vkc) *Diagrama de casos de uso específicos 5*

[*4.1.5.*](#_heading=h.ev1rqcs59gfi) *Lista de casos de uso 5*

[*4.1.6.*](#_heading=h.viuf3r6srfho) *Especificación de casos de uso 5*

[4.2.](#_heading=h.gtzq32k8lfmv) VISTA LÓGICA 7

[*4.2.1.*](#_heading=h.mbc6uj2wnuiw) *Propósito 7*

[*4.2.2.*](#_heading=h.q80gyhjaijr3) *Diagrama de clases 7*

[*4.2.3.*](#_heading=h.63btoe9dovfi) *Descripción diagrama de clases 7*

[4.3.](#_heading=h.ctuus7x8zkwo) VISTA DE IMPLEMENTACIÓN/DESARROLLO 7

[*4.3.1.*](#_heading=h.8q408jfmxxu1) *Propósito 7*

[*4.3.2.*](#_heading=h.wx0t4eezsvhb) *Diagrama de componente 7*

[*4.3.3.*](#_heading=h.4kp7xtx2m8g0) *Descripción diagrama de componente 7*

[*4.3.4.*](#_heading=h.pfi7fbvl689s) *Diagrama de paquete 7*

[*4.3.5.*](#_heading=h.ojndg7ef088f) *Descripción diagrama de paquete 7*

[4.4.](#_heading=h.6z40sma04n8u) VISTA DE PROCESOS 7

[4.4.1.](#_heading=h.r2ll4piy7fdl) Propósito 7

[4.4.2.](#_heading=h.hrctiqsihyf0) Diagrama de actividad 7

[4.4.3.](#_heading=h.xuzbj5fc7lru) Descripción diagrama de actividad 7

[4.5.](#_heading=h.o2jr8kvu5227) VISTA FÍSICA 7

[*4.5.1.*](#_heading=h.ah273fpwaume) *Propósito 7*

[*4.5.2.*](#_heading=h.75wqp6qij0kq) *Diagrama de despliegue 7*

[*4.5.3.*](#_heading=h.fkdix134qqfe) *Descripción diagrama de despliegue 7*

[5.](#_heading=h.iv3abwcn6zjn) REQUISITOS DE CALIDAD 7

[5.1.](#_heading=h.pe8dshw3pb72) Propósito 7

[*5.3.*](#_heading=h.jkasp7nmgeb4) *Reglas y criterios de evaluación de calidad 7*

[**6.**](#_heading=h.1m0ledfsga70) **PRINCIPIOS DE DISEÑO APLICADOS** 8

[*6.1.*](#_heading=h.cc2l4nq70whn) *Propósito 8*

[*6.4.*](#_heading=h.jdsu7bstkgn5) *Diseño centrado en el usuario (UX/UI, prototipos, experiencia de usuario) 8*

[**7.**](#_heading=h.3e02gmb1ifri) **CONCLUSIONES 8**

[**8.**](#_heading=h.os49nvjfys58) **BIBLIOGRAFÍA 8**

1. **INTRODUCCIÓN**

* 1. **Contexto del Problema**

El sistema aduanero chileno cumple funciones clave de fiscalización, control y facilitación del tránsito de personas, vehículos y mercancías a través de las fronteras nacionales. Sin embargo, actualmente enfrenta un problema crítico: los largos tiempos de espera que se generan en los pasos fronterizos, especialmente durante temporadas altas o feriados. Estas demoras afectan la experiencia de los viajeros, reducen la eficiencia operativa y generan un impacto negativo tanto en términos logísticos como económicos.

A partir de esta necesidad, surge la propuesta de desarrollar un sistema de software que optimice el proceso de validación documental, control vehicular y registro de productos y mascotas, mediante digitalización, integración con otros organismos y automatización de tareas clave.

* 1. **Propósito**

El presente Documento de Arquitectura del Sistema (DAS) tiene como objetivo describir la estructura y organización del software propuesto para mejorar la eficiencia del proceso aduanero en los pasos fronterizos entre Chile y Argentina. Se detallan las decisiones arquitectónicas, los patrones utilizados, los requisitos funcionales y no funcionales, y las distintas vistas arquitectónicas que soportan la solución.

* 1. **Ámbito**

Este sistema está diseñado para ser utilizado por ciudadanos nacionales y extranjeros que crucen la frontera, así como por los funcionarios del Servicio Nacional de Aduanas. Se limita a las validaciones asociadas a la **salida e ingreso de vehículos particulares**, **productos no considerados equipaje**, y **mascotas (perros y gatos)**. El registro migratorio de personas no forma parte del alcance, y se considera fuera de este proyecto.

El sistema contempla también la interoperabilidad con organismos como el **SAG** y la **PDI**, y la validación de documentos con el **Registro Civil** y **seguros internacionales**.

* 1. **Definiciones, acrónimos y abreviaciones**

| **ACRONIMO** | **DESCRIPCION** |
| --- | --- |
| *SAG* | Servicio Agrícola y Ganadero |
| *PDI* | Policía de Investigaciones de Chile |
| *CZE* | Certificado Zoosanitario de Exportación |

* 1. **Resumen ejecutivo**

Este proyecto propone un sistema informático orientado a reducir los tiempos de espera en los pasos fronterizos mediante la digitalización de formularios, automatización de validaciones y generación de informes en tiempo real. Su arquitectura contempla componentes modulares que permiten escalar el sistema y facilitar su mantenimiento, además de garantizar la interoperabilidad con sistemas externos (SAG, PDI, Registro Civil, etc.). Se aplican principios de arquitectura orientada a servicios y buenas prácticas de desarrollo, buscando mejorar tanto la experiencia del usuario como la eficiencia del personal aduanero.

* 1. **Arquitectura del sistema**

La arquitectura del sistema se describe mediante el modelo 4+1, que incluye las siguientes vistas:

* **Vista de Escenario:** Diagramas de casos de uso y especificaciones asociadas.
* **Vista Lógica:** Diagrama de clases y modelo del dominio.
* **Vista de Desarrollo:** Diagrama de componentes del sistema.
* **Vista de Procesos:** Diagramas de actividad para flujos clave como validación vehicular y control de productos.
* **Vista Física:** Diagrama de despliegue que muestra la infraestructura y la interacción entre componentes distribuidos.

1. **VISIÓN DEL SISTEMA**

* 1. **Descripción general del sistema**

El sistema propuesto corresponde a una plataforma web de uso transversal (independiente del dispositivo o sistema operativo), orientada a facilitar los trámites previos al cruce fronterizo entre Chile y países limítrofes. Los ciudadanos deberán completar en línea la declaración de salida o ingreso de vehículos, productos no considerados equipaje y/o mascotas, adjuntando los documentos requeridos según el caso.

