**SIGA**

**(DAS) Documento Arquitectura de Software**

**Versión 1.0**

**Identificación de Documento**

| **Identificación** | Equipo 7 |
| --- | --- |
| **Proyecto** | Proyecto de Aduanas “SIGA” |
| **Versión** | v.13 |

| **Documento mantenido por** |  |
| --- | --- |
| **Fecha de última revisión** | 21/06/2025 |
| **Fecha de próxima revisión** |  |

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

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
|  | v.1 | comienzo del documento | Renata Barrientos |
|  | v.2 | Primera inserción de los datos necesarios (Modelo 4+1),  Requisitos de calidad, Principio de diseño | Jordan Diaz, Darien Mora, Renata barrientos,  Damian Gonzalez |
|  | v.3 | Prototipado | Jordan Diaz, Darien Mora, Renata barrientos,  Damian Gonzalez |
|  |  |  |  |

**Tabla de Contenidos**

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

[1.1.](#_heading=h.mp8vu41wtfu6) Contexto del Problema (General) 4

[1.2.](#_heading=h.alc4abqr2y15) Propósito 4

[1.3.](#_heading=h.2ahrk6e8lc68) Ámbito 4

[1.4.](#_heading=h.uoom778oqasi) Definiciones, acrónimos y abreviaciones 4

[1.5.](#_heading=h.1ckckze5yscr) Resumen ejecutivo (General) 4

[1.6.](#_heading=h.7fep40vkcfhs) Arquitectura del sistema (General) 4

[**2.**](#_heading=h.p8fyfs2xy8r6) **VISIÓN DEL SISTEMA (General) 4**

[2.1.](#_heading=h.tadlsb5w73j7) Descripción general del sistema 4

[2.2.](#_heading=h.vv1j2ks6mbqc) Objetivos del sistema 4

[2.3.](#_heading=h.r0bkphvfq9gu) Principales funcionalidades esperadas 4

[2.4.](#_heading=h.vz9zo3yq5bpe) Supuestos y dependencias 4

[**3.**](#_heading=h.b1ho41d61et3) **ESTILOS Y PATRONES ARQUITECTÓNICOS (General) 4**

[3.2.](#_heading=h.z6go7own4c9i) Justificación del estilo según el contexto del sistema 4

[**4.**](#_heading=h.tp7nlzutpf2b) **MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS 4**

[4.1.](#_heading=h.5rvxnt7tp971) VISTA DE ESCENARIO (General y salida vehículo o entrada vehículo) 4

[*4.1.1.*](#_heading=h.x7fq4k5pwb0p) *Propósito (General) 4*

[*4.1.2.*](#_heading=h.th56oxgmveb4) *Actores (General) 4*

[*4.1.3.*](#_heading=h.jaw2s1kyilxe) *Diagrama general de casos de uso (General) 4*

[*4.1.4.*](#_heading=h.1v8v7hlkld1k) *Diagrama de casos de uso específicos (salida vehículo o entrada vehículo) 4*

[*4.1.6.*](#_heading=h.b48nwh2xftlc) *Especificación de casos de uso (UN caso de uso principal de la salida vehículo/entrada vehículo) 5*

[4.2.](#_heading=h.qmh9lkkximqn) VISTA LÓGICA (salida vehículo o entrada vehículo) 6

[*4.2.1.*](#_heading=h.s03ppcs9zd3h) *Propósito 6*

[*4.2.2.*](#_heading=h.3jap3h4fqpgw) *Diagrama de clases 6*

[*4.2.3.*](#_heading=h.hnu49bai0oy) *Descripción diagrama de clases 6*

[4.3.](#_heading=h.z8sld214ph6t) VISTA DE IMPLEMENTACIÓN/DESARROLLO (salida vehículo o entrada vehículo) 7

[*4.3.1.*](#_heading=h.1nqma5b2vjxp) *Propósito 7*

[*4.3.2.*](#_heading=h.ybk4b8rcn1rs) *Diagrama de componente 7*

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

[*4.3.4.*](#_heading=h.iznlnwbz7xab) *Diagrama de paquete 7*

[*4.3.5.*](#_heading=h.v5v25vuf5i18) *Descripción diagrama de paquete 7*

[4.4.](#_heading=h.6nle3hn3w266) VISTA DE PROCESOS (salida vehículo o entrada vehículo) 7

[4.4.1.](#_heading=h.miylj15h9fnm) Propósito 7

[4.4.2.](#_heading=h.mr03hyr16kh9) Diagrama de actividad 7

[4.4.3.](#_heading=h.ywxv2vkajuzr) Descripción diagrama de actividad 7

[4.5.](#_heading=h.3g0l3zada5y7) VISTA FÍSICA (salida vehículo o entrada vehículo) 7

[*4.5.1.*](#_heading=h.fws1useg0f48) *Propósito 7*

[*4.5.2.*](#_heading=h.diigzbi0ve29) *Diagrama de despliegue 7*

[*4.5.3.*](#_heading=h.uq09mxfdh2q2) *Descripción diagrama de despliegue 7*

[5.](#_heading=h.su7p9bm4uhlq) REQUISITOS DE CALIDAD (General) 7

[5.1.](#_heading=h.o0t9kkm32i7u) Propósito 7

[*5.3.*](#_heading=h.liel7g17yx14) *Reglas y criterios de evaluación de calidad 7*

[**6.**](#_heading=h.dcqww11upk2h) **PRINCIPIOS DE DISEÑO APLICADOS** 8

[*6.1.*](#_heading=h.djvbd8edpqga) *Propósito 8*

[6.2.](#_heading=h.dtmyiuauo4iw) Principios de diseño (por ejemplo: abstracción, acoplamiento, cohesión, encapsulamiento, modularidad) 8

