**[Nombre del proyecto]**

**(DAS) Documento Arquitectura de Software**

**Versión 1.0**

**Identificación de Documento**

| **Identificación** | Documento de Arquitectura del Sistema (DAS) – DATET |
| --- | --- |
| **Proyecto** | Sistema de Declaración Aduanera de Tránsito y Efectos de Turistas (DATET) |
| **Versión** | 1.0 |

| **Documento mantenido por** | Nicolás |
| --- | --- |
| **Fecha de última revisión** | 08/07/2025 |
| **Fecha de próxima revisión** | 09/07/2025 |

| **Documento aprobado por** | Mabele Herrera Pino |
| --- | --- |
| **Fecha de última aprobación** | 10/07/2025 |

**Historia de Revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 03/07/2025 | 0.1 | Borrador inicial del DAS, estructura base | Nicolás |
| 05/07/2025 | 0.3 | Agregado de Vista Lógica y Diagrama de Clases | Nicolás |
| 08/07/2025 | 0.5 | Inclusión de Prototipo y Evaluación Heurística Nielsen | Nicolás |
| 09/07/2025 | 0.9 | Integración de Planilla de Requisitos y Carta Gantt | Nicolás |

**Tabla de Contenidos**

[**1.**](#_heading=h.93ng64q0hgv1) **INTRODUCCIÓN 4**

[1.1.](#_heading=h.qs4r7e1cpen) Contexto del Problema (General) 4

[1.2.](#_heading=h.2l07yp8gc29k) Propósito 4

[1.3.](#_heading=h.srajqcq1g0my) Ámbito 4

[1.4.](#_heading=h.18ha54s25zkb) Definiciones, acrónimos y abreviaciones 4

[1.5.](#_heading=h.69yxrdr492ap) Resumen ejecutivo (General) 4

[1.6.](#_heading=h.e56ufoo9r5li) Arquitectura del sistema (General) 4

[**2.**](#_heading=h.nbyimy4g59vw) **VISIÓN DEL SISTEMA (General) 4**

[2.1.](#_heading=h.cr02l96fj009) Descripción general del sistema 4

[2.2.](#_heading=h.q47xe1knesy1) Objetivos del sistema 4

[2.3.](#_heading=h.j24lppa8pcre) Requisitos funcionales y no funcionales 4

[2.4.](#_heading=h.rlxn2g1u9qt9) Supuestos y dependencias 4

[**3.**](#_heading=h.n9i822hlng7a) **ESTILOS Y PATRONES ARQUITECTÓNICOS (General) 4**

[3.2.](#_heading=h.567ielagits) Justificación del estilo según el contexto del sistema 4

[**4.**](#_heading=h.98xjx83w7xg1) **MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS 4**

[4.1.](#_heading=h.aanuksqi7oxn) VISTA DE ESCENARIO (General y salida vehículo o entrada vehículo) 4

[*4.1.1.*](#_heading=h.7pqlnu4e68w9) *Propósito (General) 4*

[*4.1.2.*](#_heading=h.mm2ltzlqonfc) *Actores (General) 4*

[*4.1.3.*](#_heading=h.5fslg7r78dpq) *Diagrama general de casos de uso (General) 4*

[*4.1.4.*](#_heading=h.69v7loujp59g) *Diagrama de casos de uso específicos (salida vehículo o entrada vehículo) 4*

[*4.1.6.*](#_heading=h.yznsvcyx9bhd) *Especificación de casos de uso (UN caso de uso principal de la salida vehículo/entrada vehículo) 5*

[4.2.](#_heading=h.qmwu4jsl6tt9) VISTA LÓGICA (salida vehículo o entrada vehículo) 6

[*4.2.1.*](#_heading=h.5h2vr616615n) *Propósito 6*

[*4.2.2.*](#_heading=h.5umpgwkg4orm) *Diagrama de clases 6*

[*4.2.3.*](#_heading=h.7ohwwzdia9zi) *Descripción diagrama de clases 6*

[4.3.](#_heading=h.fv3g64k72619) VISTA DE IMPLEMENTACIÓN/DESARROLLO (salida vehículo o entrada vehículo) 7

[*4.3.1.*](#_heading=h.zifytst6wno7) *Propósito 7*

[*4.3.2.*](#_heading=h.pgjt6r7ns2bz) *Diagrama de componente 7*

[*4.3.3.*](#_heading=h.lza2t8h7pnz8) *Descripción diagrama de componente 7*

[*4.3.4.*](#_heading=h.ewvqvlt8gbud) *Diagrama de paquete 7*

[*4.3.5.*](#_heading=h.90yj1o2kodpw) *Descripción diagrama de paquete 7*

[4.4.](#_heading=h.vfbdkit9a6d8) VISTA DE PROCESOS (salida vehículo o entrada vehículo) 7

