

[Sistema Digital OM-2261]

(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.INTRODUCCIÓN** 7](#_Toc199290162)

[1.1. Contexto del Problema 7](#_Toc199290163)

[1.2. Propósito 7](#_Toc199290164)

[1.3. Ámbito 7](#_Toc199290165)

[1.4. Definiciones, acrónimos y abreviaciones 7](#_Toc199290166)

[1.5. Resumen ejecutivo 7](#_Toc199290167)

[1.6. Arquitectura del sistema 7](#_Toc199290168)

[**2.VISIÓN DEL SISTEMA** 8](#_Toc199290169)

[2.1. Descripción general del sistema 8](#_Toc199290170)

[2.2. Objetivos del sistema 8](#_Toc199290171)

[2.3. Principales funcionalidades esperadas 8](#_Toc199290172)

[2.4. Supuestos y dependencias 8](#_Toc199290173)

[**3.ESTILOS Y PATRONES ARQUITECTÓNICOS** 8](#_Toc199290174)

[3.2. Justificación del estilo según el contexto del sistema 8](#_Toc199290175)

[**4.MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS** 8](#_Toc199290176)

[**4.1.** **VISTA DE ESCENARIO** 8](#_Toc199290177)

[4.1.1. Propósito 8](#_Toc199290178)

[4.1.2. Actores 8](#_Toc199290179)

[4.1.3. Diagrama general de casos de uso 8](#_Toc199290180)

[4.1.4. Diagrama de casos de uso específicos 8](#_Toc199290181)

[4.1.5. Lista de casos de uso 8](#_Toc199290182)

[4.1.6. Especificación de casos de uso 8](#_Toc199290183)

[**4.2.** **VISTA LÓGICA** 8](#_Toc199290184)

[4.2.1. Propósito 8](#_Toc199290185)

[4.2.2. Diagrama de clases 8](#_Toc199290186)

[4.2.3. Descripción diagrama de clases 8](#_Toc199290187)

[**4.3.** **VISTA DE IMPLEMENTACIÓN/DESARROLLO** 8](#_Toc199290188)

[4.3.1. Propósito 8](#_Toc199290189)

[4.3.2. Diagrama de componente 8](#_Toc199290190)

[4.3.3. Descripción diagrama de componente 8](#_Toc199290191)

[4.3.4. Diagrama de paquete 8](#_Toc199290192)

[4.3.5. Descripción diagrama de paquete 8](#_Toc199290193)

[**4.4.** **VISTA DE PROCESOS** 9](#_Toc199290194)

[4.4.1. Propósito 9](#_Toc199290195)

[4.4.2. Diagrama de actividad 9](#_Toc199290196)

[4.4.3. Descripción diagrama de actividad 9](#_Toc199290197)

[**4.5.** **VISTA FÍSICA** 9](#_Toc199290198)

[4.5.1. Propósito 9](#_Toc199290199)

[4.5.2. Diagrama de despliegue 9](#_Toc199290200)

[4.5.3. Descripción diagrama de despliegue 9](#_Toc199290201)

[**5.REQUISITOS DE CALIDAD** 9](#_Toc199290202)

[5.1. Propósito 9](#_Toc199290203)

[5.3. Reglas y criterios de evaluación de calidad 9](#_Toc199290204)

[**6.PRINCIPIOS DE DISEÑO APLICADOS** 9](#_Toc199290205)

[6.1. Propósito 9](#_Toc199290206)

[6.4. Diseño centrado en el usuario (UX/UI, prototipos, experiencia de usuario) 9](#_Toc199290207)

[**7.CONCLUSIONES** 9](#_Toc199290208)

[**2.VISIÓN DEL SISTEMA** 10](#_Toc199290209)

[2.1. Descripción general del sistema 10](#_Toc199290210)

[2.2. Objetivos del sistema 10](#_Toc199290211)

[2.3. Principales funcionalidades esperadas 10](#_Toc199290212)

[2.4. Supuestos y dependencias 10](#_Toc199290213)

[**3.ESTILOS Y PATRONES ARQUITECTÓNICOS** 10](#_Toc199290214)

[3.2. Justificación del estilo según el contexto del sistema 10](#_Toc199290215)

[**4.MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS** 10](#_Toc199290216)