Una vez en el paso fronterizo, y tras haber sido validada previamente la identidad del viajero por parte de la **PDI**, el **técnico en revisión no invasiva** accede al sistema para validar electrónicamente la información declarada. El sistema verifica automáticamente el cumplimiento de los requisitos mediante integraciones con servicios externos (como el Registro Civil, SAG y aseguradoras), y permite al funcionario asignar un estado al trámite: **aceptado**, **pendiente** o **rechazado**, según corresponda.

* 1. **Objetivos del sistema**

El sistema tiene como objetivo principal disminuir significativamente los tiempos de espera en los pasos fronterizos terrestres de Chile, permitiendo que los viajeros puedan realizar trámites previos de manera digital antes de llegar al punto de control. De este modo, se busca agilizar los procesos de revisión documental, reducir las filas presenciales y mejorar la experiencia de viaje.

Entre los objetivos específicos del sistema se encuentran:

* Facilitar la declaración anticipada de salida o ingreso de vehículos, productos no considerados equipaje y mascotas (perros y gatos).
* Eliminar progresivamente el uso de documentos físicos en los procesos aduaneros.
* Automatizar la verificación de requisitos mediante validaciones en línea con sistemas externos (Registro Civil, SAG, PDI, seguros).
* Permitir que los técnicos en revisión y agentes aduaneros accedan a la información de manera rápida y segura.
* Generar informes estadísticos que permitan evaluar el rendimiento operativo en puntos fronterizos.
* Asegurar la disponibilidad continua del sistema, incluso en momentos de alta demanda.

Estos objetivos están alineados con la política de modernización y eficiencia del Servicio Nacional de Aduanas, y con los requerimientos funcionales y no funcionales establecidos para este proyecto.

* 1. **Principales funcionalidades esperadas**

El sistema propuesto se enfoca exclusivamente en el proceso de **salida de vehículos particulares** desde Chile hacia países limítrofes, especialmente Argentina. Está diseñado para permitir a los ciudadanos realizar una **declaración previa en línea**, y para que los funcionarios de aduanas puedan validar dicha información de forma eficiente en el paso fronterizo.

Las funcionalidades esperadas se agrupan de la siguiente manera:

#### **Funcionalidades para el ciudadano:**

* Registro y autenticación en la plataforma web.
* Llenado del **formulario de salida de vehículos** (Anexo 1), incluyendo:  
  + Información del conductor y del vehículo.
  + Adjuntar documentos obligatorios: padrón del vehículo, seguro internacional de responsabilidad civil.
  + En caso de no ser el dueño del vehículo, adjuntar poder notarial de autorización.
* Consulta del estado de revisión del trámite (aceptado, pendiente o rechazado).
* Eliminación de la necesidad de portar documentación impresa.

#### **Funcionalidades para agentes aduaneros:**

* Acceso al sistema con autenticación reforzada.
* Visualización de solicitudes previamente registradas por ciudadanos.
* Validación automática de documentos mediante integración con:  
  + **Registro Civil** (datos del padrón).
  + **Sistema de seguros internacionales**.
* Validación del poder notarial si el conductor no es el propietario.
* Confirmación de validación previa por parte de la **PDI** como paso obligatorio antes de revisar.
* Asignación de estado al trámite: aceptado, pendiente o rechazado.
* Generación de informes estadísticos para el jefe operativo de aduana.

* 1. **Supuestos y dependencias**

Se establece como parte del diseño funcional que los ciudadanos deben realizar el trámite de declaración de salida de vehículos **antes de cruzar el paso fronterizo**, utilizando la plataforma web habilitada para ello.

Para que el técnico en revisión no invasiva pueda iniciar la validación, **la autorización previa por parte de la PDI debe estar confirmada** en el sistema.

El sistema debe contar con **interoperabilidad mínima** con el Registro Civil y los sistemas de seguros internacionales, con el fin de validar los documentos adjuntos por el ciudadano.

La validación de poderes notariales y documentos escaneados se considera parte del proceso operativo, pero no se contempla inteligencia documental avanzada en esta etapa del proyecto.

Aspectos relacionados con la conectividad en terreno, requisitos de infraestructura tecnológica y responsables de operación y mantenimiento del sistema **no se abordan en esta versión**, por estar fuera del alcance definido del proyecto actual.

1. **ESTILOS Y PATRONES ARQUITECTÓNICOS**

* 1. **Estilo arquitectónico adoptado**

El sistema será desarrollado bajo una **arquitectura monolítica**, en la cual todos los módulos funcionales —interfaz de usuario, lógica de negocio y acceso a datos— estarán contenidos en una misma aplicación. Esta elección responde a criterios de simplicidad, cohesión estructural y facilidad de implementación, considerando el contexto académico del proyecto y el alcance acotado a la salida de vehículos.

La arquitectura monolítica facilita la integración entre módulos sin requerir comunicaciones distribuidas, lo que la convierte en una opción adecuada para entornos donde los recursos técnicos, el tiempo y los requerimientos de escalabilidad aún no son críticos.

* 1. **Justificación del estilo según el contexto del sistema**

La combinación de una arquitectura monolítica con el patrón MVC responde a los siguientes criterios:

* **Simplicidad**: Ideal para proyectos acotados y con recursos limitados.
* **Facilidad de desarrollo y despliegue**: Se trabaja con una sola unidad de software, sin necesidad de coordinar múltiples servicios.
* **Organización del código**: MVC permite una estructura clara y mantenible incluso dentro de una única aplicación.
* **Escalabilidad futura**: Aunque el sistema se construye como monolito, su organización interna permite evolucionar hacia estructuras más modulares si fuese necesario.
  1. **Patrones de diseño aplicados**

Para organizar la estructura interna del sistema y asegurar una separación clara de responsabilidades, se utilizará el **patrón Modelo–Vista–Controlador (MVC)**. Este patrón permite dividir el sistema en tres componentes fundamentales:

* **Modelo**: Contiene la lógica de negocio y el manejo de datos del sistema.
* **Vista**: Representa la interfaz de usuario, encargada de la presentación de datos.
* **Controlador**: Gestiona la comunicación entre la vista y el modelo, y orquesta el flujo de la aplicación.

El patrón MVC permite mantener un bajo acoplamiento entre componentes, facilitando el mantenimiento, las pruebas y futuras extensiones del sistema, sin comprometer la simplicidad de una arquitectura monolítica.