[4.4.1.](#_heading=h.r5kcryc1si6g) Propósito 7

[4.4.2.](#_heading=h.hfbgm647266n) Diagrama de actividad 7

[4.4.3.](#_heading=h.utd6w9n9q8lo) Descripción diagrama de actividad 7

[4.5.](#_heading=h.125w77uxlazg) VISTA FÍSICA (salida vehículo o entrada vehículo) 7

[*4.5.1.*](#_heading=h.typfwrg1ue9b) *Propósito 7*

[*4.5.2.*](#_heading=h.7nwg9gwcqbjb) *Diagrama de despliegue 7*

[*4.5.3.*](#_heading=h.gbcmietc455s) *Descripción diagrama de despliegue 7*

[5.](#_heading=h.nk5svjxr81ve) REQUISITOS DE CALIDAD (General) 7

[5.1.](#_heading=h.2wv3j4qw0d5z) Propósito 7

[*5.3.*](#_heading=h.34bz94f1ask2) *Reglas y criterios de evaluación de calidad 7*

[**6.**](#_heading=h.qzx220p5dxq9) **PRINCIPIOS DE DISEÑO APLICADOS** 8

[*6.1.*](#_heading=h.wexco5xcgp85) *Propósito 8*

[6.2.](#_heading=h.dnq9b04k6q0h) Principios de diseño (por ejemplo: abstracción, acoplamiento, cohesión, encapsulamiento, modularidad) 8

[**7.**](#_heading=h.xwssjjqi6onr) **PROTOTIPO 8**

[7.1.](#_heading=h.wca8lccni348) Propósito 8

[7.2.](#_heading=h.cyhh2tp0v48d) Mockups (imágenes con una breve descripción) 8

[7.3.](#_heading=h.tbwpp8tiex1) Justificar herramientas de prototipado 8

[**8.**](#_heading=h.j5eae93up4ut) **EVALUACIÓN DE CALIDAD HEURÍSTICA DE NIELSEN** 8

[8.1.](#_heading=h.jotmjp5e88lx) Propósito 8

[8.2.](#_heading=h.aeowyye2kzr6) Lista de verificación 8

[8.3.](#_heading=h.fclxk4pig3k4) Análisis y métricas de resultados 8

[**9.**](#_heading=h.6kqeoupagkdu) **CONTROL DE VERSIONES** 8

[9.1.](#_heading=h.wm4ibsv6m1cw) Propósito 8

[9.2.](#_heading=h.ifyih33wnc0z) Control de versión utilizado (justificar el tipo de control de versión utilizad (fecha, semántica o secuencial) 8

[9.3.](#_heading=h.ffuqw52tcif1) Justificar herramientas de versionamiento 8

[**7.**](#_heading=h.7jxyeh61lw04) **CONCLUSIONES 8**

[**8.**](#_heading=h.beb0nyg92s5b) **BIBLIOGRAFÍA 8**

[**9.**](#_heading=h.etglhsvfb4pb) **ANEXOS 8**

[9.1.](#_heading=h.g7ewhyw673js) Planilla de requerimientos 8

[9.2.](#_heading=h.i134rgy3dfvf) Carta Gantt 8

1. **INTRODUCCIÓN**
   1. Contexto del Problema (General)

El proceso de ingreso vehicular y declaración de efectos por parte de turistas en puntos fronterizos chilenos presenta ineficiencias en la captura, validación y seguimiento de datos. El sistema DATET se propone modernizar dicho proceso, facilitando el registro digital, validación automatizada y control logístico, con énfasis en la experiencia del usuario y la interoperabilidad institucional.

* 1. Propósito

Establecer los fundamentos arquitectónicos del sistema DATET, documentando su estructura lógica, física, de procesos y de componentes, así como los principios de diseño, requisitos de calidad y las herramientas utilizadas en su desarrollo.

* 1. Ámbito

Cubre el módulo de ingreso de turistas con vehículos, la declaración de efectos personales, la validación de datos por funcionarios aduaneros y el proceso de confirmación y trazabilidad desde interfaces web

* 1. Definiciones, acrónimos y abreviaciones

| **ACRONIMO** | **DESCRIPCION** |
| --- | --- |
| *DATET* | Declaración Aduanera de Tránsito y Efectos de Turistas |
| *MVC* | Modelo-Vista-Controlador |
| *Nielsen* | Modelo de evaluación de usabilidad heurística |
| *WCAG* | Web Content Accessibility Guidelines |

* 1. Resumen ejecutivo (General)