[**4.1.** **VISTA DE ESCENARIO** 10](#_Toc199290217)

[4.1.1. Propósito 10](#_Toc199290218)

[4.1.2. Actores 10](#_Toc199290219)

[4.1.3. Diagrama general de casos de uso 10](#_Toc199290220)

[4.1.4. Diagrama de casos de uso específicos 10](#_Toc199290221)

[4.1.5. Lista de casos de uso 10](#_Toc199290222)

[4.1.6. Especificación de casos de uso 10](#_Toc199290223)

[**4.2.** **VISTA LÓGICA** 10](#_Toc199290224)

[4.2.1. Propósito 10](#_Toc199290225)

[4.2.2. Diagrama de clases 10](#_Toc199290226)

[4.2.3. Descripción diagrama de clases 10](#_Toc199290227)

[**4.3.** **VISTA DE IMPLEMENTACIÓN/DESARROLLO** 10](#_Toc199290228)

[4.3.1. Propósito 10](#_Toc199290229)

[4.3.2. Diagrama de componente 10](#_Toc199290230)

[4.3.3. Descripción diagrama de componente 10](#_Toc199290231)

[4.3.4. Diagrama de paquete 10](#_Toc199290232)

[4.3.5. Descripción diagrama de paquete 10](#_Toc199290233)

[**4.4.** **VISTA DE PROCESOS** 10](#_Toc199290234)

[4.4.1. Propósito 11](#_Toc199290235)

[4.4.2. Diagrama de actividad 11](#_Toc199290236)

[4.4.3. Descripción diagrama de actividad 11](#_Toc199290237)

[**4.5.** **VISTA FÍSICA** 11](#_Toc199290238)

[4.5.1. Propósito 11](#_Toc199290239)

[4.5.2. Diagrama de despliegue 11](#_Toc199290240)

[4.5.3. Descripción diagrama de despliegue 11](#_Toc199290241)

[**5.REQUISITOS DE CALIDAD** 11](#_Toc199290242)

[5.1. Propósito 11](#_Toc199290243)

[5.3. Reglas y criterios de evaluación de calidad 11](#_Toc199290244)

[**6.PRINCIPIOS DE DISEÑO APLICADOS** 11](#_Toc199290245)

[6.1. Propósito 11](#_Toc199290246)

[6.4. Diseño centrado en el usuario (UX/UI, prototipos, experiencia de usuario) 11](#_Toc199290247)

[**7.CONCLUSIONES** 11](#_Toc199290248)

# **1.**[**INTRODUCCIÓN**](#_Toc198026102)

[1.1. Contexto del Problema](#_Toc198026103)

En el presente informe se abarcarán los procesos de salida y admisión temporal de vehículos entre Argentina y Chile, los cuales se realizan actualmente de forma semi-manual, mediante formularios impresos (ej. Formulario AFIP OM-2261) que deben ser completados a mano y sellados por funcionarios de aduana en ambos países, lo cual implica:

-Duplicación de datos y tiempos de espera prolongados en frontera.

-Pérdida de información o errores de transcripción manual (chasis/VIN, fechas de vencimiento, datos de pasajeros).

-Dificultades para llevar un registro histórico de movimientos y peticiones de prórroga.

[1.2. Propósito](#_Toc198026104)

El objetivo del informe es proponer una solución técnica en la cual se describan los principales componentes, estilos y patrones de diseño del sistema a desarrollar, teniendo como norte el agilizar los tiempos de realización de los trámites de salida y admisión temporal de vehículos entre Argentina y Chile.

[1.3. Ámbito](#_Toc198026105)

El sistema cubrirá:

1-Registro de datos del propietario y/o autorizado, pasajeros y vehículo.

2-Firma digital o física de la declaración jurada.

3-Gestión electrónica de sellos y firmas de funcionarios en aduanas de salida y entrada.

4-Control de fechas de vencimiento y notificaciones de prórroga.