1. **MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS**

* 1. **VISTA DE ESCENARIO**

* + 1. **Propósito**

La vista de escenario tiene como objetivo representar los diferentes **flujos de interacción entre los actores del sistema y sus funcionalidades principales**, desde la perspectiva externa del usuario. Esta vista permite identificar claramente los **casos de uso clave** que serán soportados por el sistema, agrupándolos según el rol que los ejecuta. En este proyecto, se centra en el proceso de **salida de vehículos desde Chile hacia Argentina**.

* + 1. **Actores**

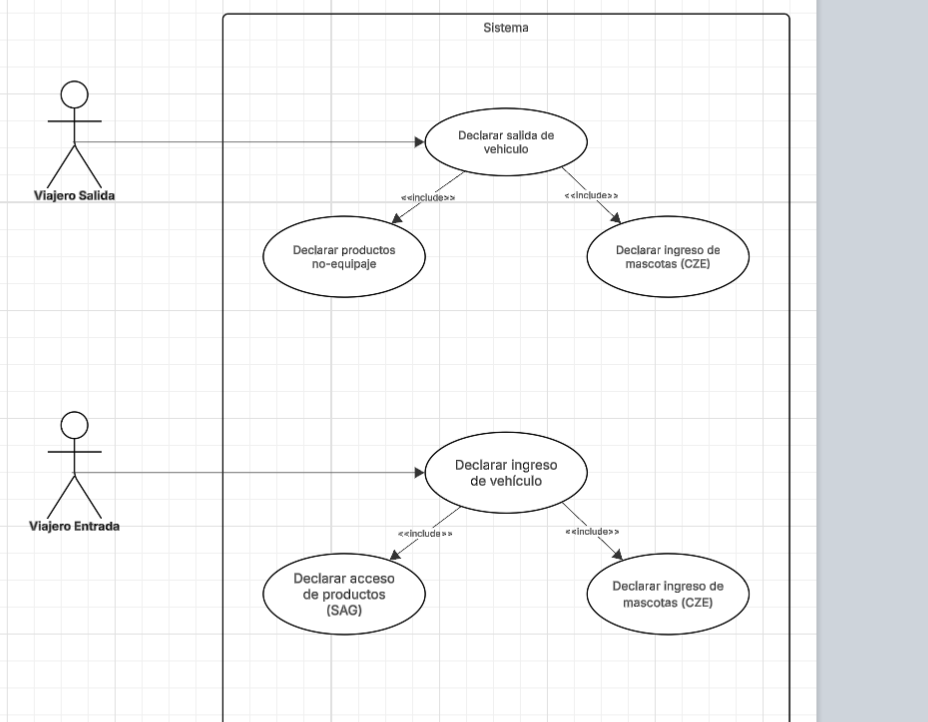
Los actores que interactúan con el sistema en el contexto de la salida de vehículos son los siguientes:

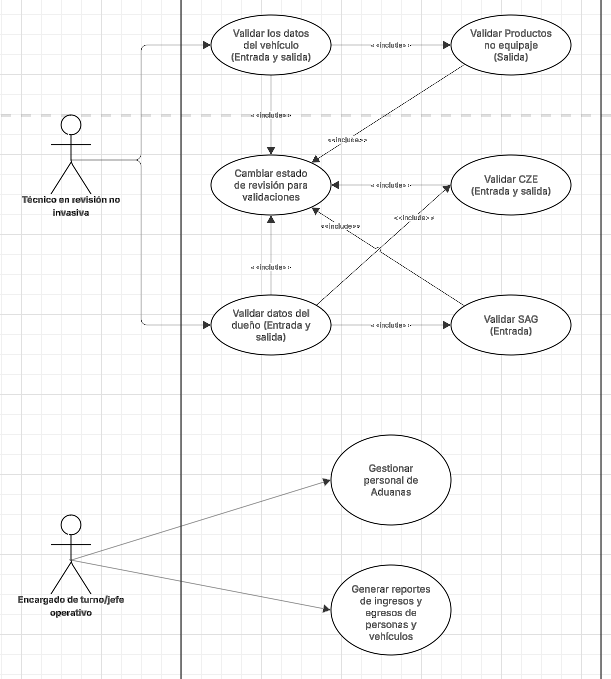
| **Actor** | **Descripción** |
| --- | --- |
| **Viajero salida** | Persona que declara la salida de un vehículo desde Chile y presenta los documentos requeridos. |
| **Técnico en revisión no invasiva** | Funcionario aduanero encargado de validar los documentos cargados por el viajero. |
| **Encargado de turno / Jefe operativo** | Responsable de generar reportes y visualizar el estado general de los trámites realizados. |

* + 1. **Diagrama general de casos de uso**

**Diagrama General de Casos de Uso del Sistema**

*Representa la relación entre los actores principales (viajero, técnico, jefe operativo) y los módulos del sistema relacionados con la declaración y validación de salida de vehículos.*

