**[Sistema de aduanas**

**(**Salida de Vehículos**)]**

**(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** |
| --- | --- | --- | --- |
| 19-05-2025 | 1.0 |  | Augustin Caceres |
| 20-05-2025 | 1.1 |  | Gustavo Alvial |
| 21-05-2025 | 1.2 |  | Matias Turra |
| 22-05-2025 | 1.3 |  | Dante Rodriguez |

**Tabla de Contenidos**

[1.1. Contexto del Problema 4](#_bjtdk89kh32)

[1.2. Propósito 4](#_dy7ya180xckh)

[1.3. Ámbito 4](#_f2s53rs5jglh)

[1.4. Definiciones, acrónimos y abreviaciones 4](#_ytmrjvg2letn)

[1.5. Resumen ejecutivo 4](#_u89gt8m9ic6q)

[2.1. Descripción general del sistema 5](#_n5lpx1pik6uq)

[2.2. Objetivos del sistema 5](#_rrlupyri4afv)

[2.3. Principales funcionalidades esperadas 5](#_l3pqnomy3hqw)

[2.4. Supuestos y dependencias (Lo que necesita para funcionar y lo que se supone para ello) 5](#_mbuaepegq31o)

[**3.1. Estilo arquitectónico adoptado 6**](#_bxv2fnur8xia)

[3.2. Justificación del estilo según el contexto del sistema 6](#_8k15mkobhiwo)

[**3.3. Patrones de diseño aplicados 6**](#_7bgbrhy8e1ke)

[*4.1.1. Propósito 7*](#_rvbp8tv1q4it)

[*4.1.2. Actores (General) 7*](#_9u30nthpf8u2)

[*4.1.3. Diagrama general de casos de uso (General) 7*](#_qp6kqx465a48)

[*4.1.4. Diagrama de casos de uso específicos (salida vehículo) 8*](#_hjiick59ymxa)

[*4.1.5. Lista de casos de uso (salida vehículo o entrada vehículo) 8*](#_7tpugopq3wdv)

[*4.1.6. Especificación de casos de uso (UN caso de uso principal de la salida vehículo/entrada vehículo) 8*](#_3qeikcjkp86b)

[*4.2.1. Propósito 11*](#_j9kqb0kcbgxo)

[*4.2.2. Diagrama de clases 11*](#_m2atojn03yjy)

[*4.2.3. Descripción diagrama de clases 11*](#_f8cweynmqr8o)

[*4.3.1. Propósito 12*](#_f3epz4o54rl)

[*4.3.2. Diagrama de componente 12*](#_ny6arc1li302)

[*4.3.3. Descripción diagrama de componente 12*](#_5g63bpi70twf)

[*4.3.4. Diagrama de paquete 13*](#_i4tqup5rw78l)

[*4.3.5. Descripción diagrama de paquete 13*](#_ozhcyajr6oqw)

[4.4.1. Propósito 14](#_d5hl2rgnynfp)

[4.4.2. Diagrama de actividad 14](#_i8p7kwdizwdz)

[4.4.3. Descripción diagrama de actividad 15](#_ipp0c53smnuq)

[*4.5.1. Propósito 15*](#_qcwnn7esjk7r)

[**4.5.2. Diagrama de despliegue 15**](#_iwejyhnhot84)

[*4.5.3. Descripción diagrama de despliegue 16*](#_7gokj79xks29)

[5.1. Propósito 17](#_nniiqpoda9xt)

[**5.2. Atributos de calidad (por ejemplo: Usabilidad, Accesibilidad (WCAG), Rendimiento, Mantenibilidad, Seguridad Portabilidad) 17**](#_sltjpjpvfew)

[*5.3. Reglas y criterios de evaluación de calidad 18*](#_bq0gm1n3g0c0)

[*6.1. Propósito 19*](#_mugzthfuvkj)

[**6.2. Principios de diseño 20**](#_lodj8xnm59sr)

1. INTRODUCCIÓN

# Contexto del Problema

El Servicio Nacional de Aduanas enfrenta serias dificultades operativas en los pasos fronterizos terrestres, en la frontera con Argentina. Las esperas para cruzar pueden extenderse entre 8 a 20 horas, sobre todo del lado chileno. Esto se ha originado por distintas razones como el incremento del flujo de personas y vehículos, escasa infraestructura, desconocimiento de los trámites digitalizados y procesos ineficientes. Estos problemas generan congestión, afectan el servicio, desincentivan a los usuarios querer cruzar por el paso fronterizo y dificultan la fiscalización efectiva

# Propósito

Desarrollar un sistema informático que modernice y automatice los procesos aduaneros, reduciendo los tiempos de espera y mejorando la fiscalización. Este sistema busca facilitar el cumplimiento normativo, entregar un servicio eficiente, garantizar la seguridad, optimizar la gestión de datos y ofrecer una experiencia más fluida tanto para los usuarios como para los funcionarios

