

**Informe Técnico: Modelamiento Arquitectónico del Sistema Aduanero 2.0**

**Equipo 5** Ignacio Londoño, Matías Gaete  
 **Caso**: Automatización de aduana  
 **Evaluación**: Parcial 2 - Modelamiento de arquitectura del Software

## **Índice**

1. Introducción
2. Justificación del uso de herramientas de gestión y planificación
3. Modelo Arquitectónico Aplicado: Enfoque 4+1  
    3.1. Vista de Escenarios – Diagrama de Casos de Uso  
    3.2. Vista Lógica – Diagrama de Clases  
    3.3. Vista de Procesos – Diagrama de Actividades  
    3.4. Vista de Implementación – Diagrama de Componentes  
    3.5. Vista Física – Diagrama de Despliegue
4. Resumen Simplificado del Sistema (Explicación Infantil)
5. Estándares de Calidad Aplicados
6. Reflexión Final y Conclusiones
7. Glosario de Siglas Relevantes

## **1.Introducción**

Este informe describe el modelamiento arquitectónico en curso del Sistema Aduanero Automatizado 2.0, iniciativa desarrollada en el marco académico para fortalecer la comprensión de conceptos de diseño, modelado y arquitectura de software aplicada al contexto nacional chileno. El enfoque está basado en prácticas modernas de ingeniería de software, con una estructura alineada al modelo 4+1 propuesto por Kruchten, apoyado en herramientas accesibles como Git, Angular, Node.js, y bases de datos relacionales como MySQL. La tecnología se adapta a la realidad local para fortalecer su posible implementación en el Servicio Nacional de Aduanas (SNA).

Cabe destacar que este trabajo representa una versión en construcción, por lo que sigue abierto a iteraciones, revisiones y mejoras conforme se obtenga retroalimentación técnica o contextual.

### **2.📄 Justificación del uso de herramientas de gestión y planificación en el contexto aduanero**

En el desarrollo del Sistema Aduanero Automatizado 2.0, se considera fundamental la incorporación de herramientas que permitan planificar, organizar y supervisar las tareas relacionadas con la implementación del sistema. Si bien este informe se centra principalmente en el modelamiento arquitectónico, resulta necesario reconocer que, dada la complejidad funcional del sistema y la variedad de actores involucrados (viajeros, transportistas, funcionarios del SNA, entidades externas del Estado), el uso de herramientas de apoyo para la gestión de actividades se vuelve altamente recomendable.

Estas herramientas permiten visualizar el avance del proyecto, distribuir tareas por componente o módulo (como autenticación, validación documental, reportes, conectividad externa), establecer flujos de trabajo claros, definir roles y responsabilidades, y realizar seguimiento del cumplimiento de entregables. La estructura modular del sistema facilita este enfoque, ya que cada funcionalidad puede abordarse como una unidad gestionable, permitiendo segmentar el desarrollo por fases o iteraciones.

Además, en un entorno como el del Servicio Nacional de Aduanas, donde se requiere trazabilidad, transparencia y eficiencia, la posibilidad de incorporar sistemas de planificación digital fortalece la capacidad institucional de escalar, monitorear y mantener el sistema a largo plazo. Su uso también permite adaptar el trabajo a metodologías ágiles, facilitando la priorización de requisitos, la reacción ante cambios, y la mejora continua del servicio.

Si bien no se ha utilizado una herramienta específica en esta etapa académica del proyecto, la arquitectura propuesta está diseñada para facilitar una futura integración con plataformas de planificación y seguimiento, sean estas propias del organismo o de uso general, con el objetivo de asegurar una correcta coordinación entre equipos técnicos y funcionales.