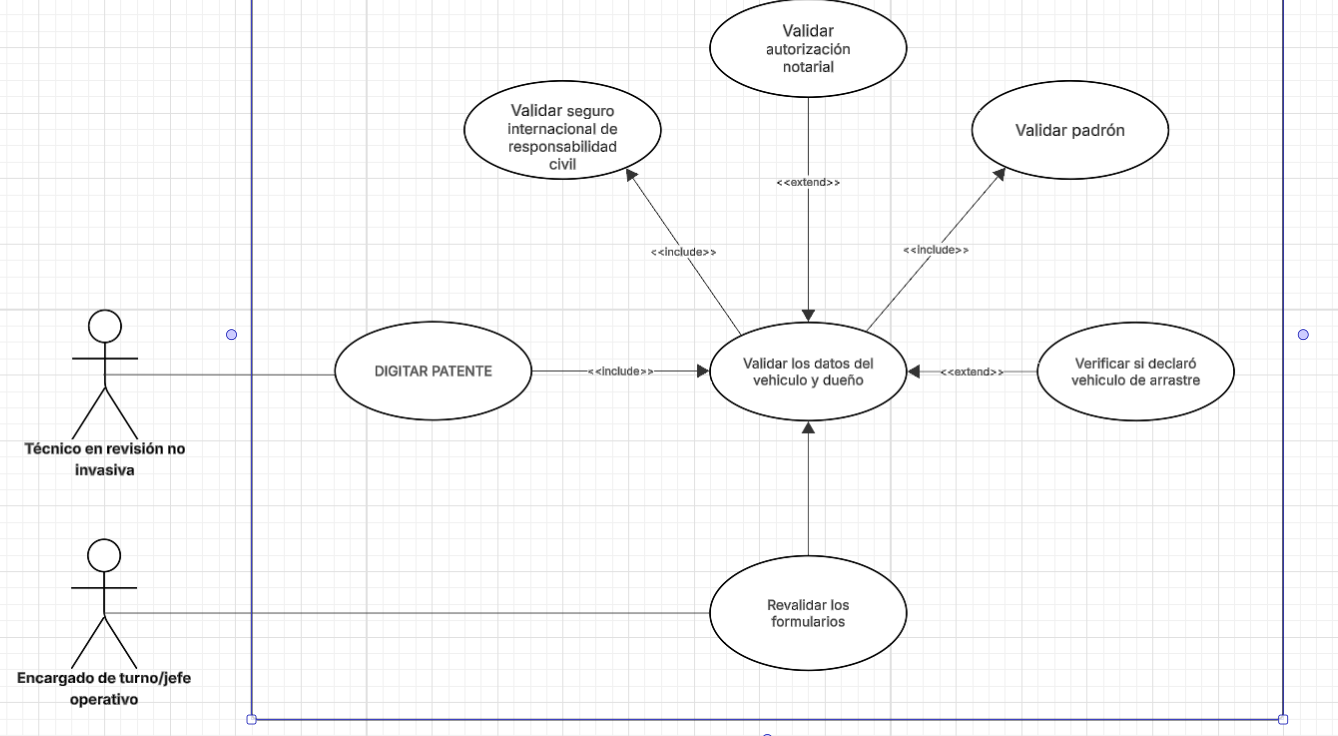
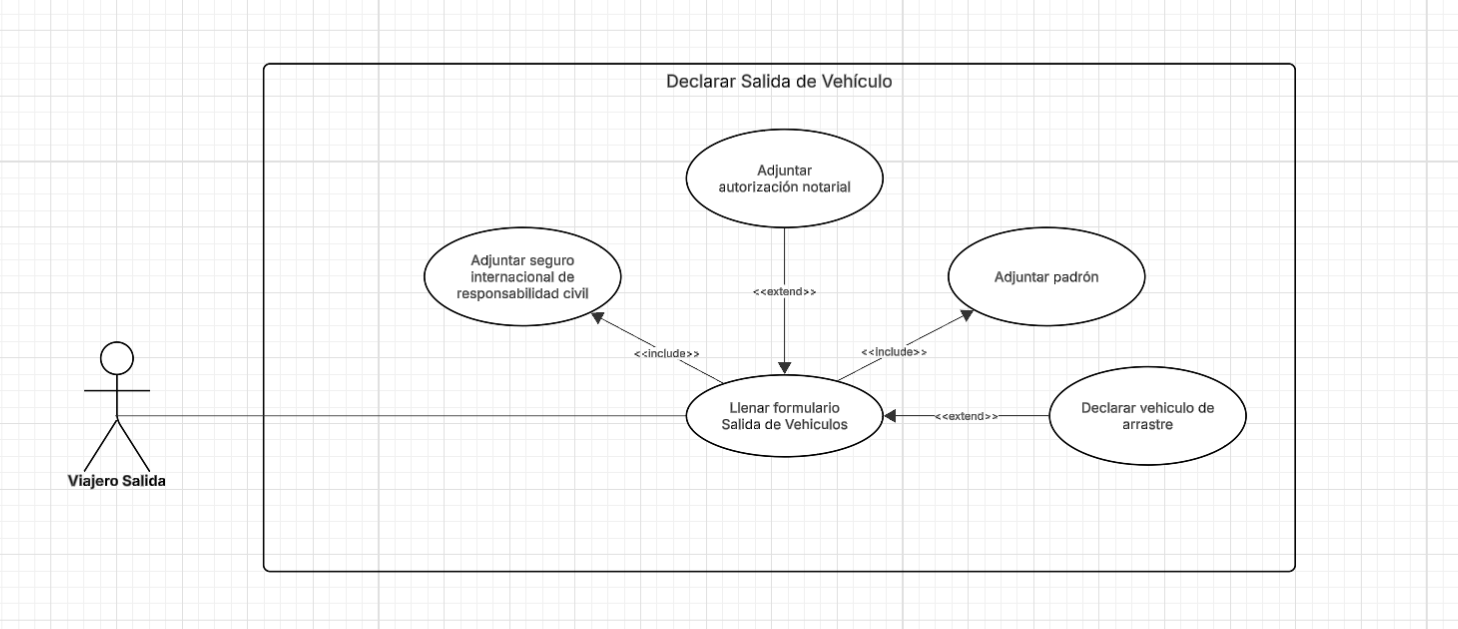




* + 1. **Diagrama de casos de uso específicos**

**Diagrama Específico de Caso de Uso: Salida de Vehículos**

*Este diagrama detalla los pasos y validaciones asociadas al proceso de salida de vehículos, incluyendo relaciones de inclusión y extensión.*

**

* + 1. **Lista de casos de uso**

| **Código** | **Nombre de Caso de Uso** | **Actores Involucrados** |
| --- | --- | --- |
| CU-SAL-001 | Gestionar salida de vehículo | Viajero salida, Técnico, Encargado de turno/Jefe operativo |
| CU-SAL-002 | Llenar formulario de salida | Viajero salida |
| CU-SAL-003 | Adjuntar padrón vehicular | Viajero salida |
| CU-SAL-004 | Adjuntar seguro internacional | Viajero salida |
| CU-SAL-005 | Adjuntar poder notarial | Viajero salida |
| CU-SAL-006 | Declarar vehículo de arrastre | Viajero salida |
| CU-SAL-007 | Digitar patente para búsqueda | Técnico revisión |
| CU-SAL-008 | Validar seguro internacional | Técnico revisión |
| CU-SAL-009 | Validar padrón vehicular | Técnico revisión |
| CU-SAL-010 | Validar autorización notarial | Técnico revisión |
| CU-SAL-011 | Validar datos del vehículo y dueño | Técnico revisión |
| CU-SAL-012 | Revalidar formularios | Técnico revisión |
| CU-SAL-013 | Verificar si declaró vehículo de arrastre | Técnico revisión |
| CU-SAL-014 | Generar reportes de egresos vehiculares | Encargado de turno/Jefe operativo |