El sistema DATET fue concebido mediante un enfoque arquitectónico en capas, con aplicación de patrones como MVC y repositorio, control de versiones mediante GitHub, y prototipado centrado en el usuario. Se ha validado funcional y heurísticamente, demostrando alta calidad en rendimiento, seguridad y usabilidad

* 1. Arquitectura del sistema (General)

Incluye vistas de escenario, lógica, desarrollo, procesos y física, dentro del modelo 4+1. La arquitectura permite trazabilidad por actor, modularidad en componentes, y despliegue adaptable en entornos institucionales

1. **VISIÓN DEL SISTEMA (General)**
   1. Descripción general del sistema

DATET permite el ingreso automatizado de turistas y sus efectos personales en pasos fronterizos, validando datos en tiempo real y generando confirmaciones seguras. La interfaz está orientada al usuario final y al funcionario de validación

* 1. Objetivos del sistema

Reducir tiempos de validación y registro

Mejorar trazabilidad documental

Aumentar la calidad de datos capturados

Elevar la usabilidad del sistema frente a perfiles diversos

* 1. Requisitos funcionales y no funcionales

Extraídos de planilla compartida, agrupados por actor: turistas, aduaneros, sistema, transportistas.  
 No funcionales incluyen disponibilidad (99.95%), tiempo de respuesta (<2 seg.), seguridad (protección ante ataques y respaldo de datos), accesibilidad y mantenibilidad

* 1. Supuestos y dependencias

Conectividad constante en aduanas

Acceso a base de datos institucional

Uso de dispositivos móviles o estaciones fijas

Cooperación interinstitucional en interoperabilidad

1. **ESTILOS Y PATRONES ARQUITECTÓNICOS** (General)
   1. Estilo arquitectónico adoptado

**Arquitectura en capas**: separación entre presentación (interfaces), lógica de negocio (validaciones y procesos), y acceso a datos (SQL y persistencia)

* 1. Justificación del estilo según el contexto del sistema

Basado en necesidad de trazabilidad, mantenimiento a largo plazo, alineación con modelo en cascada usado en planificación (ver Carta Gantt). Modularidad facilita actualizaciones sin afectación global

* 1. Patrones de diseño aplicados

**MVC**: Separación entre control, vista y modelo para interfaces

**Repositorio**: Abstracción del acceso a datos

**DTO (Data Transfer Object)**: En transporte de datos entre capas

**Factory (parcial)**: En creación de objetos para tipos de documento

1. **MODELO 4 +1 Y VISTAS ARQUITECTÓNICAS** 
   1. **VISTA DE ESCENARIO** (General y salida vehículo **o** entrada vehículo)
      1. Propósito (General)

Esta vista describe cómo el sistema DATET interactúa con sus usuarios clave mediante casos de uso. El objetivo es modelar escenarios reales que reflejen el flujo de operaciones en el ingreso de turistas, la validación de datos vehiculares y la gestión de efectos personales. Permite identificar responsabilidades del sistema y necesidades funcionales por actor

* + 1. Actores (General)

**Turista**: Persona extranjera o nacional que ingresa al país y declara sus efectos. **Conductor**: Actor que ingresa vehículos al país y gestiona su validación.

**Funcionario de Aduana**: Valida la información ingresada, autoriza el paso o solicita correcciones.

**Sistema DATET**: Encargado de recibir, procesar, validar y confirmar los datos ingresados.

**Sistema Externo**: Bases de datos institucionales conectadas para verificar documentos y permisos

* + 1. Diagrama general de casos de uso (General)

**CU-001**: Registro de turista y efectos personales. **CU-002**: Validación de documentos y datos de vehículo.

**CU-003**: Confirmación del ingreso y emisión de comprobante.

**CU-004**: Corrección de errores en datos ingresados.

**CU-005**: Exportación de registros para informes institucionales

* + 1. Diagrama de casos de uso específicos (salida vehículo **o** entrada vehículo)

Selección de aduana. Ingreso de datos del vehículo.

Validación automática (patente, permiso, identidad del conductor).

Declaración de efectos personales.

Corrección en caso de error.

Generación de confirmación

* + 1. Lista de casos de uso(salida vehículo o entrada vehículo)

| **Código** | **Nombre** | **Actores** |
| --- | --- | --- |
| CU-001-001 | Exportar saldos y puntos a vencer | Funcionario, Sistema DATET |
| CU-002-001 | Exportar actividades | Funcionario, Sistema DATET |
| CU-002-002 | Importar deuda vencida por PDV | Sistema DATET, Sistema Externo |
| CU-004-001 | Generación Archivo PDA Importación | Sistema DATET |
| CU-005-001 | Registrar ingreso vehículo | Conductor, Funcionario, Sistema DATET |
| CU-005-002 | Validar efectos personales | Turista, Funcionario, Sistema DATET |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