# Ámbito

# Definiciones, acrónimos y abreviaciones

| **ACRÓNIMO** | **DESCRIPCIÓN** |
| --- | --- |
| **PDI** | Policía de investigaciones |
| **SAG** | Servicio agrícola y ganadero |
| **UML** | Lenguaje de modelado estándar que permite construir y visualizar distintos sistemas de software |

# Resumen ejecutivo

El Servicio Nacional de Aduanas enfrenta importantes demoras en pasos fronterizos terrestres como el Complejo Los Libertadores, con esperas que superan las 8 horas. Estas se deben al alto flujo de personas, infraestructura insuficiente y procesos manuales poco eficientes.

Para abordar esta situación, se propone el desarrollo de un sistema informático que automatice trámites clave, integre información con países limítrofes, reduzca los tiempos de espera y mejore la fiscalización. La solución busca modernizar la gestión aduanera, facilitar el comercio exterior y entregar un servicio más ágil, seguro y eficiente.

# 

1. VISIÓN DEL SISTEMA

# Descripción general del sistema

El sistema que proponemos está destinado a modernizar los procesos en los pasos aduaneros terrestres de Chile. Permitirá automatizar la documentación requerida para el ingreso y salida de personas, vehículos, alimentos, mascotas, animales, etc, reduciendo tiempos de espera y mejorando la eficiencia del control fronterizo.

Se integrará con la PDI y el SAG, proporcionando una interfaz intuitiva y funcionalidades específicas para cada tipo de usuario.

A largo plazo, se busca que el sistema pueda posicionarse como una herramienta clave para impulsar y facilitar el comercio exterior, mejorando los estándares de servicio público en la frontera y así mejorando la calidad de atención

# Objetivos del sistema

* Reducir los tiempos de espera en los pasos fronterizos terrestres.
* Automatizar la recopilación y validación de documentación obligatoria.
* Integrar los sistemas de información entre Chile y países cercanos.
* Apoyar la fiscalización inteligente mediante la gestión de riesgos.
* Mejorar el control y seguridad de los procesos aduaneros.
* Generar reportes estadísticos sobre el ingreso y egreso de personas y vehículos.
* Facilitar el cumplimiento de las normas por parte de los usuarios.

# Principales funcionalidades esperadas

* Formulario digital para la autorización de salida de menores.
* Registro y validación de cuentas de usuario con control de acceso seguro.
* Trámite online para la salida y admisión temporal de vehículos particulares
* Declaración jurada digital para ingreso de alimentos, mascotas u otros bienes

regulados

* Integración con bases de datos de PDI, SAG y Aduanas extranjeras
* Interfaz web amigable
* Cumplimiento de estándares de seguridad y confidencialidad de datos.

# Supuestos y dependencias (Lo que necesita para funcionar y lo que se supone para ello)

**Supuestos:**

* Se supone que la Aduana cuenta con servidores, redes y computadores para soportar el nuevo sistema.
* Se supone que existe conectividad a internet estable en los pasos
* fronterizos.
* Se supone que los usuarios tienen acceso a dispositivos digitales para interactuar con el sistema.
* Suponemos que los países limítrofes colaborarán en el intercambio de datos para la integración de sistemas.
* No habrá resistencia significativa a la adopción de procesos digitales por parte de los viajeros.

**Dependencias:**

* Requiere personal técnico especializado para el desarrollo, mantenimiento y soporte del sistema.
* El proyecto está sujeto a la asignación de fondos que se le establezca.
* Depende de la aprobación de leyes o decretos que autoricen la digitalización de trámites
* La implementación en pasos fronterizos requiere hardware adicional.

1. **ESTILOS Y PATRONES ARQUITECTÓNICOS**

# Estilo arquitectónico adoptado

Para el diseño de nuestro sistema nos decidimos por una arquitectura basada en microservicios y organizada en capas. Esto nos permitirá poder estructurar la solución como un conjunto de servicios pequeños, independientes y especializados de manera bien organizada.

# Justificación del estilo según el contexto del sistema

Nos hemos decidido por una arquitectura basada en microservicios, ya que esta permitirá que el sistema sea escalable y mantenible. Esta arquitectura nos facilita desarrollar, probar, desplegar y actualizar funcionalidades de forma independiente, lo cual es fundamental para el sistema de aduanas, considerando que deberá integrarse con diversas entidades y sistemas externos. Y en cuanto a la organización por capas, proporciona una separación de responsabilidades, lo que nos facilita el desarrollo y mejora nuestra organización

# Patrones de diseño aplicados

Hemos optado por aplicar el patrón de diseño MVC en el desarrollo del módulo de Autorización de Salida, ya que permite una clara separación de responsabilidades entre la interfaz del usuario (Vista), la lógica de negocio (Modelo) y el flujo (Controlador). Esta estructura facilita el mantenimiento, permite reutilizar componentes y mejora la organización interna del sistema. Además, es compatible con la arquitectura en capas y microservicios utilizada, lo que asegura un desarrollo más limpio y escalable.

1. MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS
   1. **VISTA DE ESCENARIO**

# Propósito

El diagrama de caso de uso general tiene como objetivo principal representar de forma visual y clara las interacciones entre los actores y las funcionalidades clave que ofrecerá el sistema aduanero.

# Actores (General)

Los actores que podemos encontrar dentro del diagrama general serian los siguientes :

1.-Viajero

2.-SAG

3.-Personal de aduanas

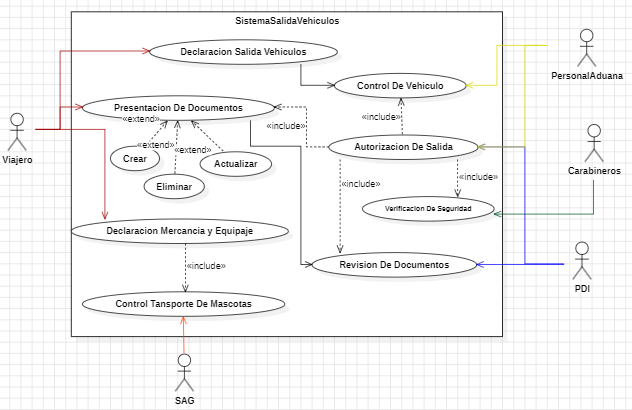
4.-Carabinero

5.-PDI

# Diagrama general de casos de uso (General)

# 

# Diagrama de casos de uso específicos (salida vehículo)



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

| **Código** | **Nombre** | **Actores** |
| --- | --- | --- |
| CU-001-001 | Presentación de documentos | Viajero |
| CU-002-002 | Declaración de salida de vehículos | Viajero |
| CU-003-003 | Declaración de mercancía y equipaje | Viajero |
| CU-004-004 | Revisión de documentos | PDI |
| CU-005-005 | Verificación de seguridad | Carabineros |
| CU-006-006 | Control de transporte de mascotas | SAG |
| CU-007-007 | Control de vehículos | Pers. de Aduana |
| CU-008-008 | Autorización de Salida | Pers. de Aduana/PDI |

# Especificación de casos de uso (**UN** caso de uso principal de la salida vehículo/entrada vehículo)

| **Caso de Uso** | Autorización de Salida | **Identificador:**  CU-008-008 |
| --- | --- | --- |
| **Actores** | Personal de aduana, PDI | |
| **Tipo** | Caso de uso primario | |
| **Referencias** | * Verificar Documentos del Conductor * Evaluar Declaración de Vehículo * Revisar Productos Declarados | |
| **Precondición** | * La Declaración de Salida ha sido creada y completada por el conductor. * El vehículo y la persona están registrados en el sistema. | |
| **Postcondición** | * La salida del vehículo queda registrada como autorizada o rechazada. | |
| **Descripción** | Permite validar y autorizar la salida temporal de un vehículo y su conductor, según la documentación presentada y los controles realizado | |
| **Resumen** | El funcionario de aduana verifica la documentación del conductor y del vehículo, válida que se cumplan las condiciones legales y obtiene respuestas para autorizar o rechazar la salida | |

**CURSO NORMAL**

| **Nro.** | **Ejecutor** | **Paso o Actividad** |
| --- | --- | --- |
| 1 | Funcionario de Aduanas | Ingresa al sistema y accede a la declaración pendiente. |
| 2 | Sistema | Muestra los datos del vehículo, conductor y productos declarados |
| 3 | Funcionario | Verifica que el vehículo cumple las condiciones para salir. |
| 4 | Sistema (Mediante API) | Consulta a PDI para validar identidad del conductor. |
| 5 | Sistema (Mediante API) | Consulta a SAG si los productos requieren permiso especial |
| 6 | Funcionario | Evalúa el estado de los permisos y aprueba la salida |
| 7 | Sistema | Genera documento de autorización y registra el evento |
| 1. El funcionario accede al sistema y busca la declaración de salida. 2. El sistema muestra los datos del vehículo, conductor y productos declarados. 3. Se solicita validación de identidad del conductor a la PDI. 4. Se solicita validación de productos a declarar al SAG. 5. Se consulta a Aduana sobre requisitos fiscales/documentales. 6. El sistema recibe las respuestas y muestra el estado de cada revisión. 7. El funcionario revisa los resultados y aprueba o rechaza la salida. 8. El sistema registra la decisión y actualiza el estado de la declaración. | | |