**5-**Integración con sistemas de aduanas de ambos países (códigos de aduana).

[1.4. Definiciones, acrónimos y abreviaciones](#_Toc198026106)

|  |  |
| --- | --- |
| **ACRONIMO** | **DESCRIPCION** |
| *VIN* | Vehicle Identification Number |
| *DNI* | Documento Nacional de Identidad |
| *OM-2261* | Formulario oficial “Salida y admisión temporal de vehículos” |
| *API* | Application Programming Interface |
| *UI* | User Interface |
| *PDF* | Portable Document Format |
| *QR* | Quick Response code |
|  |  |
|  |  |

[1.5. Resumen ejecutivo](#_Toc198026107)

Se busca desarrollar un sistema web que permita a usuarios y funcionarios aduaneros realizar en línea los tramites de salida y admisión temporal de vehículos. El sistema unificara en una base de datos la captura de datos, firma digital, gestión de sellos, generación de comprobantes y control de vencimientos, reduciendo errores manuales y tiempos de atención frontera.

## [1.6. Arquitectura del sistema](#_Toc198026108)

El proyecto empleara el modelo “4+1” vistas de Kruchten que contara con:

-Vista de escenarios (casos de uso)

-Vista lógica (clases y servicios)

-Vista de implementación (componentes y paquetes)

-Vista de procesos (flujos y concurrencia)

-Vista física (despliegue en infraestructura)

# [**2.VISIÓN DEL SISTEMA**](#_Toc198026109)

## [2.1. Descripción general del sistema](#_Toc198026110)

## Plataforma web responsiva que centraliza la tramitación digital del formulario OM-226/1, gestionando usuarios, vehículos, firmas y sellos de aduanas de Argentina y Chile, y ofreciendo alertas de vencimiento.

## [2.2. Objetivos del sistema](#_Toc198026111)

1- Digitalizar en un único flujo la salida y admisión temporal de vehículos.

2- Reducir tiempos de atención en frontera.

3- Eliminar transcripción manual y errores asociados.

4- Mantener un histórico auditable de todas las operaciones.

5- Generar notificaciones automáticas de prórroga.

## [2.3. Principales funcionalidades esperadas](#_Toc198026112)

## -Registro de solicitud de salida: Captura de datos personales, del vehículo y de los pasajeros, con firma de declaración jurada.

## -Validación de aduana de salida: Acceso de funcionarios para verificar datos, sellar y aprobar.

## -Validación de aduana de entrada: Comprobación cruzada de información al ingresar a Chile.

## -Gestión de vencimientos: Alertas automáticas al acercarse la fecha límite y solicitud de prórroga.

## -Generación de comprobantes en PDF/QR para portación física o digital.

## -Panel de administración para supervisión de operaciones y reportes

## [2.4. Supuestos y dependencias](#_Toc198026113)

# -Disponibilidad de conexión a Internet en puestos habilitados.

# -Existencia de certificados digitales válidos para firma.

# -Integración con API de datos de aduana de Argentina y Chile.

# -Navegadores modernos compatibles con firmas electrónicas

# [**3.ESTILOS Y PATRONES ARQUITECTÓNICOS**](#_Toc198026114)

## [3.1. Estilo](#_Toc198026115) arquitectónico adoptado

## Se opta por una arquitectura de microservicios en capas, compuesta por:

## -API Gateway para un único punto de entrada.

## -Servicios de dominio (usuario, vehículo, aduana, notificaciones, reportes).

## -Capa de datos: bases de datos SQL para gestión transaccional y cola de mensajes para eventos de notificación.

## -Capa de presentación: aplicación web en SPA (Single-Page Application).

## [3.2. Justificación del estilo según el contexto del sistema](#_Toc198026115)

# -Escalabilidad y despliegue independiente: cada servicio puede escalar según demanda (picos en frontera) sin afectar al resto.

# -Alta disponibilidad: fallos en un microservicio (p. ej. notificaciones) no derriben todo el sistema.

# -Desarrollo y mantenimiento facilitados: equipos pueden trabajar en paralelo en servicios acotados.

# -Integración ágil con sistemas externos: API´s desacopladas permiten conectar fácilmente con aduanas de Argentina y Chile.

# -Evolución incremental: nuevas funcionalidades (p. ej. integración con apps móviles) se añaden sin reestructurar la solución global.