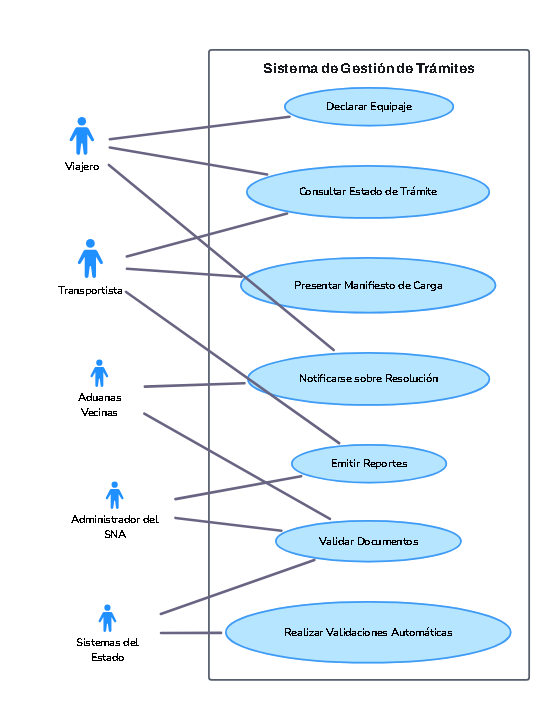
## **3.Vistas Arquitectónicas Aplicadas**

Se aplica el modelo 4+1 con enfoque funcional y representaciones en UML. Las vistas consideradas son:

| **Vista** | **Diagrama Asociado** | **Propósito** |
| --- | --- | --- |
| Vista de Escenarios | Diagrama de Casos de Uso | Validar requerimientos desde la interacción de los actores |
| Vista Lógica | Diagrama de Clases | Representar entidades del dominio y relaciones |
| Vista de Procesos | Diagrama de Actividades | Visualizar flujos funcionales y decisiones del sistema |
| Vista Física | Diagrama de Despliegue | Reflejar infraestructura de hardware/software |
| Vista de Implementación | Diagrama de Componentes | Describir la organización modular de servicios del sistema |

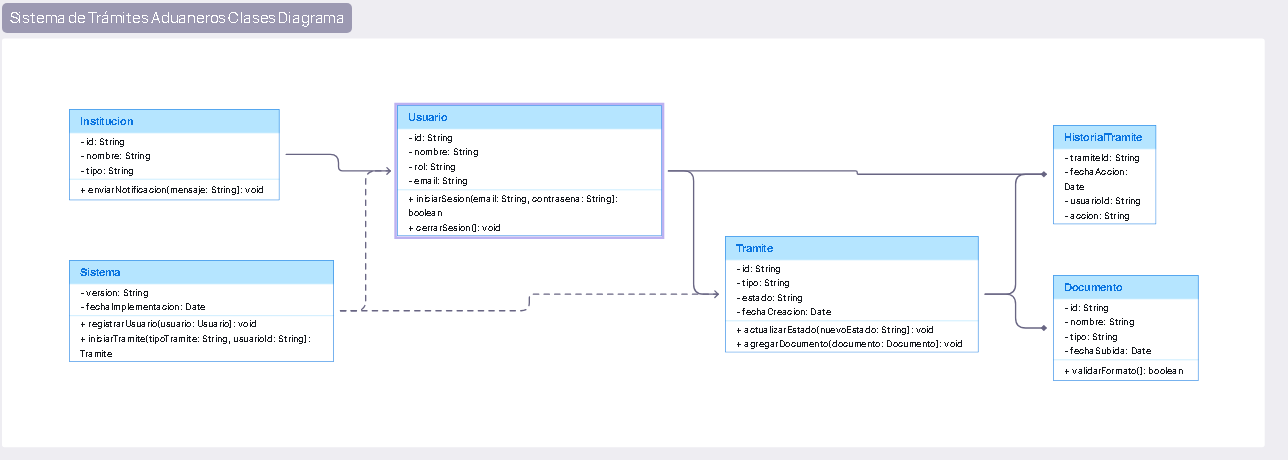
Todas las vistas están construidas con base en el contexto legal, institucional y operativo del sistema aduanero chileno.