* + 1. Especificación de casos de uso (**UN** caso de uso principal de la salida vehículo/entrada vehículo)

| **Caso de Uso** | Registro de turista con vehiculo | **Identificador:** CU-005-001 |
| --- | --- | --- |
| **Actores** | Turista, Conductor, Funcionario de Aduana, Sistema DATET | |
| **Tipo** | Primario | |
| **Referencias** | Requisitos RF-001 a RF-010, RNF-002, RNF-005 | |
| **Precondición** | El turista debe contar con documento vigente; el sistema debe estar disponible y conectado | |
| **Postcondición** | Los datos ingresados son validados, almacenados y se genera la confirmación de ingreso | |
| **Descripción** | El usuario ingresa a través de una interfaz web donde selecciona aduana y país de procedencia. Luego registra datos personales, información del vehículo y efectos personales. El sistema valida automáticamente la información y muestra una confirmación, o en caso de error, solicita correcciones | |
| **Resumen** | [Resumen de alto nivel del funcionamiento | |

**CURSO NORMAL**

| **Nro.** | **Ejecutor** | **Paso o Actividad** |
| --- | --- | --- |
| 1 | Turista ingresa a interfaz | [Descripción del paso actividad ejecutado] |
| 2 | Sistema solicita datos personales |  |
| 3 | Usuario ingresa efectos personales |  |
| [Se describe el proceso o secuencia de pasos ejecutadas usando frases cortas]  [Cada paso del proceso puede ser ejecutado por los Actores o por el sistema]  [Se describe la secuencia de acciones realizadas por los actores y la secuencia de actividades realizada por el sistema como respuesta]. | | |

**CURSO ALTERNATIVO**

| **Nro.** | **Descripción de acciones alternas** |
| --- | --- |
| 1 | Si la validación falla (paso 4), el sistema muestra mensaje de error |
| 2 | Usuario corrige datos |
| 3 | Flujo regresa al paso 2 para reingreso |
| [Cada paso descrito en el curso normal, puede tener actividades alternas, según la distribución de escenarios que ocurra en el flujo de procesos, en esta ficha se completa para cada actividad (haciendo referencia a su número) las posibles secuencias alternas] | |

* 1. **VISTA LÓGICA** (salida vehículo **o** entrada vehículo)
     1. Propósito

La vista lógica tiene como objetivo describir la estructura conceptual del sistema DATET, representando las clases principales, sus relaciones y comportamientos. Esta vista permite entender cómo se organiza la información y cómo se modelan los objetos involucrados en el proceso de ingreso vehicular y declaración de efectos, facilitando la trazabilidad, escalabilidad y mantenibilidad del sistema

* + 1. Diagrama de clases

El diagrama está compuesto por las siguientes clases principales:

* **Turista**
  + Atributos: nombre, documento, nacionalidad, teléfono, correo, domicilio temporal
  + Métodos: registrarDatos(), editarPerfil()
* **Documento** (clase abstracta)
  + Subclases: Cédula, Pasaporte
  + Métodos: validarDocumento(), obtenerTipo()
* **EfectoPersonal**
  + Atributos: descripción, cantidad, tipo (permitido / prohibido)
  + Métodos: agregar(), verificarTipo()
* **Vehículo**
  + Atributos: patente, tipo, modelo, permiso
  + Métodos: validarPermiso(), registrarVehículo()
* **FuncionarioAduana**
  + Métodos: validarIngreso(), visualizarHistorial()
* **IngresoAduanero**
  + Relaciones: asocia a Turista, Vehículo y EfectoPersonal
  + Métodos: crearIngreso(), generarConfirmación()
* **SistemaDATET**
* Métodos: procesarValidación(), guardarRegistro(), mostrarConfirmación()

* + 1. Descripción diagrama de clases

El diagrama modela de manera modular y cohesiva los elementos clave del sistema:

* La clase **Turista** interactúa con el sistema para declarar sus datos personales y efectos.
* La clase **Documento**, mediante herencia, permite manejar distintos tipos de identificación y aplicar validaciones específicas.
* La clase **EfectoPersonal** facilita el registro de bienes transportados, y está vinculada al módulo de validación para determinar si son permitidos.
* La clase **Vehículo** agrupa los atributos técnicos del medio de transporte e incorpora lógica de verificación de permisos.
* **IngresoAduanero** centraliza la relación entre Turista, Vehículo y Efectos, sirviendo como entidad principal del proceso.
* **FuncionarioAduana** accede a los datos registrados para validar el ingreso desde una interfaz segura.
* Finalmente, **SistemaDATET** actúa como orquestador de procesos, conectando las clases anteriores a través de métodos de validación, persistencia y retroalimentación visual

Los principios de diseño aplicados incluyen:

* **Encapsulamiento:** los atributos están protegidos y manipulados mediante métodos públicos.
* **Cohesión alta:** cada clase tiene una responsabilidad bien definida.
* **Acoplamiento bajo:** las clases se comunican entre sí a través de interfaces claras y bien delimitada

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

Esta vista describe cómo los elementos lógicos del sistema DATET se implementan físicamente a través de módulos, componentes y paquetes. Su objetivo es mostrar cómo se organiza el software durante el desarrollo para facilitar la colaboración, el mantenimiento y el control de versiones.