## 3.3. Patrones de diseño aplicados

# MVC Modelo Vista Controlador

# [**4.MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS**](#_Toc198026116)

## [**4.1. VISTA DE ESCENARIO**](#_Toc198026117)

### [4.1.1. Propósito](#_Toc198026118)

### El objetivo del proyecto es el de digitalizar y automatizar el trámite aduanero de salida de vehículos desde Argentina hacia Chile, buscando mejorar los tiempos de espera de los usuarios y evitar la congestión en los puestos aduaneros.

### [4.1.2. Actores](#_Toc198026119)

1-Propietario del vehículo

2-Persona Autorizada

3-Pasajeros

4-Funcionario de Aduana (Salida, Argentina)

5-Funcionario de Aduana (Entrada, Chile)

6- Autoridad Certificante

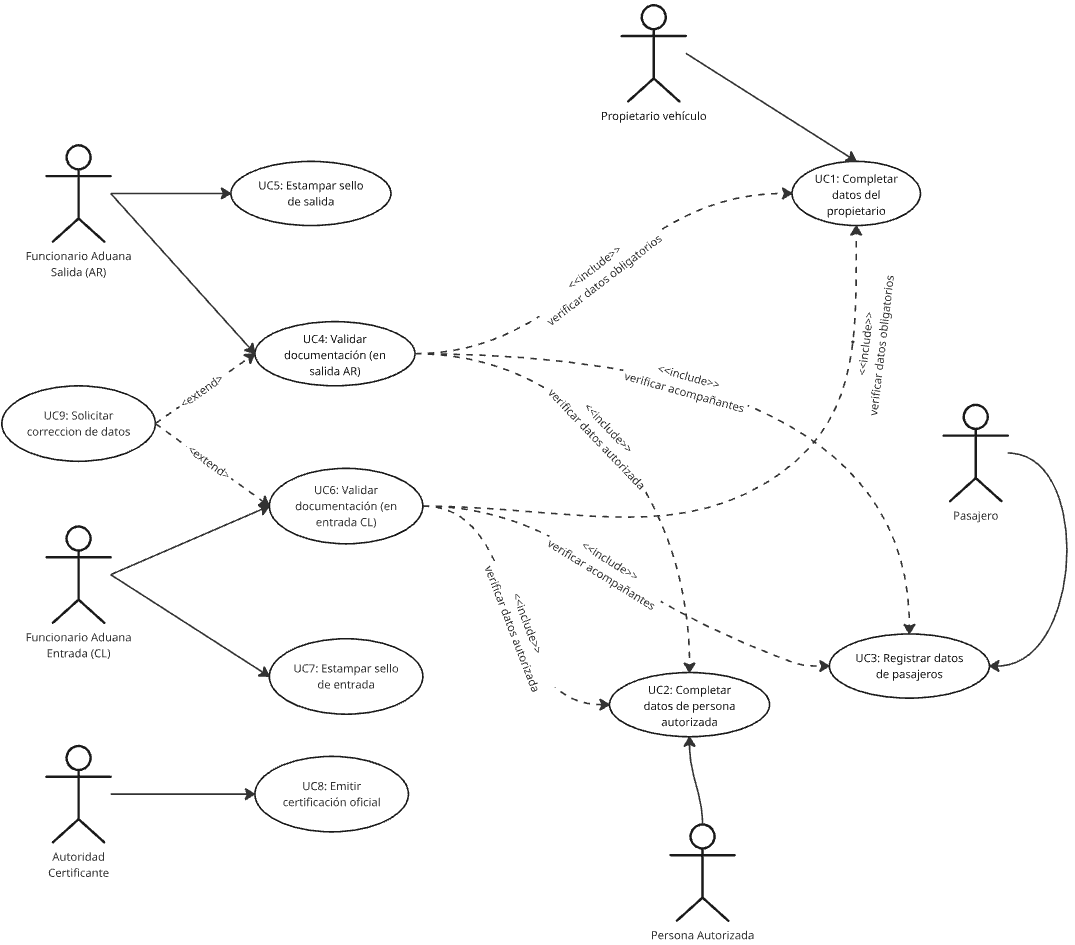
### [4.1.3. Diagrama general de casos de uso](#_Toc198026120)

### 

### 

Linkmiro:<https://miro.com/app/board/uXjVIvGztig=/?moveToWidget=3458764629941849813&cot=14>