## **3.2.Diagrama de Casos de Uso (Vista de Escenarios)**



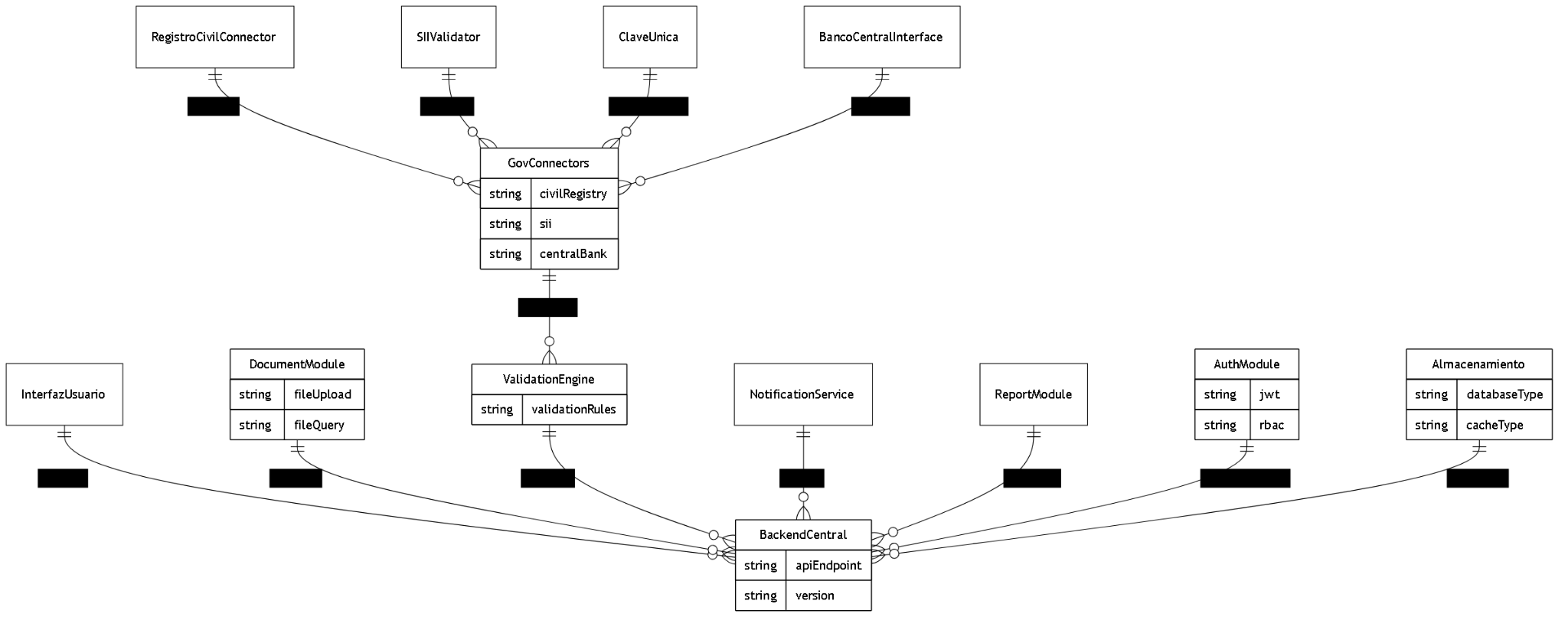
Este diagrama representa las principales interacciones que los distintos actores tienen con el sistema. Está modelado en formato de carriles verticales (swimlane) para distribuir claramente las funcionalidades según el rol. Los actores considerados son: **Viajero**, **Transportista**, **Administrador del SNA**, **Sistemas Externos del Gobierno de Chile** (como ClaveÚnica, Registro Civil y SII), y **Aduanas Limítrofes**. Cada actor está vinculado a un conjunto de casos de uso específicos. El viajero puede autenticarse con ClaveÚnica, declarar equipaje, cargar documentos y consultar el estado de su trámite. El transportista puede presentar manifiestos, validar cargas y seguir la trazabilidad de sus documentos. El administrador accede a funciones avanzadas: aprobar, rechazar trámites, generar reportes y visualizar el flujo global del sistema. Los sistemas externos, como Registro Civil o SII, no interactúan con una interfaz, pero aparecen como actores automatizados que participan en validaciones internas. El propósito del diagrama es **identificar los requerimientos funcionales del sistema desde la perspectiva del usuario**, organizando las funcionalidades claves en un formato claro y verificable. En el contexto chileno, este diagrama evidencia la necesidad de interoperabilidad con plataformas estatales y permite validar los flujos esperados en base a las normativas y protocolos vigentes.