* + 1. Diagrama de componente

El sistema está dividido en componentes desacoplados según responsabilidades funcionales:

* **Componente Interfaz Web**
  + Formulario de ingreso de datos del turista y vehículo.
  + Validación en cliente.
* **Componente Validación Backend**
  + Procesa la lógica de negocio: revisión de documentos, permisos, y efectos personales.
  + Interactúa con bases de datos y servicios externos.
* **Componente Persistencia y Base de Datos**
  + Maneja operaciones CRUD sobre entidades: Turista, Vehículo, RegistroAduanero.
  + Usa SQL para manipulaciones complejas y visualizaciones
* **Componente Autenticación y Seguridad**
  + Control de acceso para funcionarios aduaneros.
  + Cifrado básico de datos sensibles.
* **Componente de Confirmación y Exportación**
* Genera archivos de confirmación.
* Exporta datos relevantes para informes institucionales

* + 1. Descripción diagrama de componente

Cada componente está encapsulado en su propia capa con dependencias explícitas:

* La **Interfaz Web** se comunica solo con la **API del Backend**, sin acceso directo a la base de datos.
* El **componente de Validación** contiene las reglas de negocio y se conecta con servicios externos para verificación.
* La **Persistencia** utiliza scripts SQL optimizados para visualización y agrupación de datos.
* El **componente de Seguridad** se encarga de sesiones y control de usuario interno.
* La **Confirmación** proporciona exportaciones en formatos estándar y genera respuestas en tiempo real.

Esta estructura permite trabajar de manera modular, facilitando el mantenimiento, pruebas y ampliación de funcionalidades

* + 1. Diagrama de paquete

Los paquetes están organizados por contexto funcional:

* paquete.usuario: clases y interfaces para turistas y conductores.
* paquete.vehiculo: modelo de vehículos y lógica de verificación.
* paquete.efectos: registro de efectos personales y validaciones.
* paquete.validacion: motor de reglas del sistema.
* paquete.seguridad: autenticación y autorización de usuarios internos.
* paquete.baseDatos: scripts SQL, DAO y configuración de persistencia.
* paquete.confirmacion: generación de comprobantes y archivos de exportación

* + 1. Descripción diagrama de paquete

La organización en paquetes refleja una arquitectura modular y escalable:

* Cada paquete tiene una responsabilidad bien definida, cumpliendo el principio de cohesión.
* La comunicación entre paquetes se da mediante interfaces públicas, manteniendo acoplamiento bajo.
* La modularidad permite que desarrolladores trabajen en paralelo, asignando paquetes según roles específicos.
* Las dependencias externas (como verificación de documentos) están aisladas en el paquete validacion, facilitando cambios sin afectar el resto del sistema.

Esta estructura también favorece el control de versiones, como se evidencia en las entregas secuenciales y mejoras por iteración del prototipo

* 1. **VISTA DE PROCESOS** (salida vehículo **o** entrada vehículo)
     1. Propósito

La vista de procesos tiene como objetivo describir el comportamiento dinámico del sistema DATET en términos de flujo de control entre actividades. Representa los pasos operativos del sistema desde el inicio del registro hasta su validación y conclusión, ayudando a entender cómo se encadenan las acciones y decisiones durante el ingreso vehicular y declaración de efectos personales.

Esta vista es esencial para visualizar el ciclo de vida de los datos, la interacción entre actores y los puntos de validación, corrección o bifurcación del flujo, reforzando la claridad sobre la lógica funcional del sistema

* + 1. Diagrama de actividad
    2. Descripción diagrama de actividad

El proceso inicia con la selección de la aduana de entrada y país de procedencia. Luego, el turista ingresa sus datos personales, seguidos por los datos del vehículo. Posteriormente, declara los efectos personales transportados.

La información recopilada pasa por una fase de validación donde el sistema verifica que todos los datos estén completos, correctos y no infrinjan normas vigentes. Si todo es válido, se guarda la solicitud y se muestra una confirmación al usuario.

Si se detectan errores, el sistema notifica al turista o conductor, solicita correcciones, y el flujo retorna al ingreso de datos. Este comportamiento dinámico permite retroalimentación inmediata y garantiza integridad del registro antes de su formalización.