### [4.1.4. Diagrama de casos de uso específicos](#_Toc198026121)

Salida de vehículos

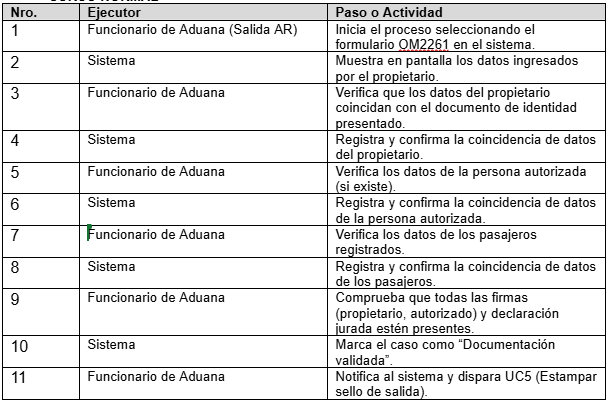
### [4.1.5. Lista de casos de uso](#_Toc198026122)

### 

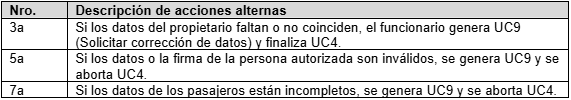
### [4.1.6. Especificación de casos de uso](#_Toc198026123)

## 

**CURSO NORMAL**



**CURSO ALTERNATIVO**

****

## [**4.2. VISTA LÓGICA**](#_Toc198026124)

### [4.2.1. Propósito](#_Toc198026125)

### Definir el modelo de objetos de alto nivel que soporta:

### -La captura y almacenamiento de los datos del formulario OM2261.

### -La representación de los distintos tipos de personas (propietario, autorizado, pasajeros).

### -La validación y el sellado de la documentación por parte de funcionarios de aduana.

### -La emisión de la certificación oficial.

### Con esto, se sientan las bases para implementar la lógica de negocio y los repositorios de datos.

### [4.2.2. Diagrama de clases](#_Toc198026126)

### 

### [4.2.3. Descripción diagrama de clases](#_Toc198026127)

## -FormularioOM2261: Representa la entidad principal del trámite.

## -Guarda estado (borrador, validado, sellado, certificado).

## -Persona y sus subclases: Modelan a los distintos actores humanos del formulario.

## -Cada uno hereda el método firmar ().

## -Vehículo: Contiene los datos técnicos del vehículo.

## -Funcionario Aduana: Actúa sobre formularios, validándolos y estampando sellos.

## -Sello: Objeto que indica el acto de sellado (tipo “Salida” o “Entrada”).

## -Certificación: Emite la constancia legal final del trámite.

## [**4.3. VISTA DE IMPLEMENTACIÓN/DESARROLLO**](#_Toc198026128)

### [4.3.1. Propósito](#_Toc198026129)

### Mostrar cómo se organiza y despliega el software que soporta el trámite OM2261 en términos de componentes (módulos ejecutables, servicios, librerías) y de paquetes (agrupación lógica de clases). Esta vista facilita planificar la construcción, el despliegue y la evolución de la aplicación.

### [4.3.2. Diagrama de componente](#_Toc198026130)

### 

### [4.3.3. Descripción diagrama de componente](#_Toc198026131)

### -UI Formulario OM2261: Aplicación web (Angular/React/Vue…) que permite al Propietario o Persona Autorizada completar el formulario y enviar la solicitud.

### -UI Panel de Gestión: Interfaz interna para funcionarios de Aduana y Autoridad Certificante, donde consultan estados, validan datos y emiten certificados.

### -API Gateway: Punto de entrada unificado para todas las peticiones HTTP, gestiona autenticación, enrutamiento y rate-limiting.

### -Servicio OM2261: Microservicio central que orquesta el flujo: recibe datos, persiste formulario, invoca validaciones y sellados.

### -Servicio Usuarios: Microservicio dedicado a obtener y validar identidad de propietarios, autorizados y pasajeros contra un directorio de usuarios.

### -Servicio Sellado: Componente encargado de generar y almacenar los objetos “Sello” (salida/entrada), registrar la fecha y tipo.

### -Servicio Certificación: Genera la “Certificación Oficial” final, asigna número, fecha de emisión y vínculo al formulario.