* + 1. **Especificación de casos de uso**

| **Caso de Uso** | Digitar patente para búsqueda | **Identificador:** CU-SAL-007 |
| --- | --- | --- |
| **Actores** | Técnico en revisión no invasiva | |
| **Tipo** | Principal | |
| **Referencias** | * Diagrama de casos de uso específico (salida de vehículos) * Requerimientos funcionales: Validación vehicular, interoperabilidad | |
| **Precondición** | * El viajero ha realizado el trámite de declaración de salida. * La autorización de salida por parte de la PDI ya ha sido confirmada. | |
| **Postcondición** | * El sistema localiza y muestra al técnico el formulario de salida, con todos los documentos adjuntos por el viajero. | |
| **Descripción** | Este caso de uso permite al **técnico en revisión no invasiva** ingresar la **patente del vehículo** en el sistema con el fin de **acceder al formulario de salida previamente completado por el viajero**. Esta búsqueda es necesaria para visualizar los datos y documentos adjuntados (padrón, seguro, autorización notarial si aplica), e iniciar el proceso de validación. | |
| **Resumen** | Mediante la digitación de la patente, el técnico localiza el trámite asociado a un vehículo en tránsito, lo que habilita la revisión documental previa a autorizar la salida. Este paso actúa como **puerta de entrada operativa** para el proceso de fiscalización en frontera. | |

**CURSO NORMAL**

| **Nro.** | **Ejecutor** | **Paso o Actividad** |
| --- | --- | --- |
| 1 | Técnico | Ingresa al sistema y selecciona el módulo de revisión. |
| 2 | Técnico | Digita la patente del vehículo en el buscador del sistema. |
| 3 | Sistema | Busca la solicitud asociada a esa patente y verifica que esté habilitada para revisión. |
| 4 | Sistema | Muestra los datos del trámite: formulario completo y documentos cargados. |
| 5 | Técnico | Revisa que los datos correspondan al vehículo presente. |

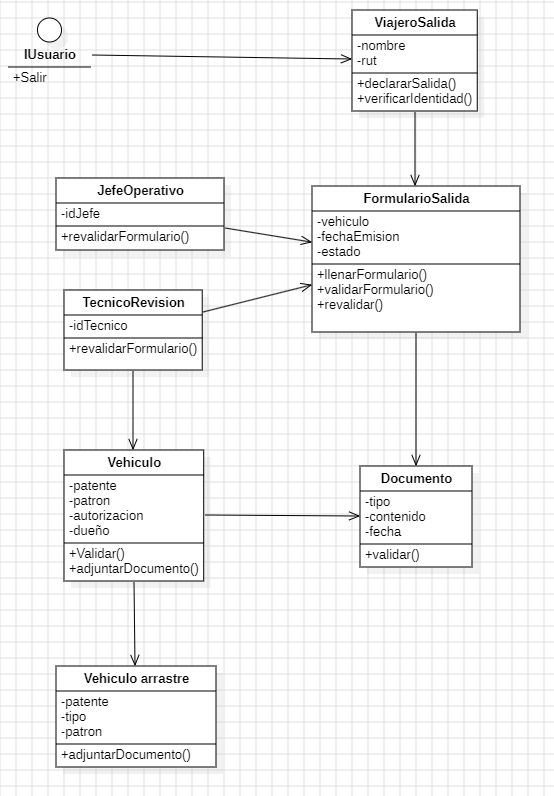
**CURSO ALTERNATIVO**

| **Nro.** | **Descripción de acciones alternas** |
| --- | --- |
| 3 | Si no existe ningúna declaración asociada a la patente, el sistema muestra un mensaje de error y detiene el flujo. |

* 1. **VISTA LÓGICA**
     1. **Propósito**

Esta vista representa la estructura interna del sistema desde una perspectiva orientada a objetos. Permite comprender la organización de clases, sus atributos, operaciones y las relaciones que sostienen entre ellas. Es fundamental para modelar la lógica del dominio aduanero que soporta el proceso de salida de vehículos.

* + 1. **Diagrama de clases**



* + 1. **Descripción diagrama de clases**

A continuación, se describen las clases clave del sistema:

**ViajeroSalida** Representa al ciudadano que declara la salida del vehículo.

* *Atributos:* nombre, rut
* *Métodos:* declararSalida(), verificarIdentidad()

**Vehiculo** Contiene los datos del vehículo que se declara.

* *Atributos:* patente, patron, autorizacion, dueño
* *Métodos:* validar(), adjuntarDocumento()

**FormularioSalida** Representa el formulario digital completado por el viajero.

* *Atributos:* vehiculo, fechaEmision, estado
* *Métodos:* llenarFormulario(), validarFormulario(), revalidar()

**TecnicoRevision** Funcionario que revisa la documentación del trámite.

* *Atributos:* idTecnico
* *Métodos:* revalidarFormulario()

**VehiculoArrastre** Clase especializada para declarar un vehículo adicional (como remolque).

* *Atributos:* patente, tipo, patron
* *Métodos:* adjuntarDocumento()

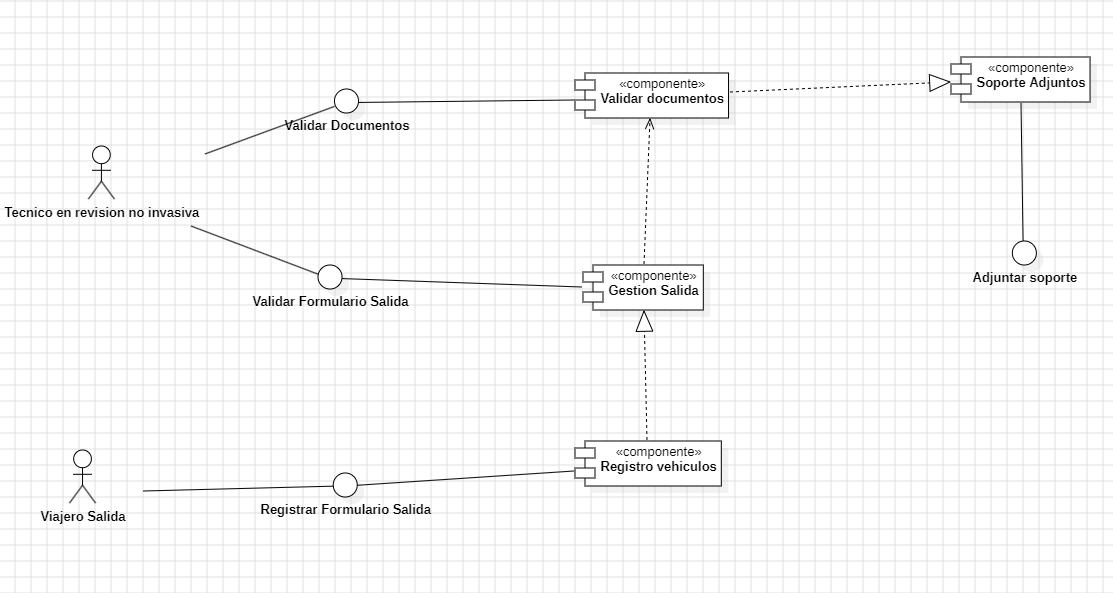
**Documento** Representa cualquier documento adjunto (ej. padrón, seguro, poder notarial).