## **3.3.Diagrama de Clases (Vista Lógica)**



Este diagrama modela la estructura interna del sistema desde el punto de vista del dominio de negocio. Describe las clases principales, sus atributos, métodos y las relaciones entre ellas. Las clases más importantes incluyen: Usuario, TrámiteAduanero, Documento, Validación, Notificación, Reporte, y Rol. Por ejemplo, Usuario tiene atributos como RUT, nombre, email, y está asociado a uno o más TrámiteAduanero. Cada TrámiteAduanero puede contener múltiples Documento, cada uno con estado, fecha de envío y tipo. Además, existe una relación entre Documento y Validación, ya que cada documento debe ser validado por el motor automático o por un funcionario. Las Notificación se relacionan con usuarios y eventos del sistema. Este diagrama responde al requerimiento de **estructurar el dominio lógico de la aplicación**, sentando las bases para la persistencia de datos, la lógica de negocio y el desarrollo orientado a objetos. En el contexto chileno, estas clases reflejan entidades reales como el RUT, documentos electrónicos con firma avanzada, y trámites sujetos a revisión institucional.

## **3.4.Diagrama de Componentes (Vista de Implementación)**



El sistema se organiza en componentes desacoplados, cada uno con una función específica y accesibles mediante APIs REST. El usuario accede desde InterfazUsuario (Angular SPA), que se comunica con BackendCentral (Node.js + Express), núcleo de la lógica del sistema. DocumentModule permite cargar y consultar archivos. ValidationEngine realiza validaciones y se conecta con GovConnectors, que integra Registro Civil, SII y Banco Central. NotificationService informa al usuario. ReportModule genera informes internos. Todos los módulos validan acceso con AuthModule, usando JWT y RBAC. Los datos se gestionan en Almacenamiento (MySQL y Redis). El sistema es modular, versionado con Git y desplegable mediante contenedores Docker. Se asegura interoperabilidad con el Estado chileno a través de Registro Civil Connector, SII Validator, ClaveÚnica y Banco Central Interface. Las conexiones entre componentes son representadas como flechas que indican dependencias funcionales claras.