**CURSO ALTERNATIVO**

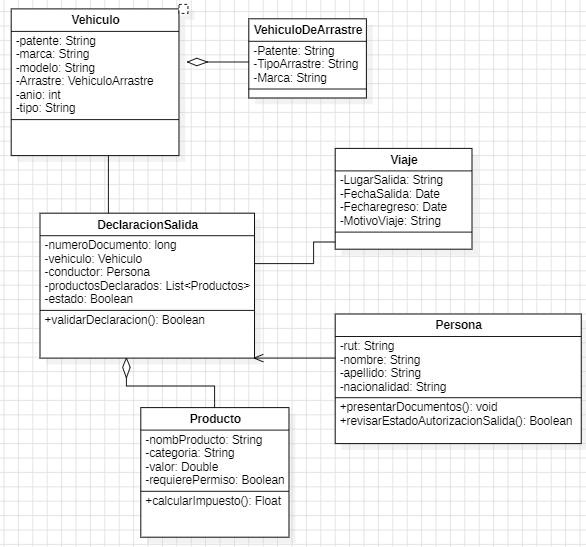
| **Nro.** | **Descripción de acciones alternas** |
| --- | --- |
| 1 | Si el funcionario no logra ingresar al sistema (por error de autenticación o problema técnico) se muestra un mensaje de error |
| 2 | Si la declaración no está disponible (por número incorrecto o aún no enviada por el usuario) el sistema informa que no hay datos para procesar |
| 3 | Si el vehículo no está registrado correctamente o falta documentación técnica, el funcionario solicita al usuario corregir los datos |
| 4 | Si la verificación de identidad del conductor falla ( en RUT/pasaporte o antecedentes) el sistema bloquea la autorización y genera una alerta |
| 5 | Si el SAG indica que los productos declarados requieren permiso especial y este no está adjunto, el sistema informa al funcionario y no permite continuar hasta que el permiso correspondiente sea ingresado |
| 6 | Si alguno de los controles anteriores no está aprobado, el funcionario no puede autorizar la salida |
| 7 | Si el sistema presenta error al generar el documento de autorización (por fallo técnico o de conectividad), se guarda un respaldo de la validación previa y se permite reintentar la emisión más tarde |

* 1. VISTA LÓGICA (salida vehículo)

# Propósito

El diagrama de clases tiene como objetivo modelar la estructura del sistema, representando las clases y entidades involucradas, sus atributos, relaciones y responsabilidades. Además este diagrama nos permite establecer relaciones entre objetos y facilitar el diseño e implementación del sistema

# Diagrama de clases



# Descripción diagrama de clases

El diagrama de clases representa los elementos del proceso de salida temporal de vehículos por el paso aduanero. En el centro del modelo se encuentra la clase DeclaracionSalida, que vincula a un Vehículo y a un Conductor. Además, esta clase incluye los productos que el viajero desea declarar, representados por la clase Producto.

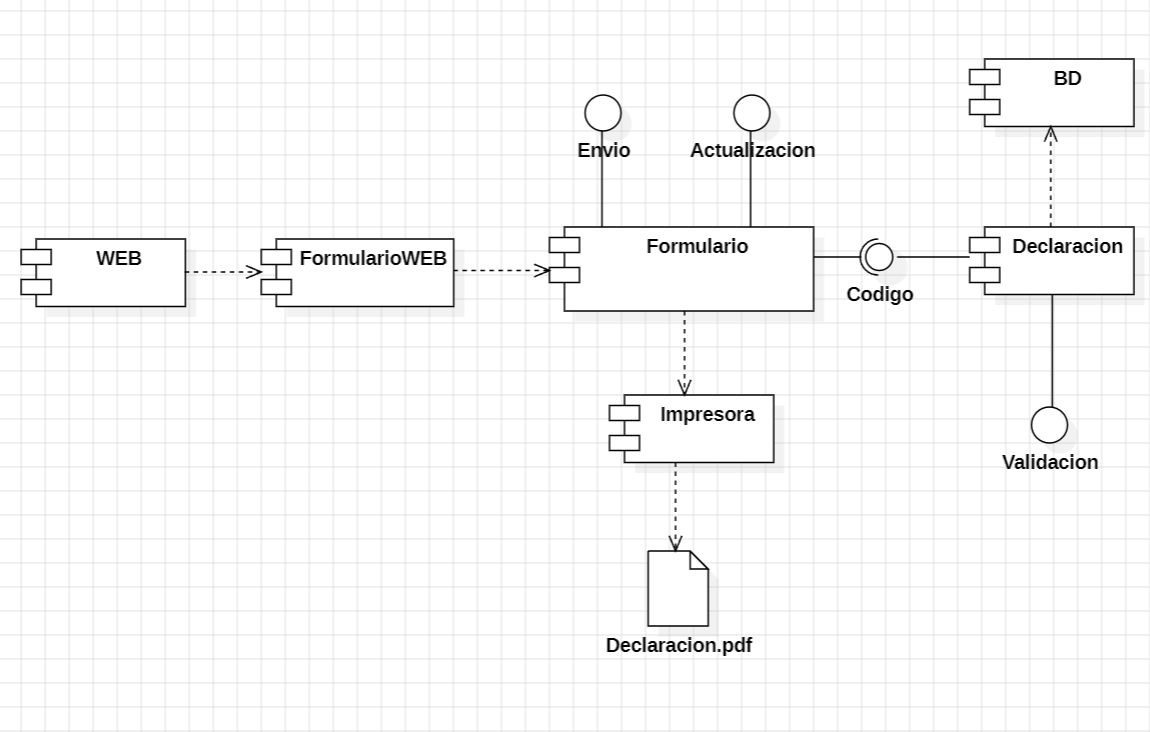
La revisión y validación de la salida se gestiona a través de la clase ControlFronterizo, que interactúa con los organismos encargados como la Aduana, PDI y SAG. Estos servicios se representan mediante interfaces que permiten verificar identidades, autorizaciones, documentación y productos.