### -Base de Datos OM2261: Almacén relacional (por ejemplo PostgreSQL) donde se guardan entidades: Formulario, Estado, Sellos, Certificaciones.

### -Almacén de Documentos: Sistema NoSQL o blob storage (por ejemplo S3, MongoDB GridFS) para guardar PDFs, imágenes de sellos y certificados.

### [4.3.4. Diagrama de paquete](#_Toc198026132)

### 

### [4.3.5. Descripción diagrama de paquete](#_Toc198026133)

## com.aduana.om2261.domain: Contiene las clases de modelo de negocio: Formulario, Persona, Vehiculo, Sello, Certificacion. Representan el estado y reglas de negocio.

## com.aduana.om2261.application: Define las interfaces de los servicios de aplicación (IFormularioService, ISelladoService, ICertificacionService) que orquestan casos de uso.

## com.aduana.om2261.infrastructure: Implementaciones de baja capa: repositorios, clientes HTTP a microservicios externos, almacenamiento de documentos, acceso a base de datos y servicios de usuarios.

## com.aduana.om2261.web: Controladores REST (FormularioController, GestionController) que exponen endpoints para la UI y consumidores de la API.

## [**4.4. VISTA DE PROCESOS**](#_Toc198026134)

### [4.4.1. Propósito](#_Toc198026135)

### Definir de forma detallada el flujo de actividades y las decisiones clave que intervienen en el trámite OM2261, mostrando cómo interactúan los actores con el sistema desde la captura de datos hasta la emisión de la certificación oficial.

### [4.4.2. Diagrama de actividad](#_Toc198026136)

### Link diagrama actividad miro: https://miro.com/app/board/uXjVIvGztig=/?moveToWidget=3458764629949228232&cot=14

### [4.4.3. Descripción diagrama de actividad](#_Toc198026137)

## -Inicio (Propietario / Persona Autorizada)

## -El actor inicia el proceso completando el formulario OM2261.

## -En caso de que el conductor no sea el propietario, completa además los datos de la persona autorizada.

## -Registra los datos de los pasajeros acompañantes (si los hubiera).

## -Persistencia (Sistema) El sistema recibe el formulario completo y lo almacena.

## -A continuación, notifica al funcionario de aduana responsable de la Salida en Argentina.

## -Validación de Salida (funcionario de Aduana – Salida)

## -El funcionario verifica que todos los campos obligatorios del formulario estén completos y sean correctos.

## -Si faltan datos o hay errores, se detiene el flujo y se ejecuta la acción “Solicitar corrección de datos” (flujo alternativo).

## -Si todo está OK, estampa el sello de salida y notifica al funcionario de entrada.

## [**4.5. VISTA FÍSICA**](#_Toc198026138)

### [4.5.1. Propósito](#_Toc198026139)

### Mostrar la topología de hardware y software donde se despliegan los componentes del sistema OM2261, con sus nodos, contenedores, bases de datos y canales de comunicación. Esto facilita planificar la infraestructura, redes y cargas de trabajo.

### [4.5.2. Diagrama de despliegue](#_Toc198026140)

### 

### [4.5.3. Descripción diagrama de despliegue](#_Toc198026141)

# Cliente Web (Navegador) Actor que ejecuta la interfaz de usuario (Angular/React/Vue), envía y recibe peticiones HTTPS al API Gateway.