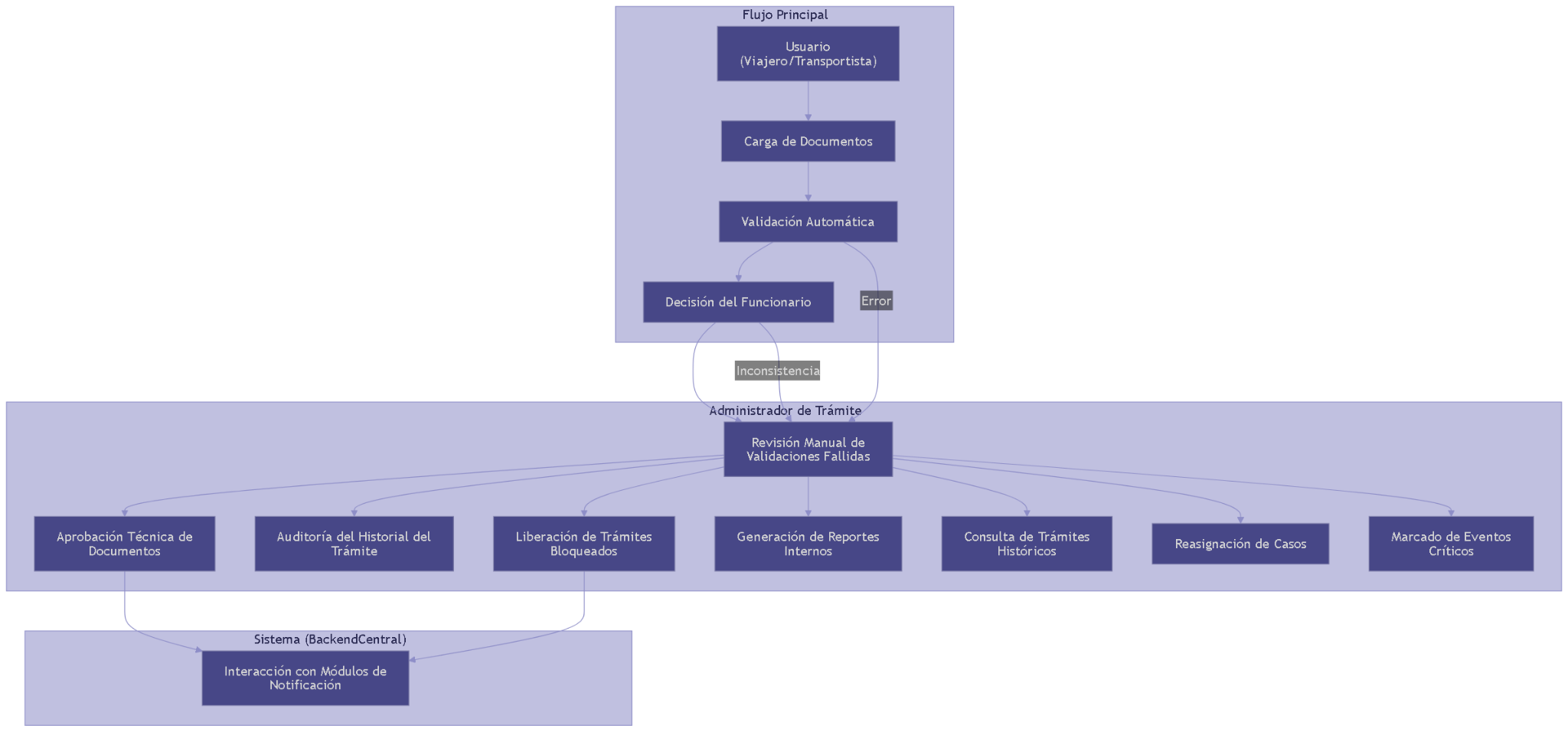
o de manera más simplifica

Este dibujo muestra cómo diferentes partes del sistema aduanero trabajan juntas, como si fueran piezas de un rompecabezas. La parte llamada **InterfazUsuario** es donde las personas usan el sistema, como una pantalla. Esa parte habla con el **BackendCentral**, que es como el cerebro que organiza todo. Ese cerebro se conecta con otras partes: una que guarda documentos (**DocumentModule**), otra que revisa si todo está bien (**ValidationEngine**), otra que manda mensajes (**NotificationService**), otra que hace reportes (**ReportModule**), y una que revisa si tienes permiso para entrar (**AuthModule**). Hay una parte que guarda los datos (**Almacenamiento**) y otra que se conecta con sistemas del gobierno de Chile, como el Registro Civil y el SII (**GovConnectors**). Todas esas piezas se comunican y ayudan a que el sistema funcione sin errores.