* 1. **VISTA DE IMPLEMENTACIÓN/DESARROLLO** (salida vehículo)

# Propósito

El propósito de la vista de implementación es mostrar el sistema desde el punto de vista del programador y encargarse de la gestión del software. Esto permite facilitar la resolución de problemas complejos, aumentar la eficiencia y optimizar las operaciones empresariales.

# Diagrama de componente



# Descripción diagrama de componente

diagrama de componentes, su objetivo es ofrecer una vista física y de alto nivel del sistema. Este se desglosa en componentes que sirven para construir el sistema; dichos componentes son autónomos y encapsulados, lo que permite su reutilización y reemplazo de forma sencilla.

explicación:

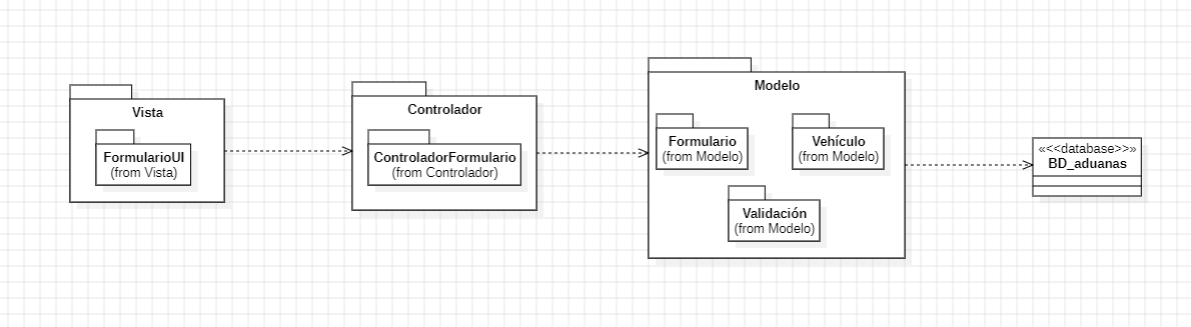
El usuario interactúa con WEB la cual representa el entorno que utiliza el componente FormularioWEB, este se encarga de gestionar el formulario desde la web, también este interactúa con Formulario, componente principal, que maneja las acciones principales.

El Formulario puede:

* Enviarse (Envío)
* Actualizarse (Actualización)
* Imprimirse → Impresora → Declaración.pdf
* Validar o consultar declaración mediante código → Declaración

Declaración se encarga de validar, guardar o buscar los datos en la BD.

# Diagrama de paquete



# Descripción diagrama de paquete

El diagrama de paquete busca mostrar la organización lógica y modular del sistema, mostrando cómo las responsabilidades están separadas según el patrón de diseño MVC.