El diagrama también permite identificar claramente los puntos críticos del proceso, como validación de documentos, efectos prohibidos o permisos vehiculares, lo que facilita su evaluación, automatización y mejora continua

* 1. **VISTA FÍSICA** (salida vehículo **o** entrada vehículo)
     1. Propósito

La vista física tiene como objetivo representar cómo se despliega el sistema DATET en su entorno operativo real, identificando nodos físicos, componentes de software, canales de comunicación y dependencias tecnológicas. Esta vista es crucial para garantizar que la arquitectura conceptual se traduzca eficazmente en una infraestructura capaz de soportar los requerimientos funcionales y no funcionales del sistema en el contexto de las aduanas chilenas

* + 1. Diagrama de despliegue

El diagrama de despliegue para el escenario de entrada vehicular incluye los siguientes nodos:

* **Estación de Aduana Local**
  + Dispositivo del Funcionario Aduanero (PC o Tablet)
  + Conectado a red LAN institucional
  + Ejecuta interfaz administrativa para validación
* **Servidor Web Institucional (Back-End DATET)**
  + Ejecuta lógica de negocio y API de validación
  + Responsable de procesamiento de solicitudes
  + Alojado en infraestructura segura del gobierno (on-premise o nube privada)
* **Base de Datos Institucional (PostgreSQL / SQL Server)**
  + Almacena registros de turistas, vehículos, efectos y autorizaciones
  + Protegida mediante autenticación y respaldo automático
* **Sistema de Verificación Externa**
  + Conexiones API REST con bases migratorias (RENIEC, servicios de extranjería)
  + Validación de documentos en tiempo real
* **Nube Privada para Backup y Exportación**
* Almacenamiento de confirmaciones y respaldos
* Acceso limitado mediante credenciales institucionales

* + 1. Descripción diagrama de despliegue

El sistema se implementa en una configuración distribuida, donde la interacción del usuario ocurre desde dispositivos ubicados en las estaciones de aduana. Estas estaciones se comunican con el servidor web central a través de una red segura (VPN o LAN institucional).

El servidor procesa la lógica de validación, consulta las bases de datos para verificar documentos, permisos vehiculares y efectos personales, y devuelve los resultados al dispositivo cliente. Además, se conecta vía API a sistemas externos como RENIEC para validaciones de identidad.

Los datos persistentes se almacenan en una base de datos relacional que contiene todas las entidades del sistema, y se replica regularmente en una nube privada para respaldo y generación de informes. El sistema garantiza la protección contra accesos no autorizados y está preparado para escalar horizontalmente en caso de aumento de usuarios simultáneos.

Esta vista demuestra que la arquitectura física cumple con los requisitos de seguridad, disponibilidad, interoperabilidad y mantenibilidad exigidos por el sistema DATET

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

Establecer los atributos de calidad esenciales para el sistema DATET en función de sus objetivos, contexto de uso y requerimientos técnicos. Estos requisitos no funcionales permiten asegurar que el sistema no solo cumpla su función, sino que lo haga de manera eficiente, segura, accesible y mantenible

* 1. Atributos de calidad

| **ATRIBUTO DE CALIDAD** | **DESCRIPCION** | **JUSTIFICACIÓN** |
| --- | --- | --- |
| Usabilidad | La interfaz debe ser intuitiva, clara y fácil de navegar para todos los usuarios | Evaluada con la plantilla de Nielsen, con puntaje superior al 80%. |
| Accesibilidad (WCAG) | El sistema debe cumplir al menos el nivel AA de las WCAG | Probado con herramientas como WAVE; interfaz responsiva y semántica adaptativa |
| Rendimiento | Respuesta del sistema inferior a 2 segundos ante ingreso o validación de datos | Validado mediante pruebas funcionales en prototipo; escalabilidad prevista |
| Seguridad | Protección frente a accesos no autorizados y respaldo de información sensible | Cifrado de datos, control de sesiones y cumplimiento de buenas prácticas OWASP |

* 1. Reglas y criterios de evaluación de calidad

| Atributo | Criterio de Evaluación | Herramienta/Método Aplicado |
| --- | --- | --- |
| Usabilidad | Puntaje ≥80% en evaluación heurística Nielsen | Checklist Nielsen, revisión por pares |
| Accesibilidad | Cumplimiento WCAG nivel AA | Prueba con WAVE, revisión de contraste y estructura semántica |
| Rendimiento | Respuesta del sistema <2 seg. en escenarios con carga duplicada | Pruebas con usuarios simulados, test de carga manual en prototipo |
| Seguridad | Acceso autorizado, respaldo diario, protección frente a DDoS | Autenticación funcional, checklist OWASP, simulación de ataques |