* *Atributos:* tipo, contenido, fecha
* *Métodos:* validar()

**JefeOperativo** Responsable de generar reportes o validar procesos especiales.

* *Atributos:* idJefe
* *Operaciones:* revalidarFormulario()

* 1. **VISTA DE IMPLEMENTACIÓN/DESARROLLO** (salida vehículo **o** entrada vehículo)
     1. Propósito
     2. Diagrama de componente



* + 1. Descripción diagrama de componente

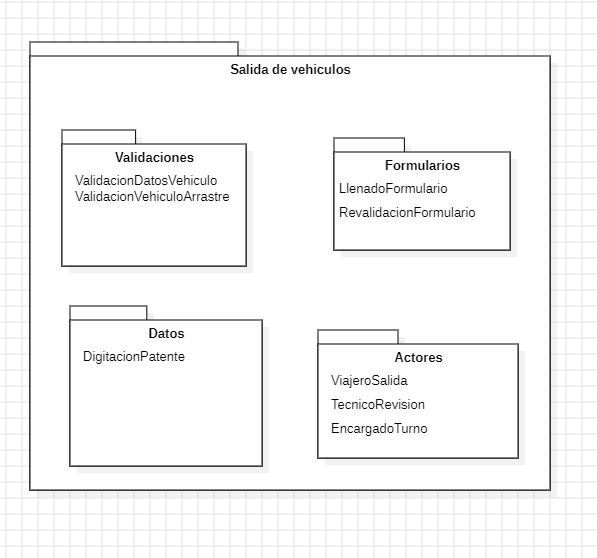
El diagrama de componentes adjunto ilustra la arquitectura modular y la interacción de los principales subsistemas que componen el sistema de gestión de salidas. Este diagrama se enfoca en cómo los componentes individuales ofrecen y requieren servicios a través de interfaces, mostrando una vista de alto nivel de la estructura física del sistema.

Se destacan cuatro componentes clave:

1. **Validar documentos:** Encargado de procesar y verificar la validez de los documentos.
2. **Soporte Adjunto:** Proporciona la funcionalidad para adjuntar y gestionar soportes, siendo un servicio crucial para la validación de documentos.
3. **Gestión Salida:** Componente central que coordina y valida los formularios de salida. Muestra una dependencia directa del componente "Validar documentos".
4. **Registro vehículos:** Responsable del registro de la información de los vehículos en el contexto de la salida.

El diagrama también representa la interacción de los actores externos con el sistema a través de las interfaces expuestas por estos componentes. Este diagrama permite comprender cómo las diferentes partes del sistema se conectan y colaboran para ejecutar el flujo de trabajo de la gestión de salidas.

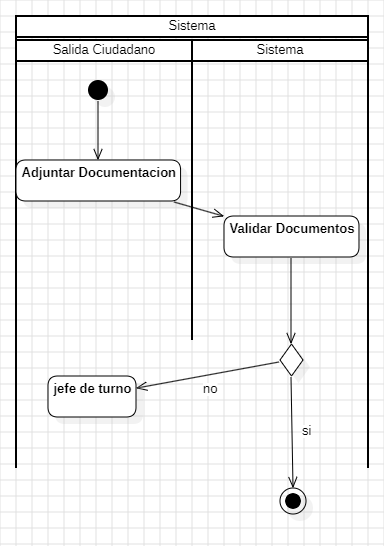
* + 1. **Diagrama de paquete**

****

* + 1. **Descripción diagrama de paquete**

El diagrama de paquetes organiza el sistema en capas lógicas que permiten una arquitectura modular, escalable y mantenible. Cada paquete agrupa elementos con responsabilidades similares, siguiendo principios de bajo acoplamiento y alta cohesión

* 1. **VISTA DE PROCESOS** (salida vehículo **o** entrada vehículo)
     1. Propósito
     2. Diagrama de actividad



* + 1. Descripción diagrama de actividad

### **Descripción del Diagrama de Actividad**

El diagrama de actividad modela el flujo de trabajo y la secuencia de acciones relacionadas con la "Salida Ciudadano" y la interacción con el "Sistema". Utiliza carriles de natación (swimlanes) para separar las responsabilidades entre el ciudadano y el Sistema.

El proceso se inicia con una actividad inicial y sigue los siguientes pasos:

1. **Adjuntar Documentación:** El proceso comienza con la acción del Ciudadano de adjuntar la documentación necesaria.
2. **Validar Documentos:** Una vez adjuntados, el Sistema toma el control para realizar la validación de dichos documentos.
3. **Decisión de Validación:** Después de la validación, se presenta un punto de decisión:
   * Si los documentos no son válidos, el proceso se dirige al "jefe de turno", implicando una intervención o escalada.
   * Si los documentos son válidos, el proceso finaliza exitosamente.

Este diagrama es fundamental para comprender la lógica de negocio y el orden de las operaciones, identificando las responsabilidades de cada parte involucrada y los posibles caminos a seguir según el resultado de una decisión.

### **Propósito del Diagrama de Actividad**

El propósito principal de este diagrama de actividad es visualizar y modelar el flujo de trabajo secuencial del proceso de "Salida Ciudadano" y la subsiguiente validación de documentos.

Sirve para:

* **Clarificar los pasos:** Muestra de forma clara las actividades que se deben realizar y en qué orden.
* **Identificar responsabilidades:** Asigna las actividades a los "carriles" (Ciudadano y sistema), dejando claro quién realiza qué acción.
* **Representar la lógica de decisión:** Muestra el punto de decisión sobre la validez de los documentos y los caminos alternativos (aprobación/escalada).
* **Facilitar la comprensión:** Ayuda a las partes interesadas (desarrolladores, analistas de negocio, usuarios) a entender rápidamente cómo se ejecuta este proceso específico.