[**7.**](#_heading=h.425hubc00p8g) **PROTOTIPO 8**

[7.1.](#_heading=h.pklrinvjwefk) Propósito 8

[7.2.](#_heading=h.glmvjdbw1xjw) Mockups (imágenes con una breve descripción) 8

[7.3.](#_heading=h.5m0k30ralcll) Justificar herramientas de prototipado 8

[**8.**](#_heading=h.zhl1ehmv6vkv) **EVALUACIÓN DE CALIDAD HEURÍSTICA DE NIELSEN** 8

[8.1.](#_heading=h.ow30nk7wbkv7) Propósito 8

[8.2.](#_heading=h.5z28kxeb8d2q) Lista de verificación 8

[8.3.](#_heading=h.48wb2uug0jqb) Análisis y métricas de resultados 8

[**9.**](#_heading=h.eq8d7tassrbj) **CONTROL DE VERSIONES** 8

[9.1.](#_heading=h.9keqd171a8ua) Propósito 8

[9.2.](#_heading=h.u7s79p54t490) Control de versión utilizado (justificar el tipo de control de versión utilizad (fecha, semántica o secuencial) 8

[9.3.](#_heading=h.tks9oktns66a) Justificar herramientas de versionamiento 8

[**7.**](#_heading=h.8wgswoirzcv1) **CONCLUSIONES 8**

[**8.**](#_heading=h.ntz7v9v7y06o) **BIBLIOGRAFÍA 8**

1. **INTRODUCCIÓN**
   1. Contexto del Problema

“ La Aduana en Chile se encarga de fiscalizar, facilitar y controlar el ingreso y salida de mercancías, asegurando el cumplimiento de las leyes tributarias, comerciales y sanitarias. Además, recauda impuestos aduaneros, combate el contrabando

y protege al país del ingreso de productos peligrosos o ilegales. “

* Documento “La Aduana en Chile”.
  1. Propósito

El propósito de este documento es definir, detallar y verificar que los vehículos que entran y salen no estén transportando productos peligrosos y/o ilegales tanto dentro como fuera de Chile. Este documento sirve como guía, mantención e implementación del sistema para garantizar una mejora en el servicio de aduanas para garantizar un correcto funcionamiento de los pasos migratorios, optimizando los procesos y con ello la agilización de los diversos trámites de migración.

* 1. Ámbito

**Nombre del Sistema:** Sistema Integrado de Gestión Aduanera o SIGA.

El SIGA será una plataforma de software diseñada para optimizar y gestionar los procesos clave del Servicio Nacional de Aduanas de Chile, tal como se describe en el caso de negocio. El SIGA abarca las funciones clave del proceso de aduana incluyendo el registro de mercancías, la gestión de la documentación, los procesos de inspección, el control de pagos de impuestos y aranceles, y los trámites específicos para el ingreso temporal de vehículos y mascotas.

* 1. Definiciones, acrónimos y abreviaciones

| **ACRÓNIMO** | **DESCRIPCIÓN** |
| --- | --- |
| *SIGA* | Sistema Integral de Gestión Aduanera |
| *SNA* | Servicio Nacional de Aduanas |
| *LogIn* | Proceso de ingresar a una cuenta o sistema |
| *API* | Interfaz de programación de aplicaciones |

* 1. Resumen ejecutivo

Se requiere desarrollar un sistema web de gestión general en torno a las actuales funciones del SNA de Chile. Este sistema se conformará de módulos con objetivos específicos a detallar en este documento.

* 1. Arquitectura del sistema

Se utilizará una estructura modular, implementando microservicios autogestionados. La separación de funciones agiliza los procesos de manejo y localización de datos, disminuye los tiempos de espera, y aislará posibles errores focalizados para que estos no afecten funciones paralelas.

(ej. vista de escenario, vista lógica, vista de desarrollo, vista de proceso, vista física)

1. **VISIÓN DEL SISTEMA**
   1. Descripción general del sistema

El Proceso de Aduanas tiene el objetivo de agilizar y optimizar sus procesos, considerando las horas de demora en gestión a comparación con países limítrofes. En este proyecto, el enfoque es desarrollar sistemas de gestión que apoyen esta visión.

Algunos factores o problemáticas que puedan afectar a nuestro producto y a sus requisitos son:

**Demanda y competencia del mercado:** El saber cuál es el nivel actual de demanda, tendencias y el saber sí otros servicios son eficientes o rentables a un costo menor afectando a la calidad de nuestro servicio.

**Cambios en las necesidades del usuario:** Las necesidades del usuario o cliente están en constante cambio pudiendo demandar nuevas funcionalidades.

* 1. Objetivos del sistema

El producto SIGA, será un sistema de software centralizado que permitirá a los funcionarios de aduanas gestionar de manera eficiente los diversos trámites y regulaciones asociados al comercio exterior y al ingreso de personas y bienes al país, tal como se describe en el caso de negocio.

* 1. Principales funcionalidades esperadas

**Control de Ingreso:** Administrar la documentación necesaria para autorizar el ingreso de personas al país y almacenar cada instancia en formato de informe.

**Declaración de Importación:** General informe con cada instancia de especies que requieran constatación de ingreso.

**Ingreso temporal Vehículos:** En el caso de ingreso terrestre Argentina-Chile, existe la instancia de ingreso de vehículos autorizados temporalmente. Se debe mantener una documentación de dichas autorizaciones.

**Ingreso de mascotas:** Administra datos de la declaración jurada que autoriza el ingreso de mascotas al país.

**Control de Egreso:** Administrar la documentación necesaria para autorizar la salida de personas del país y almacenar cada instancia en formato de informe.