# NGINX / API Gateway Punto de entrada unificado: gestiona TLS, balanceo de carga y enruta tráfico a los microservicios.

# Cluster de Microservicios Cada servicio corre en un contenedor o pod independiente:

# -Servicio OM2261: orquesta el flujo principal (recepción, validación y sellado).

# -Servicio Usuarios: valida identidad de propietario, autorizado y pasajeros contra directorio.

# -Servicio Sellado: genera y almacena objetos “Sello” (salida/entrada).

# -Servicio Certificación: emite la certificación oficial final.

# PostgreSQL (Base OM2261) Base de datos relacional donde se guardan formularios, estados, sellos y certificados.

# Blob Storage (PDFs, Sellos) Almacén de objetos para documentos generados (PDF del formulario, imágenes de sellos).

# Keycloak (Auth Service) Servicio de autenticación/autorización que emite tokens JWT y gestiona usuarios/roles.

# Flujos de comunicación

# El Cliente Web realiza llamadas HTTPS al API Gateway.

# El Gateway enruta las solicitudes al Servicio OM2261.

# OM2261 habla con:

# -Servicio Usuarios para obtener datos de personas.

# -Servicio Sellado y Servicio Certificación para realizar sellados y emitir constancias.

# -La Base de Datos vía JDBC para persistir entidades.

# -El Blob Storage para guardar PDFs y sellos.

# -El Auth Service para validar y renovar tokens OAuth2/JWT.

# [**5.REQUISITOS DE CALIDAD**](#_Toc198026142)

## [5.1. Propósito](#_Toc198026143)

### Definir los atributos de calidad que el sistema OM2261 debe cumplir para garantizar una experiencia de usuario óptima, un mantenimiento ágil, un rendimiento adecuado, y un alto nivel de seguridad y accesibilidad, así como su facilidad de despliegue en distintos entornos.

### [5.2. Atributos de calidad](#_Toc198026144)

|  |  |  |
| --- | --- | --- |
| **ATRIBUTO DE CALIDAD** | **DESCRIPCIÓN** | **JUSTIFICACIÓN** |
| Usabilidad | Facilidad con que el propietario, Persona Autorizada y Funcionarios interactúan con la interfaz (web o móvil) para completar y validar el formulario. | Una UI clara y coherente minimiza errores al ingresar datos críticos (formulario OM2261), acelera el proceso aduanero y reduce la carga de soporte. |
| Accesibilidad | Cumplimiento de las pautas WCAG 2.1 (nivel AA) para usuarios con discapacidad visual, auditiva o motriz, incluyendo navegación por teclado y etiquetas semánticas. | Garantiza que el sistema sea utilizable por todos los actores, cumpliendo normativas internacionales y reduciendo riesgos de exclusión o sanciones legales por inaccesibilidad. |
| Rendimiento | Rapidez en la carga de páginas, en el tiempo de respuesta de las APIs de validación y en las operaciones de sellado/certificación. | Un rendimiento adecuado evita cuellos de botella en los pasos críticos (verificación y sellado) y mejora la experiencia, especialmente en pasos manuales de aduana con alta concurrencia de vehículos. |
| Mantenibilidad | Facilidad para comprender, corregir, extender o reutilizar el código; baja complejidad ciclomática y alta cobertura de pruebas automatizadas. | Reduce el costo y tiempo de evolución del sistema (nuevos cambios normativos, corrección de errores), y favorece la incorporación de nuevos desarrolladores sin curva de aprendizaje excesiva. |
| Seguridad | Protecciones frente a ataques (inyección, XSS, CSRF, configuración insegura), cifrado de datos sensibles en tránsito y reposo, autenticación y autorización robustas. | El formulario OM2261 maneja datos personales y aduaneros críticos; una brecha podría comprometer la integridad del proceso aduanero y la privacidad de los usuarios, con consecuencias legales y reputacionales. |
| Portabilidad | Capacidad de desplegar el sistema en diferentes entornos (Windows, Linux, contenedores Docker/Kubernetes) sin cambios significativos. | Facilita el aprovisionamiento en distintos entornos de aduanas (oficinas fronterizas, nube, contenedores) y garantiza continuidad operativa al migrar infraestructura o adoptar nuevas plataformas. |

### [5.3. Reglas y criterios de evaluación de calidad](#_Toc198026144)

|  |  |
| --- | --- |
| **ATRIBUTO** | **CRITERIOS DE MEDICION** |
| Usabilidad | • Puntuación SUS > 80 |
| • Tasa de éxito de tarea > 90% |  |
| • Tiempo medio de completar formulario  < 2 min |  |
| Accesibilidad | • Cumplimiento WCAG 2.1 nivel AA |
| • O errores críticos en auditoría Axe o WAVE |  |
| • Navegación completa por teclado |  |
| Rendimiento | • Tiempo de carga primera vista < 2 s (95 % de las peticiones) |
| •TTFB < 500 ms |  |
| • API de sellado/respuesta < 300 ms |  |
| Mantenibilidad | •Cobertura de pruebas unitarias/integración > 80% |
| • Complejidad ciclomática media de métodos < 10 |  |
| • Ratio comentarios/código < 5% |  |
| Seguridad | • OWASP ASVS nivel 2 cumplido |
| • O vulnerabilidades críticas en escaneo SAST/DAST |  |
| • Autenticación MFA y cifrado TLS 1.2+ |  |
| Portabilidad | • Despliegue exitoso en Windows Server y Linux (Docker/K8s) |
| • Tiempo de provisionamiento < 15 min |  |
| • Scripts de CI/CD idempotentes |  |
|  |  |