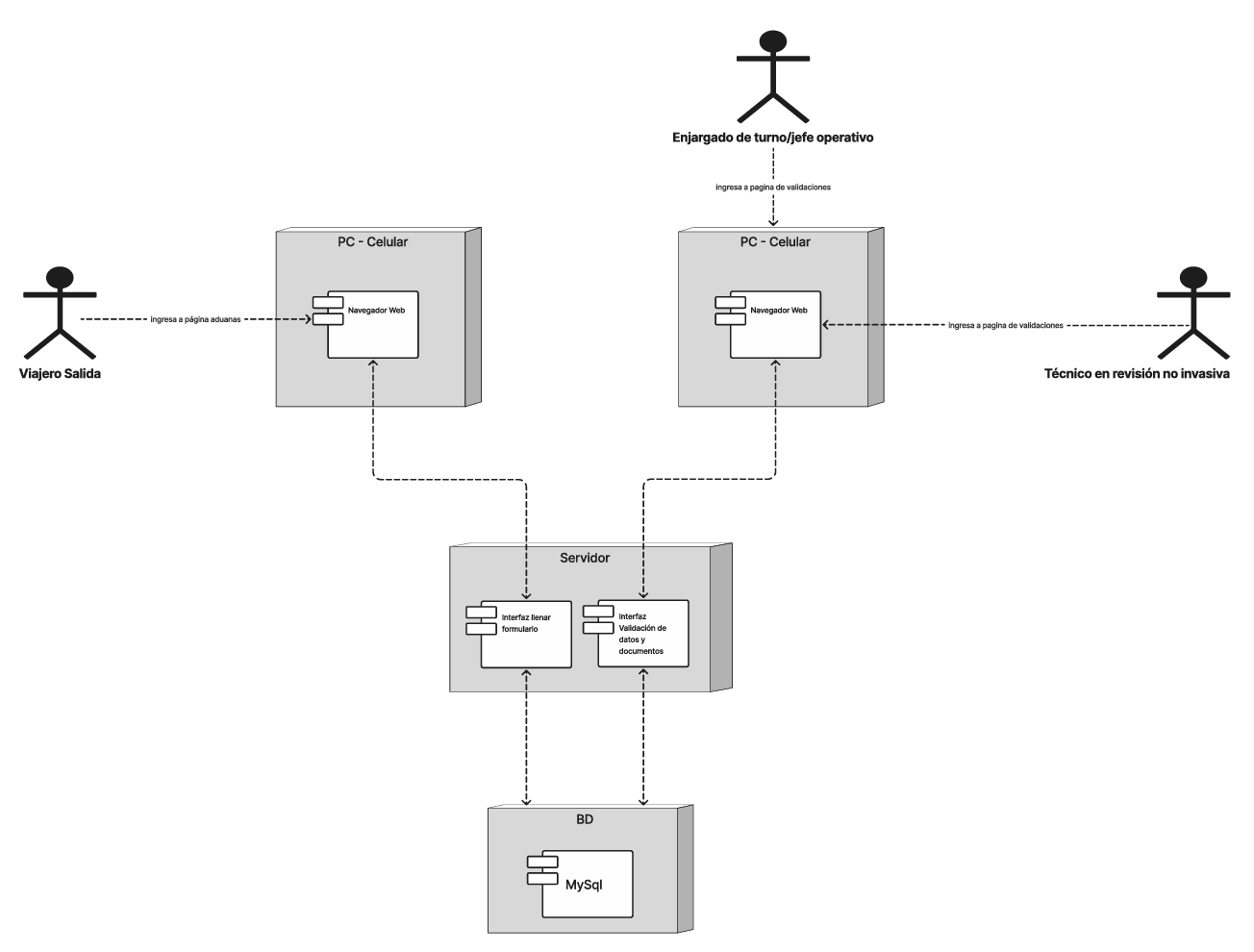
* 1. **VISTA FÍSICA**
     1. **Propósito**

Esta vista describe cómo el sistema se distribuye físicamente en el entorno tecnológico, identificando los dispositivos, servidores y componentes de red que permiten la ejecución de la aplicación. Es esencial para comprender la infraestructura necesaria para poner en funcionamiento el sistema en condiciones reales.

* + 1. **Diagrama de despliegue**

**Diagrama de Despliegue del Sistema de Salida de Vehículos**

*Este diagrama representa los nodos físicos involucrados en la ejecución del sistema, incluyendo dispositivos cliente, el servidor central y la base de datos. También muestra cómo interactúan los distintos actores (viajero, técnico, jefe operativo) con el sistema a través de navegadores web.*



* + 1. **Descripción diagrama de despliegue**

El sistema está desplegado en una arquitectura cliente-servidor tradicional. Los actores interactúan mediante navegadores web desde dispositivos como PC o celulares:

* El **viajero salida** accede al sistema mediante un navegador para llenar el formulario y adjuntar documentos desde cualquier dispositivo con conexión.
* El **técnico en revisión no invasiva** y el **jefe operativo** acceden también a través de navegadores, pero utilizando una interfaz especializada para validación y administración.

Ambos tipos de usuarios se conectan a un **servidor central**, que aloja la lógica de aplicación y gestiona la conexión a una base de datos relacional **MySQL**, donde se almacenan los datos del trámite, documentos adjuntos y registros de validación.

1. **REQUISITOS DE CALIDAD**
   1. **Propósito**

Esta sección tiene como propósito establecer los **atributos de calidad clave** que debe cumplir el sistema para ser considerado exitoso desde el punto de vista técnico y funcional. Estos requisitos no están ligados a funcionalidades específicas, pero afectan la experiencia del usuario, el mantenimiento del sistema y su fiabilidad operativa.

* 1. **Atributos de calidad**

| **Atributo de Calidad** | **Descripción** | **Justificación** |
| --- | --- | --- |
| **Disponibilidad** | El sistema debe estar disponible 24/7, incluso en horarios de alta demanda. | Los pasos fronterizos operan de forma continua, por lo que la disponibilidad del sistema es crítica para no entorpecer el flujo vehicular. |
| **Rendimiento** | Las operaciones del sistema deben ejecutarse en tiempos mínimos, especialmente las validaciones. | Si el sistema es lento, se generarán filas y demoras, anulando el propósito del proyecto. |
| **Seguridad** | Protección de los datos personales y documentos, con autenticación reforzada según el rol. | Se manejan documentos sensibles (padrón, seguros, poderes notariales), lo que requiere garantizar la confidencialidad. |
| **Usabilidad** | La interfaz debe ser clara, simple y apta para usuarios con distintos niveles de experiencia. | Los viajeros no siempre tienen conocimientos técnicos, por lo que la usabilidad es clave para asegurar adopción. |
| **Escalabilidad** | El sistema debe poder adaptarse a un mayor volumen de usuarios sin rediseño estructural. | En fechas de alto tráfico (feriados, vacaciones), la demanda crece y debe ser soportada eficientemente. |
| **Interoperabilidad** | El sistema debe comunicarse correctamente con servicios externos (Registro Civil, PDI, aseguradoras). | La validación automatizada depende de la capacidad del sistema para integrar y consumir servicios externos. |
| **Portabilidad** | Acceso desde distintos dispositivos y sistemas operativos sin necesidad de instalar software adicional. | Permite a los usuarios completar el trámite desde cualquier lugar o dispositivo disponible. |
| **Mantenibilidad** | El sistema debe estar organizado en componentes que permitan fácil actualización y mejora. | La evolución del sistema y la corrección de errores serán más eficientes si hay buena mantenibilidad interna. |