**Salida temporal de vehículos:** En el caso de ingreso terrestre Chile-Argentina, existe la instancia de salida del país de vehículos autorizados temporalmente. Se debe mantener una documentación de dichas autorizaciones.

* 1. Supuestos y dependencias

**Suposiciones:**

**Infraestructura tecnológica:** Se supone que el Servicio de Aduanas cuenta con la infraestructura tecnológica básica necesaria para la implementación y operación del SIGA, incluyendo servidores, redes y estaciones de trabajo compatibles.

**Disponibilidad de personal:** Se asume la disponibilidad de personal del Servicio de Aduanas para participar en las fases de análisis, pruebas y capacitación del SIGA.

**Marco legal y regulatorio:** Se presupone que el marco legal y regulatorio aduanero actual se mantendrá relativamente estable durante el desarrollo del proyecto. Cambios significativos en las leyes o regulaciones podrían requerir modificaciones en los requisitos del SIGA.

**Integración con sistemas existentes:** Se asume la disponibilidad de documentación y la colaboración de los responsables de los sistemas existentes del Servicio de Aduanas para lograr la integración con el SIGA, en caso de ser necesaria.

**Conocimiento del negocio:** Se presupone que el equipo de desarrollo tendrá acceso a expertos del Servicio de Aduanas con el conocimiento necesario del proceso aduanero para definir y validar los requisitos del SIGA.

**Estándares y protocolos:** Se asume que los estándares y protocolos de comunicación necesarios para interactuar con sistemas externos (como los de países limítrofes) serán estables y estarán documentados.

**Dependencias:**

**Disponibilidad de información de sistemas externos:** El correcto funcionamiento de ciertas funcionalidades del SIGA (como la validación de información con sistemas de países limítrofes) dependerá de la disponibilidad y la correcta operación de esos sistemas externos.

**Colaboración de terceros:** La integración con otros sistemas o la adopción de nuevos protocolos podría depender de la colaboración y la disposición de terceros (proveedores de software, autoridades de otros países) para proporcionar la información o realizar las adaptaciones necesarias.

**Aprobaciones y decisiones internas:** El avance del proyecto SIGA dependerá de la obtención oportuna de las aprobaciones y las decisiones por parte de las autoridades competentes dentro del Servicio de Aduanas.

**Suministro de hardware y software:** La implementación del SIGA dependerá del suministro oportuno y la correcta instalación del hardware y software necesarios.

**Capacitación de usuarios:** La adopción exitosa del SIGA dependerá de la planificación y la ejecución efectiva de la capacitación de los usuarios finales (funcionarios de aduanas).

1. **ESTILOS Y PATRONES ARQUITECTÓNICOS** 
   1. Estilo arquitectónico adoptado

Se utilizará el modelo de arquitectura de vistas “4+1”. Este modelo implica la implementación de las vistas: lógica, desarrollo, procesos, física

* 1. Justificación del estilo según el contexto del sistema

Se utilizará el modelo de arquitectura de vistas “4+1”, lo que implica la implementación de las siguientes vistas: lógica, desarrollo, procesos, física, y una vista adicional unificadora: escenarios.

Este modelo entrega una visualización clara e integral de las facetas del sistema, gráfica la relación entre ellas. Un sistema claro y detallado de modelamiento facilita la construcción de la aplicación.

| VISTAS | PERSPECTIVA | DIAGRAMA |
| --- | --- | --- |
| lógica | usuario | clases |
| desarrollo | desarrollador | componentes  de paquete |
| procesos | integradores | actividad  procesos |
| fisica | ingenieros de sistema | despliegue |
| escenarios | usuario | caso de uso |

* 1. Patrones de diseño aplicados (ej. patrón MVC, repositorio, etc.)

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

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

El propósito de la vista lógica es representar la estructura funcional interna del sistema, mostrando cómo se organizan las entidades principales.

* + 1. Actores

| Actores | Descripción |
| --- | --- |
| Usuario general | Todo actor que requiere de un LogIn para realizar acciones dentro del sistema. |
| Funcionario | Empleado registrado del SNA de Chile. |
| Ciudadano | Persona que requiera el trámite de salida terrestre del país. |

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





* + 1. Lista de casos de uso

| Requisitos funcionales | | |
| --- | --- | --- |
| R1 | Registrar usuario | usuario general |
| R2 | Completar formulario de salida | ciudadano |
| R3 | Constatación de productos no equipaje | ciudadano |
| R4 | Informe de flujo de vehículos | funcionario |

| Requisitos funcionales Específicos | | |
| --- | --- | --- |
| R1 | Registrar usuario | usuario general |
| R1.1 | El sistema debe registrar un usuario nuevo y guardar su información de cuenta |  |
| R1.2 | Un usuario existente debe poder ingresar identificándose con su RUT |  |
| R2 | Completar formulario de salida | ciudadano |
| R2.1 | Un ciudadano debe completar un formulario con la instancia de Propietario |  |
| R2.2 | Un ciudadano debe completar un formulario con la instancia de Vehículo |  |
| R2.3 | Un ciudadano debe completar un formulario con la instancia de Viaje |  |
| R2.4 | El sistema debe guardar los formularios enviados correctamente |  |
| R3 | Constatación de productos no equipaje | ciudadano |
| R3.1 | Un ciudadano debe completar una declaración de efectos (solicitud de salida) constatando productos no equipaje |  |
| R4 | Informe de flujo de vehículos | funcionario |
| R4.1 | El sistema debe generar un reporte con cada instancia de salida de vehículos |  |



| **Caso de Uso** | Completar formulario de salida | **Identificador:**  R2 |
| --- | --- | --- |
| **Actores** | Ciudadano | |
| **Tipo** | Primario | |
| **Referencias** | el ciudadano debe completar un formulario con la instancia de propietario.  un ciudadano debe completar un formulario con la instancia de vehículo.  un ciudadano debe completar un formulario con la instancia de viaje.  el sistema debe guardar los formularios enviados correctamente. | |
| **Precondición** | El usuario debe estar registrado y haber iniciado sesión en el sistema. | |
| **Postcondición** | El formulario de salida se guarda exitosamente en el sistema. | |
| **Descripción** | Un usuario registrado como ciudadano completa un formulario de salida para un vehículo que viaja al extranjero. | |
| **Resumen** | El ciudadano ingresa datos del propietario, vehículo y viaje, y el sistema guarda el formulario. | |