## 

## 

## **3.5.Diagrama de Actividades (Vista de Procesos)**



## 

## 

## En el contexto del sistema aduanero automatizado, el Administrador de Trámite es un actor clave que complementa el flujo principal del proceso digital. Aunque el usuario (viajero o transportista) inicia el trámite cargando los documentos a través del sistema, y el funcionario del Servicio Nacional de Aduanas toma decisiones finales de aprobación o rechazo, existen situaciones en las que se requiere la intervención de un actor con privilegios intermedios: el Administrador. Este actor tiene facultades para intervenir en momentos críticos del proceso, especialmente cuando se presentan errores técnicos, validaciones incompletas o se detectan inconsistencias que no pueden ser resueltas automáticamente. Para representar correctamente esta participación, se debe incluir un nuevo carril (swimlane) titulado “Administrador de Trámite” dentro del diagrama de actividades. En dicho carril se ubicarán acciones como: revisión manual de validaciones fallidas, aprobación técnica de documentos en disputa, auditoría del historial del trámite y liberación de trámites en estado bloqueado. Estas acciones no se ejecutan en todos los flujos, sino cuando el sistema o el funcionario lo derivan como excepción. Por ejemplo, si el módulo de validación detecta un error en el RUT pero este proviene de una inconsistencia del Registro Civil y no de una falla del usuario, el sistema puede derivar ese documento al Administrador para revisión técnica. De forma similar, si un usuario presenta múltiples rechazos por formato o estructura de archivo, el Administrador puede intervenir para revisar si el rechazo fue apropiado o si es necesario reconfigurar la validación. Además, este actor puede generar reportes internos, consultar trámites históricos, reasignar casos o marcar eventos como críticos. En términos visuales, este actor debe estar representado por un carril propio, con sus actividades específicas, y conectado al Sistema (BackendCentral) mediante flechas que representen el flujo alternativo o de excepción. También puede tener interacción directa con los módulos de Notificación, en caso de que deba enviar un aviso manual al usuario sobre alguna revisión especial o desbloqueo. Su participación aporta robustez al sistema, ya que permite control humano en los puntos donde la automatización no es suficiente o donde se requiere interpretación técnica. Este rol puede ser desempeñado por personal interno de tecnología, validadores especializados o supervisores operativos del SNA. Al incluirlo en el diagrama de actividades, se fortalece la claridad del proceso, se evidencian las rutas de excepción, y se asegura la trazabilidad de decisiones complejas dentro del sistema. Por lo tanto, el Administrador no solo es un actor complementario, sino una figura esencial para garantizar la continuidad operativa, la calidad de servicio y la transparencia del flujo digital aduanero

resumen simplificado

Imagina que hay una persona que quiere pasar por una aduana digital para declarar lo que lleva. Esa persona puede ser un viajero o alguien que transporta cosas. Lo primero que hace es entrar al sistema con una clave especial del gobierno (se llama ClaveÚnica). Luego, sube los papeles que necesita, como si los pusiera en una caja digital.