* 1. **Reglas y criterios de evaluación de calidad**

| **Atributo de Calidad** | **Criterio de Evaluación** | **Método / Herramienta de Validación** |
| --- | --- | --- |
| **Disponibilidad** | El sistema debe estar disponible el 99% del tiempo. | Monitoreo continuo del servidor y logs de uptime. |
| **Rendimiento** | Tiempo de respuesta ≤ 2 segundos para validaciones críticas. | Pruebas de carga con herramientas como JMeter o Postman. |
| **Seguridad** | Uso de HTTPS, contraseñas encriptadas, MFA para agentes. | Auditoría de seguridad, revisión de configuración SSL y pruebas con OWASP ZAP. |
| **Usabilidad** | Puntuación mínima de 70/100 en test heurístico; navegación sin errores críticos. | Evaluación con guía de Nielsen y test con usuarios reales (pruebas de tarea). |
| **Escalabilidad** | Soporte de al menos 2x usuarios estimados sin degradación. | Simulación de carga progresiva y observación de métricas de consumo. |
| **Interoperabilidad** | Conexión estable y sin errores con al menos 3 servicios externos clave. | Pruebas de integración API con mocks y entorno real de prueba. |
| **Portabilidad** | Acceso funcional y visual correcto en Chrome, Firefox, Android e iOS. | Pruebas en distintos navegadores y dispositivos. |
| **Mantenibilidad** | Cumplimiento del patrón MVC, código documentado, modularidad respetada. | Revisión interna del repositorio, análisis estático del código y checklist de buenas prácticas. |

1. **PRINCIPIOS DE DISEÑO APLICADOS**

* 1. **Propósito**

Esta sección presenta los principios de diseño de software aplicados en la construcción del sistema, con el objetivo de asegurar la **claridad estructural, mantenibilidad, escalabilidad y usabilidad** del producto. Estos principios guían las decisiones arquitectónicas y de codificación para cumplir con los objetivos de calidad establecidos.

* 1. **Principios de diseño**

| **Principio** | **Descripción** | **Aplicación en el sistema** |
| --- | --- | --- |
| **Abstracción** | Permite modelar entidades complejas usando representaciones simples que ocultan detalles internos. | La clase FormularioSalida representa de forma simple todos los documentos y datos del trámite, sin exponer la lógica de validación interna. |
| **Encapsulación** | Protege el estado interno de un objeto, permitiendo el acceso solo a través de métodos definidos. | Las clases como Vehiculo o Documento exponen solo las operaciones necesarias, evitando manipulación directa de atributos. |
| **Cohesión** | Mide el grado en que los elementos de un módulo están relacionados entre sí. | Cada clase tiene una función específica y no mezcla responsabilidades (ej. TecnicoRevision no gestiona formularios ni usuarios). |
| **Acoplamiento** | Mide la dependencia entre módulos o clases; idealmente debe ser bajo. | Las clases como Validacion y FormularioSalida se comunican mediante interfaces bien definidas, reduciendo dependencias directas. |
| **Modularidad** | Divide el sistema en unidades independientes que pueden ser desarrolladas y mantenidas separadamente. | El sistema está organizado en módulos como autenticación, validación, y reportes, lo que facilita su evolución. |
| **Descomposición** | Divide tareas complejas en partes manejables. | El proceso de salida de vehículo se separa en etapas como declaración, adjunto de documentos, y validación técnica. |

1. **PRINCIPIOS DE DISEÑO APLICADOS** 
   1. Propósito
   2. Principios de diseño (por ejemplo: abstracción, acoplamiento, cohesión, encapsulamiento, modularidad)

| **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 |
|  |  |  |
|  |  |  |
|  |  |  |

1. **PROTOTIPO**
   1. Propósito
   2. Mockups (imágenes con una breve descripción)
   3. Justificar herramientas de prototipado
2. **EVALUACIÓN DE CALIDAD HEURÍSTICA DE NIELSEN**
   1. Propósito
   2. Lista de verificación
   3. Análisis y métricas de resultados
3. **CONTROL DE VERSIONES**
   1. Propósito
   2. Control de versión utilizado (justificar el tipo de control de versión utilizad (fecha, semántica o secuencial)
   3. Justificar herramientas de versionamiento

1. **CONCLUSIONES**

El diseño del sistema propuesto para la gestión de la salida de vehículos en pasos fronterizos terrestres busca resolver un problema concreto: **la congestión y los largos tiempos de espera** que enfrentan los viajeros al momento de cruzar la frontera.

Para ello, se ha construido una arquitectura basada en una **aplicación web monolítica**, organizada internamente bajo el patrón **Modelo–Vista–Controlador (MVC)**. Esta elección permite una implementación más simple y coherente con el nivel de complejidad del proyecto, sin dejar de lado principios sólidos de diseño como **modularidad, cohesión, encapsulamiento y bajo acoplamiento**.

El sistema digitaliza el proceso aduanero, permitiendo a los viajeros registrar con anticipación su formulario de salida, adjuntar los documentos necesarios y conocer el estado de su trámite. Por su parte, el **técnico en revisión no invasiva** y el **jefe operativo** cuentan con interfaces diseñadas para simplificar sus tareas y facilitar el control eficiente de los trámites.

Además, el sistema está preparado para **interoperar con servicios externos** (como Registro Civil, PDI y aseguradoras), lo cual refuerza su confiabilidad y automatización.

El diseño incluye principios de **usabilidad, accesibilidad y diseño centrado en el usuario**, permitiendo que cualquier persona —independientemente de su nivel técnico— pueda completar el proceso sin fricciones.

Se espera que la implementación del sistema contribuya a una **reducción significativa de los tiempos de espera** en frontera, a una **mejor fiscalización de los trámites vehiculares**, y a una **mayor trazabilidad y control estadístico** por parte del Servicio Nacional de Aduanas.

1. **BIBLIOGRAFÍA**