1. **PRINCIPIOS DE DISEÑO APLICADOS** 
   1. Propósito

El objetivo de esta sección es documentar los principios fundamentales de diseño de software aplicados en la construcción del sistema DATET. Estos principios aseguran que el sistema sea robusto, mantenible, escalable y fácil de comprender. Además, respaldan decisiones arquitectónicas alineadas con las mejores prácticas de ingeniería de software, garantizando eficiencia y adaptabilidad en el desarrollo y evolución del sistema

* 1. 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 | Representa entidades mediante modelos simplificados, ocultando complejidad interna | Las clases como Documento, Vehículo y IngresoAduanero encapsulan datos esenciales sin exponer detalles internos |
| Encapsulamiento | Oculta atributos internos y controla acceso mediante métodos | Las clases controlan la modificación de sus atributos mediante setters y validaciones internas |
| Acoplamiento bajo | Los módulos son independientes y se comunican mediante interfaces bien definida | La interfaz web se conecta al backend a través de una API, sin acceso directo a la lógica de negocio |

1. **PROTOTIPO**
   1. Propósito

El prototipo tiene como objetivo validar anticipadamente el diseño de interfaz, los flujos operativos y las funcionalidades clave del sistema DATET antes del desarrollo completo. Sirve como instrumento de iteración centrado en el usuario, permitiendo obtener retroalimentación directa, identificar oportunidades de mejora y reducir riesgos de usabilidad, accesibilidad o ambigüedad en la experiencia de usuario.

Este enfoque asegura que el producto final se alinee con las necesidades reales de los actores involucrados en el ingreso vehicular y la declaración de efectos personales en puntos fronterizos

* 1. Mockups (imágenes con una breve descripción)

Se desarrollaron múltiples mockups que representan las interfaces y flujos más relevantes del sistema. A continuación se describen los principales:

**Pantalla de Selección de Aduana y País de Procedencia:** Formulario inicial donde el turista elige el paso fronterizo y país desde el que ingresa.

**Formulario de Registro de Datos Personales:** Campo para nombre, documento, domicilio temporal, teléfono y correo. Incluye autocompletado para agilizar el proceso.

**Pantalla de Declaración de Efectos Personales:** Selector de tipo de objeto, cantidad y visualización de estado (permitido/prohibido), con retroalimentación visual inmediata.

**Módulo de Validación y Confirmación:** Interfaz resumen de todos los datos ingresados. Muestra mensaje de éxito o alerta en caso de errores detectados.

**Interfaz Administrativa para Funcionario Aduanero:** Tabla de registros recientes, acceso a filtros por nombre, país o tipo de vehículo, y botón para exportación de informes

Los mockups fueron validados por usuarios reales mediante pruebas de usabilidad tipo "think aloud", identificando mejoras en flujos de navegación, retroalimentación visual y consistencia estructural

* 1. Justificar herramientas de prototipado

1. **Figma:** Utilizado para crear prototipos interactivos en alta fidelidad. Se eligió por su facilidad de uso, capacidad de colaboración simultánea, versión gratuita con funciones profesionales y exportación versátil. Permitió iterar rápidamente y compartir versiones para validación externa. **GitHub:** Integrado al proceso para documentar versiones del prototipo (v1, v2, v3) y controlar cambios durante las iteraciones. Su estructura de ramas y commits facilitó el seguimiento del desarrollo por entregas.
2. **Discord (comunicación interna):** Utilizado como canal colaborativo para coordinar ajustes, retroalimentaciones y pruebas internas entre miembros del equipo.

Esta combinación de herramientas permitió construir, evaluar y mejorar el prototipo de forma ágil, con trazabilidad en la evolución del diseño y enfoque centrado en la experiencia de usuario

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

El propósito del control de versiones dentro del sistema DATET es garantizar trazabilidad, organización y seguimiento en las distintas etapas del desarrollo. Permite documentar cada modificación realizada, recuperar versiones previas ante errores o fallos, y coordinar el trabajo colaborativo entre distintos integrantes del equipo técnico.

Esta práctica contribuye a la mantenibilidad del sistema, asegurando que cada iteración del prototipo y del código tenga una justificación técnica, una fecha asociada y una estructura clara que evite conflictos entre desarrolladores

* 1. Lista de verificación

El sistema DATET empleó un modelo de control de versiones **secuencial**, que refleja el progreso por iteraciones del prototipo y del sistema:

* **Versión 1 (v1):** Diseño inicial basado en requerimientos básicos funcionales, con formularios estáticos y estructura preliminar de vistas.
* **Versión 2 (v2):** Incluyó mejoras en la validación de datos, ajustes en interfaz y aplicación de los principios de Nielsen tras evaluación heurística.
* **Versión 3 (v3):** Implementación de control de seguridad, refinamiento de flujos críticos y funcionalidad de exportación.  
   Cada versión está documentada con fecha, entregables definidos y objetivos alcanzados según lo registrado en GitHub.