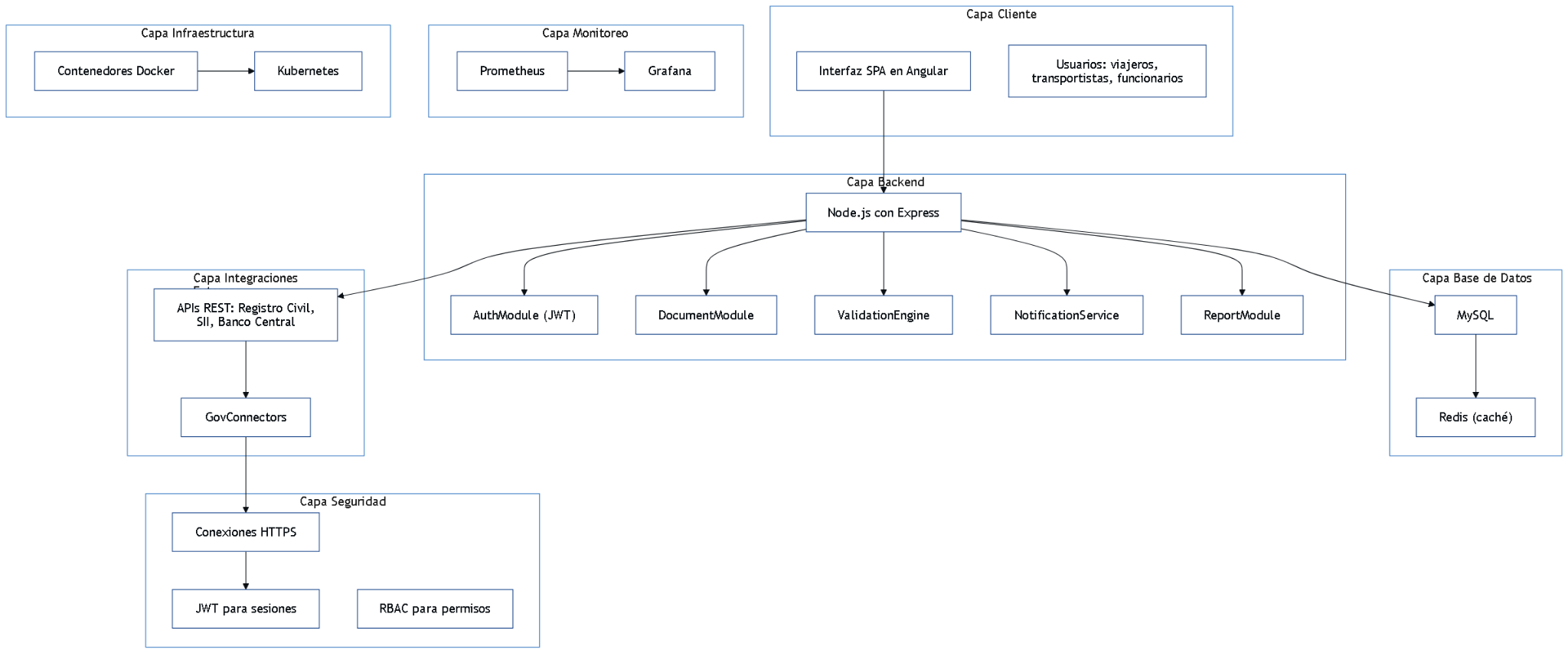
Después, un computador muy inteligente revisa si todo está bien escrito y si el nombre y la firma de esa persona son verdaderos. Si algo está mal, el sistema le dice que tiene que arreglarlo. Si todo está bien, el sistema pregunta a otros servicios del gobierno si la persona está al día con sus papeles y cuentas.

Cuando todos dicen que sí, el sistema avisa a un funcionario que trabaja en la aduana. Esa persona revisa todo por última vez y dice si el trámite se aprueba o no. A veces, si algo no funciona bien, otra persona especial, que se llama “Administrador del Trámite”, ayuda a arreglar las cosas por dentro.

Cuando todo está completo, el sistema guarda la información como si fuera en una caja fuerte, y le avisa al usuario que su trámite ya terminó. Así, todo funciona más rápido, más seguro, y sin errores.

## **Diagrama de Despliegue (Vista Física)**

### **📦 Tabla de Capas del Diagrama de Despliegue – Sistema Aduanero 2.0**



| **🧱 Capa** | **🛠️ Tecnologías** | **📋 Funciones Principales** |
| --- | --- | --- |
| 🧑‍💻 Cliente | Angular (SPA) | Interfaz para viajeros y transportistas. Permite login, carga de documentos y consultas. |
| 🧠 Backend | Node.js + Express | Procesa lógica de negocio, validaciones y coordina la comunicación entre componentes. |
| 💾 Base de Datos | MySQL, Redis | Guarda datos estructurados. Redis acelera procesos con caché y colas para tareas internas. |
| 🌐 Integraciones | APIs REST (RC, SII, BC) | Conexión con servicios externos para validar identidad, impuestos y garantías financieras. |
| 🔐 Seguridad | HTTPS, JWT, RBAC | Protege datos, autentica sesiones y limita acciones según el perfil del usuario. |
| 📊 Monitoreo | Prometheus, Grafana | Recolecta métricas del sistema, muestra dashboards, emite alertas por errores o sobrecarga. |
| 🛠️ Infraestructura | Docker, Kubernetes | Ejecuta contenedores, gestiona escalamiento, fallos y despliegue en nubes públicas o privadas. |

El diseño permite escalabilidad horizontal, despliegue en cloud pública/privada, y resiliencia ante fallos.