# Herramientas o métodos que se utilizarán:

# Usabilidad:

# – Pruebas de usabilidad con usuarios finales (entrevistas y tareas guiadas)

# – Evaluación heurística según Nielsen

# Accesibilidad:

# – Auditorías automáticas con Axe y WAVE

# – Pruebas manuales de navegación por teclado y lectura de pantalla

# Rendimiento:

# – Pruebas de carga y estrés con JMeter o Gatling

# – Auditorías de rendimiento de front-end con Lighthouse

# Mantenibilidad:

# – Análisis estático de código con SonarQube

# – Revisión de código por pares y métricas de cobertura con JaCoCo/Cobertura

# Seguridad:

# – Escaneo SAST (e.g. SonarQube, Checkmarx) y DAST (e.g. OWASP ZAP, Burp Suite)

# – Pentesting periódico y revisión de dependencias vulnerables (SCA)

# Portabilidad:

# – Pruebas de despliegue automático en entornos locales y en la nube mediante Docker Compose y Kubernetes

# – Validación de contenedores con herramientas de CI/CD (GitLab CI, Jenkins)

# [**6.PRINCIPIOS DE DISEÑO APLICADOS**](#_Toc198026145)

## [6.1. Propósito](#_Toc198026146)

## Garantizar que la experiencia de uso satisfaga las necesidades de todos los actores (propietario, autorizado, pasajeros, funcionarios), reduciendo errores y acelerando el proceso aduanero.

## [6.12. P](#_Toc198026146)rincipios 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 |
|  |  |  |
|  |  |  |
|  |  |  |

## [6.3. Diseño centrado en el usuario (UX/UI, prototipos, experiencia de usuario)](#_Toc198026147)

# Investigación de usuarios: Entrevistas y observación en frontera para entender flujos y puntos de dolor.

# Definición de personas y escenarios: Creación de perfiles (p. ej. “Funcionario de Salida”, “Propietario frecuente”) y sus recorridos principales.

# Wireframes y prototipos interactivos: Bocetos de baja fidelidad para validar estructura; prototipos de alta fidelidad para flujos críticos (Figma).

# Pruebas de usabilidad: Test moderados con funcionarios reales midiendo tasa de éxito (> 90 %) y tiempo de tarea (< 2 min).Sistema de diseño accesible: Componentes reutilizables con pautas WCAG 2.1 nivel AA, navegación por teclado y lectores de pantalla.

# Iteración continua: Incorporación de feedback en sprints cortos, ajustando labels, validaciones y flujos según hallazgos.

# [**7.CONCLUSIONES**](#_Toc198026148)

La aplicación de estos principios de diseño y prácticas de UX garantiza que el sistema OM2261 sea robusto, flexible y realmente orientado a las necesidades de todos sus actores.

Al mantener cohesión, bajo acoplamiento y modularidad, el software resulta fácil de mantener y extender.

El encapsulamiento y la abstracción protegen la integridad de los datos y permiten evolucionar las implementaciones sin afectar la capa de presentación ni otros servicios.

Por último, el enfoque de diseño centrado en el usuario, con prototipos validados y pruebas de usabilidad, asegura una experiencia intuitiva y accesible, reduciendo errores y tiempos de tarea en cada cruce fronterizo.

En conjunto, estos principios forman la base para un sistema aduanero eficiente, seguro y sostenible a largo plazo.

# [**8.**](#_Toc198026148)**BIBLIOGRAFIA**

https//serviciosweb.afip.gob.ar/genericos/formularios/pdf/OM2261.pdf

<https://platzi.com/tutoriales/1248-pro-arquitectura/4142-modelo-de-arquitectura-41/>

https://www.argentina.gob.ar/aplicaciones/fronteras/recomendaciones/chile