Este enfoque secuencial facilitó la documentación formal del avance del proyecto, alineado con las entregas académicas y validaciones funcionales

* 1. Análisis y métricas de resultados

**GitHub:** Se seleccionó GitHub como plataforma principal por su capacidad para gestionar múltiples versiones, ramas independientes, historial de cambios y colaboración remota.

* Permitió registrar cada ajuste en el prototipo y arquitectura lógica mediante commits asociados a comentarios explicativos.
* Se utilizó también para compartir el código fuente y documentar revisiones externas durante validaciones.
* Su interfaz gráfica amigable y sincronización con herramientas como Figma y Discord lo hicieron ideal para un entorno académico y ágil.

El uso de esta herramienta contribuyó no sólo a la calidad del proceso técnico, sino también al orden metodológico del proyecto, alineando el control de versiones con la gestión del conocimiento compartido

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

El propósito de esta sección es reflexionar sobre los resultados alcanzados durante el diseño y desarrollo arquitectónico del sistema DATET, sintetizando los aprendizajes metodológicos, técnicos y estratégicos. Las conclusiones permiten evaluar la efectividad del enfoque aplicado, la calidad de los artefactos generados, y las oportunidades de mejora para futuras iteraciones o implementaciones similares

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

El proyecto DATET logró consolidar una arquitectura clara, modular y escalable, basada en el estilo en capas y patrones como MVC y repositorio. La evolución metodológica desde un enfoque tradicional en cascada hacia prácticas ágiles centradas en el usuario permitió validar tempranamente aspectos clave de usabilidad, accesibilidad y rendimiento.

Las herramientas utilizadas (Figma, GitHub, Discord) facilitaron una colaboración ágil y trazable, permitiendo entregas secuenciales (v1, v2, v3) alineadas con validaciones técnicas y heurísticas.

La integración de principios de diseño como cohesión, encapsulamiento y acoplamiento bajo fortaleció la mantenibilidad del sistema.

* 1. Justificar herramientas de versionamiento

Para futuras fases o proyectos similares, se recomienda iniciar desde una etapa de prototipado temprano con participación directa de usuarios finales para acelerar la validación de interfaz y funcionalidades. Mantener el uso de control de versiones estructurado, documentando cada iteración con objetivos claros, fechas y resultados de validaciones.

Incorporar ciclos de mejora continua (PDCA) posterior al despliegue, evaluando métricas reales de uso, errores frecuentes y feedback externo.

Favorecer la modularidad técnica mediante la reutilización de componentes previamente validados y desacoplados, facilitando actualizaciones y nuevas funcionalidades

1. **CONCLUSIONES**

El sistema DATET refleja una evolución metodológica sólida, transitando desde un enfoque tradicional (carta Gantt y arquitectura en capas), hacia prácticas ágiles centradas en el usuario (prototipado, heurísticas, control de versiones). Su arquitectura facilita la validación de datos migratorios y vehiculares, asegurando confiabilidad y calidad. La documentación completa permite escalar y mantener el sistema según estándares técnicos

1. **BIBLIOGRAFÍA**

Jakob Nielsen (Usabilidad Heurística) WCAG Guidelines v2.1

Documentación Figma

OWASP Secure Design Principles

Guías de Arquitectura Software – IEEE/ISO

1. **ANEXOS**
   1. Planilla de requerimientos

Requerimientos agrupados por actor: Turistas, Conductores, Funcionarios de Aduana, Transportistas. Requerimientos funcionales: ingreso de datos, validación vehicular, confirmación de registro, exportación de información.

Requerimientos no funcionales: rendimiento (<2 seg.), seguridad, disponibilidad (99.95%), accesibilidad (WCAG), portabilidad y mantenibilidad.

Categorías y codificación de cada requerimiento (por ejemplo, RF-001, RNF-003).

Criterios de aceptación y prioridad estimada de implementación.

Esta planilla sirvió como base para la definición de los casos de uso, el diseño del prototipo y la especificación de vistas arquitectónicas

* 1. Carta Gantt

Desglose de actividades por fase del ciclo de vida (Análisis, Diseño, Desarrollo, Validación, Despliegue, Cierre). Duración estimada de cada tarea, fechas de inicio y término.

Asignación de responsables y recursos por actividad (Nicolás, Felipe, equipo técnico).

Indicadores de control como hitos, entregables y dependencias.

Relación directa entre tareas del Gantt y los componentes arquitectónicos descritos en el DAS.