**CURSO NORMAL**

| **Nro.** | **Ejecutor** | **Paso o Actividad** |
| --- | --- | --- |
| 1 | ciudadano | Accede al formulario de salida del sistema |
| 2 | ciudadano | Completa lo datos del propietario. |
| 3 | ciudadano | Completa los datos del vehiculo |
| 4 | ciudadano | Completa los datos del viaje |
| 5 | ciudadano | envia el formulario |
| 6 | sistema | valida los datos |
| 7 | sistema | guarda el formulario enviado correctamente |
| 8 | sistema | muestra un mensaje de confirmación al ciudadano |
|  | | |
|  | | |

**CURSO ALTERNATIVO**

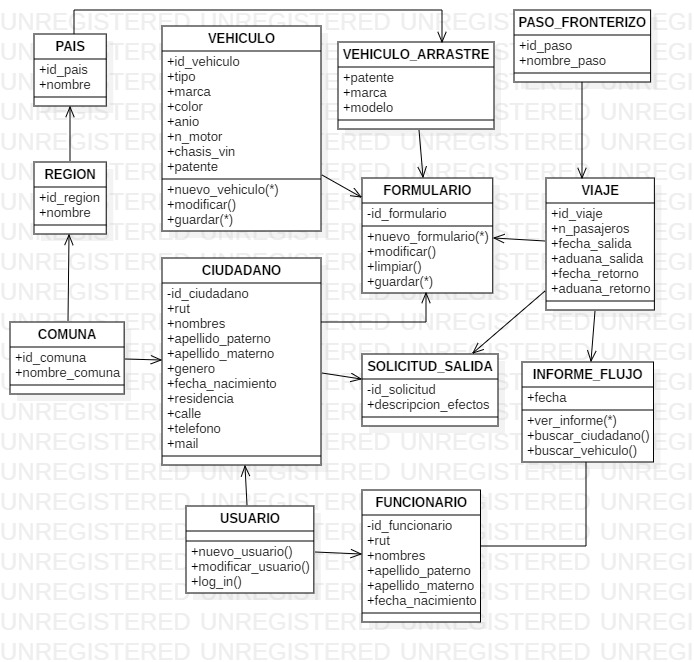
| **Nro.** | **Descripción de acciones alternas** |
| --- | --- |
| 6 | **Datos incorrectos:** Si el sistema detecta datos incorrectos se notifica al ciudadano sobre los errores y se le permite corregirlos. |
| 7 | **Error al guardar:** Si ocurre un error al guardar el formulario, el sistema informa al ciudadano y le sugiere reintentar o contactar soporte. |
|  | |

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

El propósito de la vista lógica es representar la estructura funcional interna del sistema muestra cómo se organizan las entidades principales.

El diagrama de clases muestra la relación entre elementos y usuarios del sistema. Da además un listado de los atributos de cada una de estas clases graficadas.

* + 1. Diagrama de clases



* + 1. Descripción diagrama de clases

Una clase Usuario, que garantiza el registro e inicio de sesión, dividido en dos subclases con atributos propios: Funcionario, Cliente.

Se organizan los datos de procedencia (ubicación, nacionalidad) de todas las clases que lo requieran con datos pre-existentes dentro de las clases: Comuna, Región, País, Paso Fronterizo.

Clases que tomarán datos de dos o más clases para generar documentos informe: Formulario; Solicitud Salida; Informe Flujo.

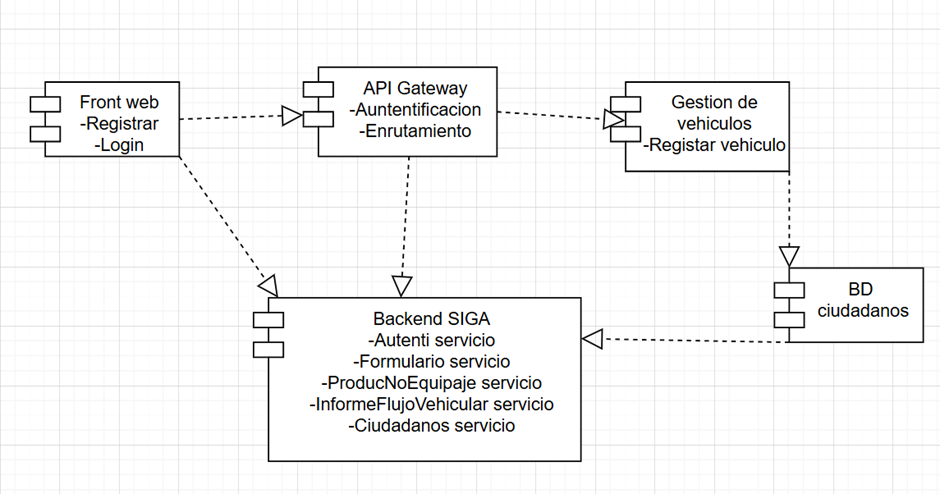
* 1. **VISTA DE IMPLEMENTACIÓN/DESARROLLO** 
     1. Propósito

El propósito de la vista de implementación/Desarrollo permite representar cómo se estructuran los componentes que gestionan la autentificación de los usuarios, el

manejo de formularios, la generación de informes y la lógica de negocio relacionada

con el control aduanero.