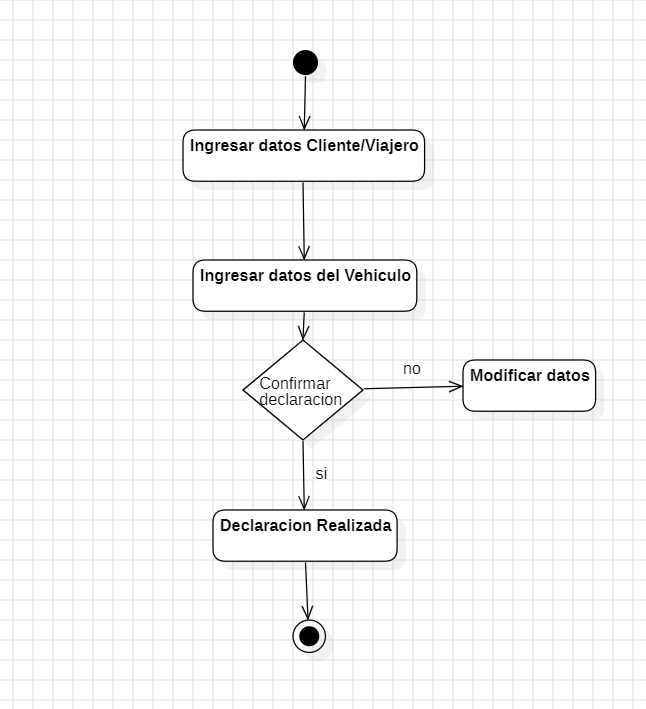
El paquete de vista contiene la interfaz gráfica del sistema, responsable de la interacción directa con el usuario, el cuál solo se comunica con el controlador. Este último actúa como intermediario entre la vista y el modelo, recibiendo eventos desde la interfaz y ejecutando la lógica correspondiente invocando las distintas funcionalidades del modelo, el cuál incluye la lógica y las entidades del sistema teniendo una dependencia directa hacia una base de datos externa. Por último se encuentra la base de datos del sistema, que no forma parte del paquete modelo, sino que es una dependencia utilizada para la persistencia de los datos.

* 1. **VISTA DE PROCESOS** (salida vehículo)

# Propósito

El propósito del diagrama de actividad es mostrar el flujo de trabajo dentro de un sistema, ayudando a entender lo que ocurre, cuando ocurre y las decisiones que se toman durante su ejecución.

# Diagrama de actividad



# Descripción diagrama de actividad

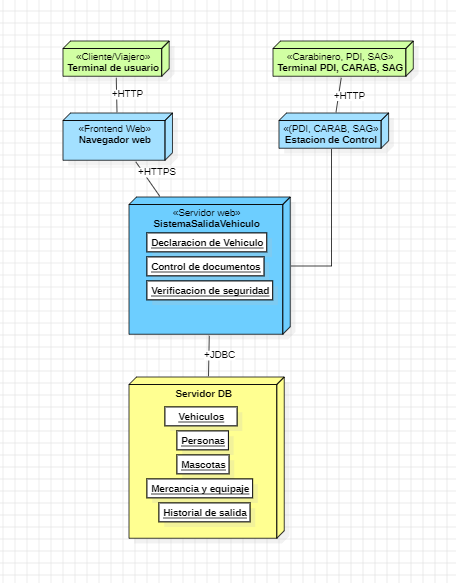
Representa el flujo de acciones que debe seguir el Cliente/Viajero dentro del sistema para declarar la salida de su vehículo. En la que el usuario debe primero ingresar sus datos personales para posteriormente ingresar los datos de su vehículo, al confirmar sus declaraciones la declaración será realizada y confirmada por la estación de control. en caso de que el usuario no confirme sus datos, será redirigido a la modificación de datos donde el usuario podrá editar su información hasta que la confirmación de sus declaraciones sea aceptada

* 1. **VISTA FÍSICA** (salida vehículo)

# Propósito

El propósito de este diagrama es mostrar cómo se distribuyen de forma física los componentes dentro del sistema. Mostrando una vista física simplificada de la infraestructura del sistema, mostrando que hace cada parte del sistema, donde se encuentran y cómo interactúan entre sí.

# Diagrama de despliegue



# Descripción diagrama de despliegue

El propósito de este diagrama es mostrar la estructura física del sistema, mostrando cómo se distribuyen los componentes del sistema sobre diferentes nodos de hardware y el cómo se comunican entre sí.

Nodos de usuario: está conformado por la terminal de usuario y la terminal de PDI, SAG y CARAB, donde en la terminal de usuario el cliente accede por medio de una terminal o dispositivo publico (computadoras de aduanas, etc) para realizar su declaración de salida de vehículo, registrar su equipaje y mercancía y poder presentar sus documentos.

Mientras que en las estaciones de control, administradas por PDI, SAG Y CARAB, se encargan de validar la información ingresada por el cliente/viajero y realizar los controles pertinentes como la revisión de documentos, verificación de seguridad y las declaraciones de vehículo.

Nodo de Servidor web: dentro de este nodo se ingresan y se procesan los datos dentro del sistema para que una vez estén validados puedan ser almacenados y guardados dentro de la base de datos. Dentro de este nodo se encuentran 3 módulos que son:

Modulo Declaracion de vehículo: permite que el cliente/viajero registre los datos de su vehículo antes de salir

Módulo Control de documentos: Su función es revisar que los documentos tanto del vehículo como del cliente/viajero sean correctos y verídicos, es usado por el personal (PDI, SAG, CARAB).