El sistema se compone de siete capas funcionales. En la capa Cliente, los usuarios acceden desde el navegador usando una aplicación Angular SPA. Esta interfaz se conecta con el Backend, desarrollado en Node.js con Express, que ejecuta la lógica de negocio, valida información y coordina los servicios. El Backend se comunica con la Base de Datos, que utiliza MySQL para datos estructurados y Redis como caché distribuido. Mediante la capa de Integraciones Externas, el sistema consulta APIs REST de servicios del Estado como Registro Civil, SII y Banco Central. La Seguridad se implementa con HTTPS, autenticación mediante JWT y control de acceso RBAC. El Monitoreo se realiza con Prometheus y Grafana, recolectando métricas y generando alertas. Finalmente, todos los servicios están contenidos en Docker y orquestados con Kubernetes, lo cual permite despliegue automatizado, escalabilidad dinámica y alta disponibilidad en la nube.

## **Estándares de Calidad Aplicados**

El diseño del sistema considera las siguientes normas y buenas prácticas:

* **ISO/IEC 25010**: guía para calidad de software en términos de mantenibilidad, fiabilidad, usabilidad y eficiencia.
* **UML 2.5**: para modelado estructurado y semántico claro.
* **Principios SOLID**: para mantener cohesión y bajo acoplamiento.
* **Control de versiones Git** con convenciones semánticas.
* **Seguridad** basada en firma electrónica, JWT y encriptación.
* **Modularidad** para facilitar pruebas unitarias y mantenimiento progresivo.

## **Reflexión Final y Conclusiones**

La arquitectura planteada se fundamenta en un enfoque progresivo que busca alcanzar altos niveles de calidad, modularidad y seguridad en un sistema institucional crítico como lo es el aduanero. Se ha promovido la adopción de patrones de diseño y estilo arquitectónico apropiados para proyectos escalables, así como una estructura por capas y componentes autónomos para facilitar despliegues incrementales.

Durante el proceso se reflexionó sobre la importancia de representar los distintos puntos de vista de un sistema (escenarios, lógica, procesos, física e implementación), lo cual fortalece la comprensión del dominio, permite realizar ajustes técnicos con mayor rapidez y mejora la comunicación entre los distintos actores involucrados.

Este trabajo ha fomentado la integración de conocimientos técnicos y prácticos en torno al uso de herramientas modernas, técnicas de modelamiento UML y control de versiones, bajo un marco institucional y legal chileno. Así, se establecen bases reales para futuros desarrollos, sin perder de vista que el documento y su contenido están abiertos a mejoras sucesivas que consoliden aún más la solución propuesta.

**Glosario de Siglas Relevantes en el Contexto Chileno**

| **Sigla** | **Significado Completo** |
| --- | --- |
| SNA | Servicio Nacional de Aduanas |
| SII | Servicio de Impuestos Internos |
| RUT | Rol Único Tributario |
| ClaveÚnica | Plataforma de autenticación oficial del Estado de Chile |
| API | Interfaz de Programación de Aplicaciones |
| JWT | JSON Web Token |
| RBAC | Control de Acceso Basado en Roles |

**Nota:** Toda la documentación está sujeta a revisión por parte de los usuarios institucionales, docentes y técnicos del curso. Se espera que continúe evolucionando en futuras entregas y versiones.

“Aunque el equipo no utilizó herramientas como Jira durante el desarrollo, se integraron conceptos clave como flujos de trabajo, control por roles, gestión de tareas y automatización, los cuales fueron modelados mediante diagramas UML y principios de diseño de software moderno.”