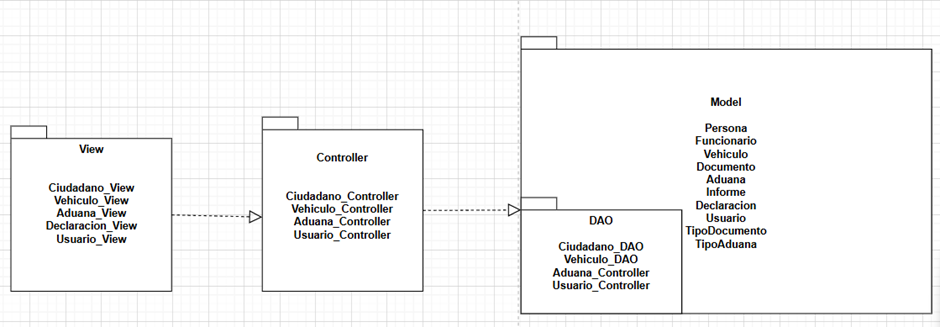
* + 1. Diagrama de componentes



* + 1. Descripción diagrama de componente

El diagrama de componentes muestra la interacción entre el "Front web" (Registro, Login), la "API Gateway" (Autentificación, Enrutamiento), el "Backend SIGA" (Autenticación, Formulario, Producto No Equipaje, Informe Flujo Vehicular, Ciudadanos), la "Gestión de Vehículos" (Registro Vehículo) y la "BD Ciudadanos".

* + 1. Diagrama de paquete



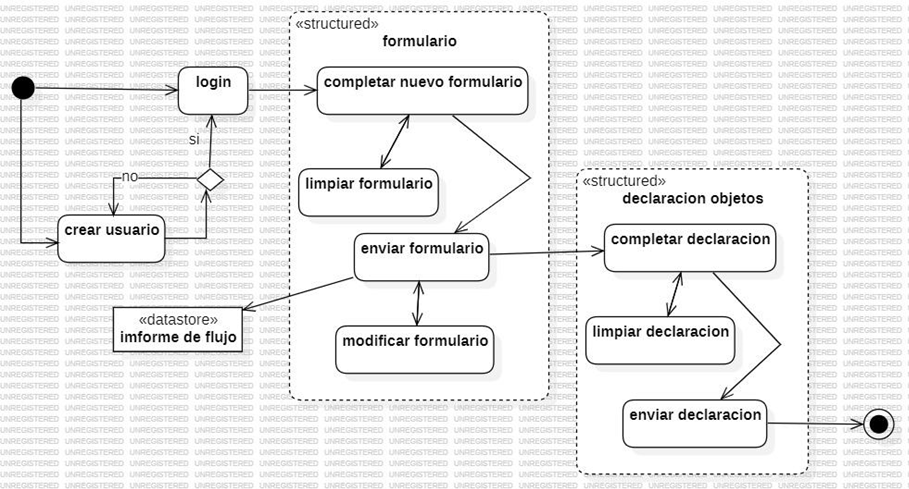
* + 1. Descripción diagrama de paquete

El diagrama de paquete muestra cómo se agrupan las clases según su responsabilidad y cómo se relacionan entre sí los paquetes, siguiendo una arquitectura MVC + DAO.

* 1. **VISTA DE PROCESOS**
     1. Propósito

El objetivo principal de este diagrama es mostrar cómo se agrupan las clases según su responsabilidad y cómo se relacionan entre sí los paquetes.Siguiendo una arquitectura MVC + DAO (Modelo–Vista–Controlador + Objeto de Acceso a Datos).

* + 2. Diagrama de actividad



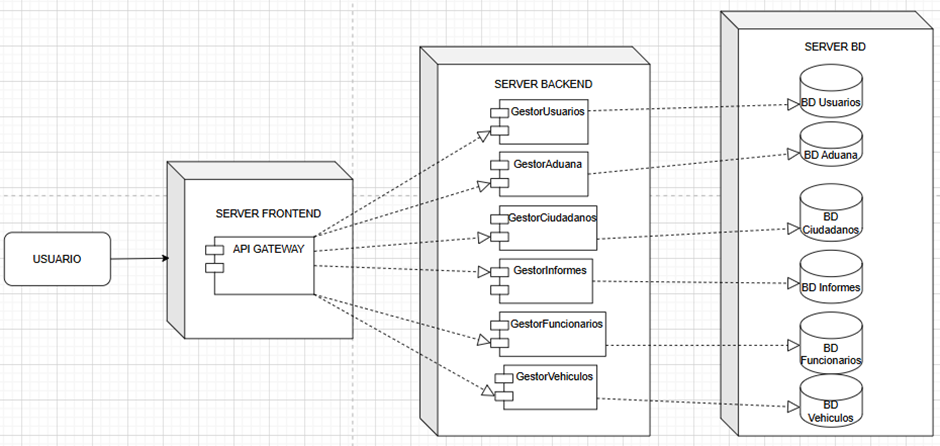
* + 1. Descripción diagrama de actividad

El diagrama de actividad representa el flujo de trabajo del sistema, relacionado con la gestión de formularios y declaraciones de objetos. Incluye el proceso de login, creación de usuario, completar y enviar formularios, modificar formularios.

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

El propósito de la vista física es representar cómo se implementan físicamente los componentes del sistema en el hardware.

* + 1. Diagrama de despliegue



* + 1. Descripción diagrama de despliegue

El diagrama de despliegue representa una arquitectura distribuida donde el usuario interactúa a través de una API Gateway. Esta API Gateway enruta las solicitudes a diferentes servicios de backend, y cada uno de estos servicios interactúa con su propia base de datos, proporcionando modularidad, escalabilidad y una clara separación de responsabilidades.