Módulo Verificacion de seguridad: Verifica si existen alertas de antecedentes o información que impida la salida del vehículo:

Nodo Base de datos: en este nodo se almacena toda la información importante que se registra dentro del sistema dentro de una conexión segura que sería JDBC, la información importante sería:

-Vehículos: datos de los vehículos que se han registrado

-Personas: información del cliente/viajero

-Mascotas, Equipaje y mercancía: Declaraciones dadas por el cliente/viajero

-Historial de salida: Registro y autorizaciones dadas por la estación de control.

1. **REQUISITOS DE CALIDAD** (General)

# Propósito

Los requisitos de calidad sirven para que el software no sólo funcione, sino que también sea fácil de usar, accesible para todos, y que no dé problemas más adelante. Ayudan a que el diseño esté bien organizado y que las partes del sistema trabajen bien juntas sin complicarse.

También hacen que el programa sea más cómodo para el usuario, con una interfaz clara y pensada para quien lo va a usar. Siguiendo ciertas reglas y usando prototipos, se puede mejorar el software antes de terminarlo.

En resumen, los requisitos de calidad ayudan a crear un software útil, entendible y que dure más tiempo sin dar tantos dolores de cabeza.

# Atributos de calidad (por ejemplo: Usabilidad, Accesibilidad (WCAG), Rendimiento, Mantenibilidad, Seguridad Portabilidad)

| **ATRIBUTO DE CALIDAD** | **DESCRIPCIÓN** | **JUSTIFICACIÓN** |
| --- | --- | --- |
| Usabilidad | La usabilidad es la facilidad con la que un usuario puede aprender a usar el sistema, interactuar con él y lograr lo que necesita al ocuparlo, por lo tanto se busca un sistema intuitivo, eficiente y satisfactorio para la experiencia | Si un usuario no logra entender el uso del sistema por mucho que este pueda funcionar correctamente, no lograra servir para su cometido |
| Accesibilidad | Este atributo es importante porque tiene en consideración a las personas con diferentes capacidades para que puedan ocupar el software, entonces al diseñar un sistema accesible se asegura un beneficio no solo al usuario si no también a la empresa, Este atributo está regido bajo los estándares WGAC | Un software accesible permite que muchas personas sin importar de su capacidad puedan ocuparlo sin problemas, no es solo una obligación legal sino también un método para ampliar el público objetivo |
| Mantenibilidad | Busca facilidad al momento de efectuar cambios en el software cuando sea necesario en momentos como una mala organización o un desorden completo dentro del software, un software bien diseñado que tiene en mente la mantenibilidad permite una rápida corrección de errores y actualizaciones | Todo sistema al avanzar el tiempo necesita ser actualizado o corregido para poder mantenerse funcional y dentro del mercado, cualquier falla en este atributo puede generar que el software sea costoso y complicado de mantener |
| Rendimiento | El rendimiento busca que el software tenga rapidez y eficiencia al momento de ejecutar sus funciones, especialmente en momentos específicos como manejar una gran cantidad de usuario, lo que se busca es que el software cumplir un tiempo de respuesta específico sin ningún contratiempo, o que los recursos puedan manejar esta carga sin degradarse | Este atributo es importante porque al tener en cuenta la experiencia del usuario, si este al interactuar se encuentra frente a un software lento o que le consume varios recursos, la experiencia termina siendo negativa, más en estos tiempos donde el público está acostumbrado a aplicaciones con una alta velocidad de respuesta y amigable con sus recursos |

# Reglas y criterios de evaluación de calidad

Usabilidad: Para este atributo la forma en que se medirá el criterio de aceptación sobre el software será mediante pruebas reales con los usuarios bajo el criterio de jakob nielsen y su sistema de puntuación, para poder observar su interacción con el sistema, se busca permitir que el usuario pueda fallar y que el error sea reversible

por lo tanto como criterio más específico tomemos al usuario y que al momento de este ir rellenando el formulario, se equivoque en ingresar algún dato, el sistema necesita revertir ese error si el usuario lo desea sin necesidad de caerse o volver entrar a la página, también se buscará medir el tiempo de demora de los usuarios al completar cierta tarea y que el sistema también pueda mostrar un animaciones de carga para demostrar de forma visual del estado, por ejemplo que no necesite más de 2 segundos para saber en donde se encuentran los trámites que el busque, otro método que ocupamos para evaluar la usabilidad sería mediante encuestas para saber la experiencia del usuario y si el sistema cumple con lo que propone, por último bajo el punto 2 de nielsen que el sistema sea consistente con el mundo real, por lo tanto haríamos que la realización de trámites sea lo más parecido a como lo harías en persona quitando las grandes tiempos de espera para que el usuario pueda familiarizarse de forma mas rapida

Para poder medir el cumplimiento de cada punto ocuparemos las siguientes herramientas, como la antes mencionada evaluación heurística de Jakob Nielsen, pruebas de carga, pruebas de usabilidad y validaciones de usuario con su retroalimentación hecha mediante las encuestas