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

El propósito es definir los requisitos de calidad del sistema, asegurando que cumpla con los estándares esperados de rendimiento, usabilidad, seguridad y mantenibilidad.

* 1. Atributos de calidad

| **ATRIBUTO DE CALIDAD** | **DESCRIPCIÓN** | **JUSTIFICACIÓN** |
| --- | --- | --- |
| Usabilidad | Facilitar la navegación entre funciones | Una interfaz bien organizada mejora la experiencia de usuario, reduce errores y permite que los usuarios completen tareas más rápido y con menor frustración. |
| Rendimiento | Rapidez con la que procesa solicitudes el sistema | Asegura agilidad en el control aduanero, evitando demoras, congestión y frustración. |
| Seguridad | Encriptar datos en la base de datos | Proteger los datos sensibles o información personal de los usuarios asegurando que permanezcan ilegibles para los atacantes |
| Modularidad | Las tareas del sistema están divididas en instancias con tareas propias. | Evita la dependencia innecesaria entre tareas/elementos no relacionados. |

* 1. Reglas y criterios de evaluación de calidad

| **Atributo de calidad** | **Criterio a evaluar** | **Métodos y herramientas** |
| --- | --- | --- |
| Disponibilidad | 99.9% de tiempo de actividad en el mes medido como porcentaje de tiempo operativo | Monitoreo continuo de uptime como new relic, sistemas de alerta 24/7 |
| Rendimiento | Consultas/Solicitudes procesadas en menos de 3 segundos | Prueba de carga/estrés como con jmeter, monitoreo de aplicaciones para latencia |
| Usabilidad | Coherencia visual, navegación intuitiva, bajo nivel de errores de usuario. | Pruebas heurísticas (Evaluación de Nielsen), pruebas de usabilidad con usuarios reales, encuestas de satisfacción. |
| Escalabilidad | Capacidad para manejar el 70% de aumento en usuarios y datos sin degradación del rendimiento. | Pruebas de carga con escenarios de estrés, monitoreo de métricas de infraestructura (CPU, RAM, red). |
| Seguridad | Cumplimiento de políticas de acceso, detección y prevención de vulnerabilidades. | Auditorías de seguridad, pruebas de penetración (pentesting), escaneo de vulnerabilidades, revisiones de código. |
| Mantenibilidad | Modularidad del código, documentación clara, facilidad para implementar cambios. | Revisión de código, métricas de complejidad ciclomática, análisis de dependencias de módulos. |

1. **PRINCIPIOS DE DISEÑO APLICADOS**

* 1. Propósito

Asegurar y evidenciar la implementación de métodos de calidad y cumplimiento de requerimientos no funcionales.

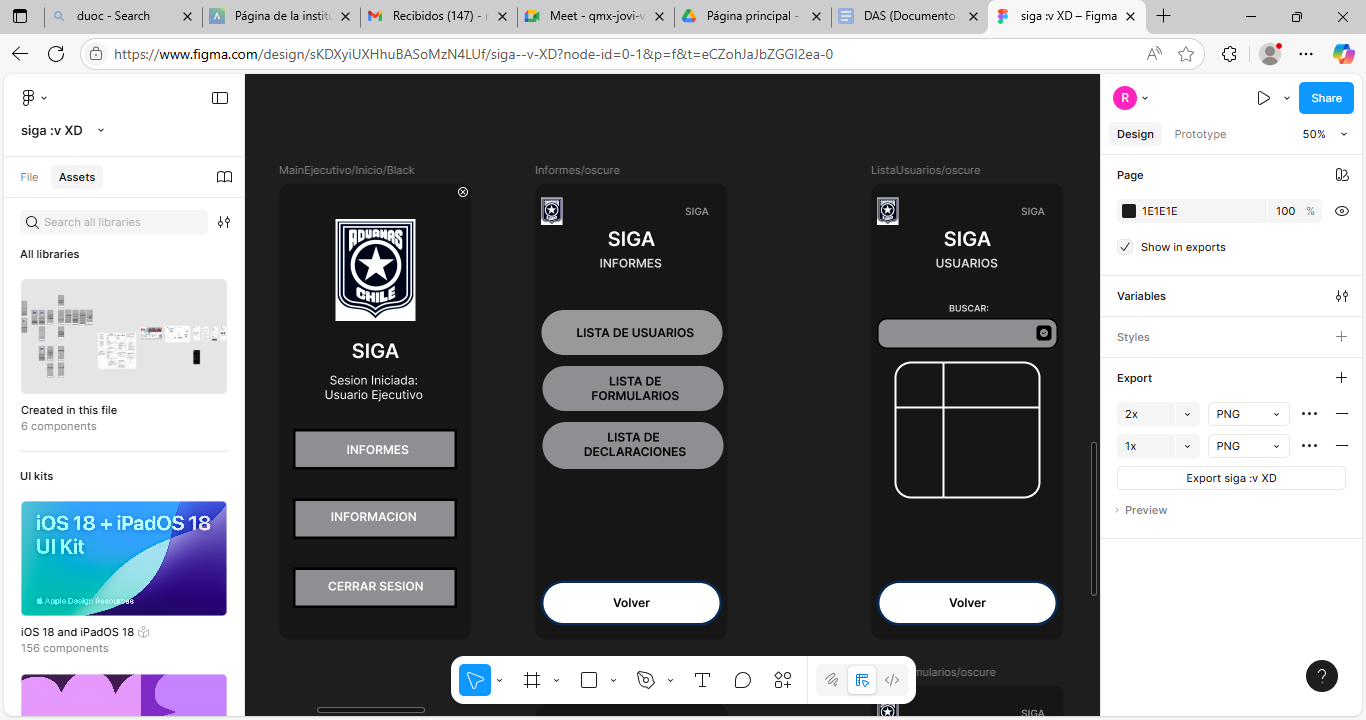
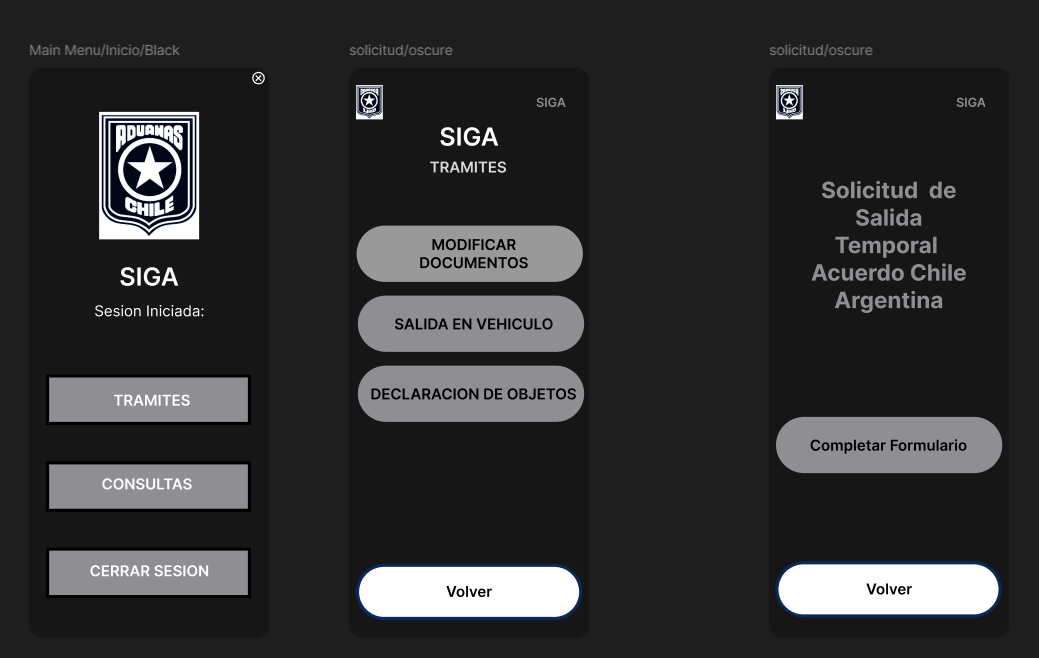
* 1. Principios de diseño