Accesibilidad: la norma para medir el cumplimiento de este atributo principalmente es el cumplimiento del criterio WCAG de los distintos niveles como por ejemplo:

Orientación de la pantalla (AA) Es necesario que tanto los sitios web como las aplicaciones soporten la orientación vertical y horizontal para facilitar la accesibilidad de usuarios con discapacidad que usan para la navegación en redes, dispositivos montados en soportes. Cancelación del puntero (A) Para evitar la activación accidental de contenidos por parte del usuario con discapacidad, este criterio establece qué eventos específicos tienen que ser activados solo cuando el usuario vuelva a mover el puntero. Además tiene que estar siempre la posibilidad de volver de manera fácil a los contenidos que se querían visualizar.

Los métodos para evaluar que esto se cumpla será con la interacción de los usuarios con el sistema y con la lista de chequeos de la WCAG

Mantenibilidad: Se evalúa según qué tan fácil es de entender, modificar y extender el código, también midiendo la complejidad del código, su modularidad, la duplicación y la calidad de documentación, por último considerar el tiempo en el que toma arreglar errores o agregar nuevas funcionalidades.

herramientas o métodos que ocuparemos para medir si se cumple el criterio serian análisis estático de código ocupando herramientas como SonarQube, también revisando el código entre programadores haciendo un índice de mantenibilidad y cobertura de pruebas

Rendimiento: Se medirá mediante tiempos de carga, que al momento de acceder al trámite no te demores más de 2 segundos en ingresar, también evaluando el uso de la cpu, memoria y la red, siguiendo con pruebas de carga y estrés para asegurar un funcionamiento correcto bajo situaciones de alta demanda

las herramientas que ocuparemos para evaluar esto bajo métricas como tiempo de carga inicial, tiempo de respuesta a eventos del usuario, consumo de recursos bajo un tiempo prolongado, para las pruebas de rendimiento se ocuparía LightHouse herramienta de google que permite hacer análisis del rendimiento web, siguiendo con Jmeter aplicación de java que permite hacer pruebas de carga

1. PRINCIPIOS DE DISEÑO APLICADOS

# Propósito

Los principios de diseño tienen como propósito facilitar la creación de software estructurado, comprensible, mantenible y escalable. Aplicarlos correctamente desde las primeras etapas del desarrollo permite construir sistemas más fáciles de modificar, con menor riesgo de errores y con mejor rendimiento a largo plazo.

# 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 | Se ocultan todo lo que son detalles menores de un objeto, mostrando solo lo esencial | El sistema oculta la complejidad interna del sistema, mostrando solo lo necesario como botones. |
| Acoplamiento | Es el grado de dependencia entre módulos y clases | Los módulos del sistema funcionan de forma independiente, comunicándose por interfaces |
| Encapsulamiento | Se ocultan los detalles internos de un objeto, se permite que el acceso a sus datos sean a través metodos publico, protegiendo de manera segura los datos del usuario. | El sistema protege la información interna ingresada por el usuario |
| Modularidad | El sistema se divide en distintos módulos, donde cada uno tiene una función específica. | el sistema está dividido en módulos separados, facilitando mejoras |

# 

1. CONCLUSIONES

En conclusión, la propuesta busca modernizar y optimizar los procesos de aduana, específicamente el trámite de salida de Chile, mediante un sistema sólido y coherente que aborde la problemática de congestión fronteriza, demoras excesivas y trámites manuales innecesarios. Para ello, proponemos una arquitectura moderna y escalable basada en microservicios y el patrón MVC.

Asimismo, se plantea la automatización de validaciones, declaraciones y autorizaciones, junto con la integración de sistemas como la PDI y el SAG, con el fin de lograr una fiscalización coordinada. Esta solución considera atributos de calidad clave para garantizar un sistema confiable, manejable e intuitivo.

Finalmente, el uso del modelo 4+1 permite visualizar el sistema desde múltiples perspectivas lógica, física, de desarrollo y de procesos, facilitando su comprensión integral y su correcta implementación.

1. BIBLIOGRAFÍA

Teacup Lab. (2023, 24 abril). *Los 10 principios heurísticos de Nielsen explicados con ejemplos*. Teacup Lab.<https://www.teacuplab.com/es/blog/los-10-principios-heuristicos-de-nielsen-explicados-con-ejemplos/>

Torres Burriel Estudio. (2023, 15 febrero). *WCAG 2.1: Qué son y cómo respetarlas*. Torresburriel Estudio.<https://torresburriel.com/weblog/wcag-2-1-que-son-y-como-respetarlas/>

Google. (s.f.). *Lighthouse*. Chrome for Developers.<https://developer.chrome.com/docs/lighthouse?hl=es-419>

Apache Software Foundation. (s.f.). *Apache JMeter™*.<https://jmeter.apache.org/>