| **PRINCIPIO** | **DESCRIPCIÓN** | **APLICACIÓN EN EL SISTEMA** |
| --- | --- | --- |
| Usabilidad | Facilidad con la que los usuarios pueden aprender a usar, operar y entender el sistema, y qué tan atractivo es para ellos. | Las interfaces están diseñadas para ser intuitivas y fáciles de usar, con formularios claros y navegación sencilla, reduciendo la curva de aprendizaje para funcionarios y ciudadanos. |
| Rendimiento | Capacidad del sistema para responder en un tiempo determinado ante una carga de trabajo específica. | El sistema está optimizado para procesar solicitudes y consultas en menos de 3 segundos, incluso bajo cargas altas, asegurando una experiencia fluida en pasos fronterizos de alto tráfico. |
| Cohesión | Grado de unión entre elementos. | Los elementos de cada módulo están altamente relacionados. Cada módulo cuenta con elementos esenciales para su funcionamiento. |
| Modularidad | Las tareas del sistema están divididas en instancias con tareas propias. | Existen dentro del sistema principal (ingreso de vehículos), cuatro módulos diferenciados según la fase del trámite. |
| Acoplamiento | El acoplamiento es el que tan dependientes son los módulos entre sí. | para un bajo acoplamiento en el sistema hay que minimizar las dependencias directas entre módulos para usar interfaces y el patrón MVC para desacoplar las vistas, controladores y modelos. |
| Reutilización | Uso de componentes existentes en nuevas funcionalidades. | Los componentes de formularios y validación se reutilizan para ingreso y salida de vehículos. |

1. **PROTOTIPO**
   1. Propósito

Se visualiza el flujo de acción de usuarios y se obtiene retroalimentación temprana del servicio principal.

* 1. Mockups



El prototipo sigue dos rutas de flujo de acción según el tipo de usuario que ingresa: Ciudadano; Ejecutivo.

En la primera imagen, las tres interfaces pertenecen a la instancia de un usuario Tipo Cliente:

1. Se despliega al confirmar el inicio de sesión y muestra un menú de dos ítems: Trámites y Consultas. Más abajo, la opción de Cierre de Sesión.

2. Tras clickear la opción de Trámites, el sistema procede a mostrar los dos tipos de documento a completar: Salida en Vehículo y Declaración de Objetos. Además, la opción de modificar datos ya guardados de los mismos.

3. Siguiendo con la realización de trámites, la tercera instancia muestra el inicio para completar la instancia Formulario: Solicitud de Salida Temporal Acuerdo Chile Argentina (formulario principal para la declaración de específicos de un propietario; vehículo; y detalles del viaje).

En la segunda imagen, las tres interfaces pertenecen a la instancia de un usuario Tipo Ejecutivo:

1. Se despliega al iniciar, una pantalla menú similar a la del usuario Cliente, esta vez con los ítems: Informes y Consultas, y la misma opción de cierre de sesión.

2. Seleccionar la instancia Informes desplegará un menú de tres ítems que almacenan los datos enviados por usuarios con calidad de Ciudadano al sistema. Lista de Usuarios (información de personas naturales ingresadas en el sistema con una cuenta SIGA); Lista de Formularios (Datos enviados en la instancia Formulario por Ciudadanos); Lista de declaraciones (Datos enviados en la instancia Declaración de Objetos no Equipaje por Ciudadanos).

Cada una de estas pantallas cuenta con una tabla de datos y una barra de búsqueda.

* 1. Justificar herramientas de prototipado

Programa utilizado: Figma.

Se ha seleccionado Figma como plataforma de trabajo principalmente porque permite el trabajo colaborativo simultáneo. Pero además, es posible crear un prototipo de alta fidelidad, funcional, con una imagen clara de interfaces y funcionalidad mínima.

1. **EVALUACIÓN DE CALIDAD HEURÍSTICA DE NIELSEN**

* 1. Propósito

Se trata de una guía para evaluar la calidad de un sistema informático desde la perspectiva del usuario. De Forma sencilla; mejora la usabilidad.

* 1. Lista de verificación

| Visibilidad del estado del sistema | > La app cuenta con pantallas que confirman el envío correcto de datos |
| --- | --- |
| Coincidencia sistema-mundo real | > Los formularios de la app se asemejan al formato que tienen como documentos de papel  > Los formularios están seccionados según su asunto principal (Vehículo, Propietario, Viaje, Declaración de Objetos) |
| Control y libertad | > Opción de “Limpiar” formularios  > Opción de “Modificar” datos de usuario ya guardados  > Siempre esta la opcion para el usuario de volver a la pantalla anterior |
| Consistencia y estándares | > Proteccion de datos personales  > El sistema guarda automáticamente los datos de instancias tipo Formulario, en calidad de informes que pueden ser consultados |
| Prevención de errores |  |
| Reconocer en lugar de recordar | > Cada botón está etiquetado con la acción correspondiente  > Siempre esta la opcion para el usuario de volver a la pantalla anterior |
| Flexibilidad y eficiencia de uso | > Cada instancia está conectada a una página inicial/principal, a la que es fácil volver  > El sistema distingue entre dos tipos de usuario diferente, con funciones acorde  > Los formularios cuentan con instancias separadas para facilitar su ejecución y la administración de datos |
| Estética y diseño minimalista | > No se utilizan ítems innecesarios  > Se mantiene en colores neutros |
| Ayuda al usuario reconocer y recuperarse de errores | > La información no completada, o que no cumple con el requisito indicado, será señalado con una alerta y detendrá el progreso del trámite hasta ser corregido |
| Ayuda y documentación | > El menú cuenta con una sección de Información para ayudar al usuario con el uso de la app  > El sistema guarda automáticamente los datos de instancias tipo Formulario, en calidad de informes que pueden ser consultados |

* 1. Análisis y métricas de resultados

| **Criterio Evaluado** | **Métrica aplicada** | **Resultado** | **Observación** |
| --- | --- | --- | --- |
| Facilidad de aprendizaje | Tiempo promedio para completar un formulario por 1ª vez | < 3 minutos | Cumple, según pruebas con usuarios simulados |
| Tasa de error | Formulario enviado con errores en primer intento | < 5% | Validaciones evitaron errores críticos |
| Satisfacción del usuario | Encuesta de retroalimentación básica (escala 1 a 5) | 4.6 promedio | Interfaces claras, navegación fluida |
| Recordación de funciones | ¿Los usuarios recuerdan cómo acceder a su trámite? | 90% lo hace sin ayuda | Menús bien estructurados y opciones identificables |
| Consistencia visual | Análisis visual según normas de estilo | Alto nivel de coherencia | Tipografía, botones, y estructura se mantienen |

1. **CONTROL DE VERSIONES**
   1. Propósito

Permitir el seguimiento de los cambios realizados al documento y al sistema, manteniendo un historial claro de modificaciones.

* 1. Control de versión utilizado (justificar el tipo de control de versión utilizado (fecha, semántica o secuencial)

| **Parte** | **Significado** | **Ejemplo** |
| --- | --- | --- |
| Major | Cambios estructurados o Incompatibles | v.2.0.0 |
| Meinor | Nuevas versiones compatibles | v.1.1.0 |
| Patch | Correcciones o ajustes menores | v.1.0.1 |

* 1. Justificar herramientas de versionamiento

Se utiliza GitHub, una herramienta de control de versiones distribuido ampliamente usada en la industria del software. Esto nos permitirá:

* Rastrear cambios y autores.
* Revertir a versiones anteriores.
* Trabajar de forma colaborativa y simultánea.

1. **CONCLUSIONES**

El desarrollo del sistema SIGA se ha estructurado bajo estándares de calidad, con una arquitectura modular basada en microservicios que permite escalabilidad y mantenibilidad.

La aplicación del modelo 4+1 permitió visualizar y documentar desde distintos puntos de vista técnicos y funcionales, facilitando la comprensión del sistema para distintos perfiles.

El uso de Figma permitió validar la usabilidad y funcionalidad del sistema mediante prototipos funcionales de alta fidelidad.

Las heurísticas de Nielsen aplicadas garantizan una interfaz amigable y accesible para usuarios internos y ciudadanos.

Se establece una base sólida para fases futuras del proyecto, incluyendo la implementación, pruebas de aceptación y despliegue en producción.

1. **BIBLIOGRAFÍA**

* "Documentación oficial del Servicio Nacional de Aduanas, marco legal y procedimientos vigentes : Normativa Aduanera Chilena"
* Sistema de control de vehículos motorizados: Login Web
* Nielsen, J. (1994). *Heuristic Evaluation*. Nielsen Norman Group.
* ISO/IEC 25010:2011. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE)*.
* <https://www.aduana.cl/solicitud-de-declaracion-de-admision-o-salida-temporal-de-efectos/aduana/2018-12-16/132320.html>