



**UNIVERSIDAD POLITÉCNICA DE VICTORIA**

---

**DISEÑO E IMPLEMENTACIÓN DE UN  
MIDDLEWARE PARA LA INTEROPERABILIDAD  
ENTRE LA PLATAFORMA NEZ Y EL  
CONCENTRADOR DE SERVICIOS JUB**

**T E S I N A  
QUE PARA OBTENER EL GRADO DE  
INGENIERÍA EN TECNOLOGÍAS DE LA  
INFORMACIÓN**

**PRESENTA:  
JOSE MANUEL ALONSO CEPEDA**

**DIRECTOR  
DR. MARCO AURELIO NUÑO MAGANDA  
CO-DIRECTOR**

**DR. JOSÉ LUIS GONZÁLEZ COMPEÁN  
ORGANISMO RECEPTOR  
CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS  
AVANZADOS DEL INSTITUTO POLITÉCNICO  
NACIONAL (CINVESTAV) UNIDAD TAMAULIPAS  
CIUDAD VICTORIA, TAMAULIPAS, JUNIO DE 2021**

---



Ciudad Victoria,  
Tamaulipas, a  
**23 de**  
**Septiembre de**  
**2025**

**CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL INSTITUTO  
POLITECNICO NACIONAL UNIDAD TAMAULIPAS (CINVESTAV)**  
**DR. JOSÉ LUIS GONZÁLEZ COMPÉAN**  
**PRESENTE**



La Universidad Politécnica de Victoria tiene a bien presentar a **ALONSO CEPEDA JOSE MANUEL** estudiante del programa académico de **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, con número de matrícula **2130207** y seguro facultativo IMSS número **46-18-03-6197-7**; quién deberá realizar su práctica profesional de **ESTADÍA**, a partir del **01 de Septiembre de 2025** al **05 de Diciembre de 2025**, con duración de **600 horas** desarrollando el proyecto **"Diseño e implementación de un middleware para la interoperabilidad entre la plataforma Nez y el concentrador de servicios Jub"** que le fué asignado.

Al concluir se le extenderá la carta de liberación al evaluarlo satisfactoriamente y además le solicitaremos su valiosa opinión respondiendo el formulario que se le enviará por mail, referente al desempeño del practicante, la información que nos proporcione, es de vital importancia para la mejora de los programas académicos que ofrece la Universidad Politécnica de Victoria para la formación de profesionistas altamente especializados.

El practicante deberá cumplir con el Reglamento Interno aplicable al personal en su centro de trabajo.

Sin otro particular.

**ATENTAMENTE**

**OTHÓN CANO GARZA**  
**DIRECTOR DE VINCULACIÓN**

C.C.P. MARCO AURELIONUÑOMAGANDA  
ASESOR INSTITUCIONAL



**UNIVERSIDAD POLITÉCNICA DE VICTORIA**

Av. Nuevas Tecnologías 5902  
Parque Científico y Tecnológico de Tamaulipas  
Carretera Victoria Soto La Marina Km. 5.5  
Cd. Victoria, Tamaulipas. C.P. 87138

Tel: (834) 1711100 al 10  
[www.upvictoria.edu.mx](http://www.upvictoria.edu.mx)



# **CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITECNICO NACIONAL UNIDAD TAMAULIPAS (CINVESTAV)**

Victoria Tamaulipas, a 23 de Septiembre del 2025  
Asunto: Carta de Aceptación

**M.A OTHÓN CANO GARZA**  
**DIRECTOR DE VINCULACIÓN**  
**UNIVERSIDAD POLITÉCNICA DE VICTORIA**  
**PRESENTE**

Hacemos de su conocimiento que hemos aceptado al estudiante **ALONSO CEPEDA JOSE MANUEL** del programa académico de **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, con número de matrícula **2130207** de la Universidad Politécnica de Victoria, para realizar su **ESTADÍA** en nuestra empresa **CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITECNICO NACIONAL UNIDAD TAMAULIPAS (CINVESTAV)**, durante el periodo comprendido del día **01 de Septiembre de 2025** al **05 de Diciembre de 2025**, el estudiante estará colaborando en el proyecto **"Diseño e implementación de un middleware para la interoperabilidad entre la plataforma Nez y el concentrador de servicios Jub"** con una carga horaria total de **600 horas**

Al concluir satisfactoriamente sus prácticas profesionales, se le entregará al estudiante su carta de liberación debidamente formalizada.

Con el objetivo de colaborar en la mejora de los programas académicos de la Universidad Politécnica de Victoria, estamos de acuerdo en responder a la brevedad posible el formulario de evaluación del desempeño del practicante previo a emitir la liberación de **ESTADÍA**.

Sin otro particular.

**ATENTAMENTE**  
**DR. JOSÉ LUIS GONZÁLEZ COMPÉAN**  
**ASESOR EMPRESARIAL**



Ing. Juan Camaney de Alba Rojas  
PRESENTE

INSERTAR AQUÍ  
DOCUMENTO  
DE CARTA  
DE LIBERACION  
DEBIDAMENTE  
FIRMADO



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



**Tamaulipas**  
Gobierno del Estado



Secretaría  
de Educación



## CARTA DE ACEPTACIÓN DEL DOCUMENTO PARA SU IMPRESIÓN

Cd. Victoria, Tamaulipas a 26 de Abril de 2025

Jose Manuel Alonso Cepeda  
PRESENTE

Le comunico que el Programa Académico de Ingeniería en Tecnologías de la Información le ha otorgado la autorización para la impresión de su Tesina de Estadía Práctica cuyo título es:

**Diseño e implementación de un middleware para la  
interoperabilidad entre la plataforma Nez y el concentrador de  
servicios Jub**  
**ATENTAMENTE**

---

Dr. Marco Aurelio Nuño Maganda  
ASESOR INSTITUCIONAL

c.c.p Director de programa académico

**UNIVERSIDAD POLITÉCNICA DE VICTORIA**

Av. Nuevas Tecnologías 5902  
Parque Científico y Tecnológico de Tamaulipas  
Carretera Victoria Soto La Marina Km. 5.5  
Cd. Victoria, Tamaulipas. C.P. 87138

Tel: (834) 1711100 al 10  
[www.upvictoria.edu.mx](http://www.upvictoria.edu.mx)



## EVALUACIÓN DE ESTADÍA

### Rúbrica para evaluación de la presentación y el reporte de estadía

Nombre del alumno: **JOSE MANUEL ALONSO CEPEDA**

Calificación final: \_\_\_\_\_

Periodo: **MAYO-AGOSTO 2025**

Ponderación	Aspecto a Evaluar	Competente 10	Independiente 9	Básico Avanzado 8	No Competente 5
40	Resultados y Actividades	Estrechamente relacionados al perfil de egreso de su programa académico	Parcialmente relacionados al perfil de egreso de su programa académico	Escasamente relacionados al perfil de egreso de su programa académico	Escasamente relacionados al perfil de egreso de su programa académico
30	Exposición de las actividades de la estadía	Detalladas y sustentadas con respecto a los resultados que se obtuvieron	Detalladas y sustentadas parcialmente con respecto a los resultados que se obtuvieron	Detalladas parcialmente con respecto a los resultados que se obtuvieron	Detalladas escasamente con respecto a los resultados que se obtuvieron
10	Material visual Lenguaje verbal	Uso el lenguaje y la terminología apropiadas; El material visual está organizado, adecuado y suficiente	Uso el lenguaje y la terminología apropiadas El material visual está parcialmente organizado y es suficiente	Uso el lenguaje y la terminología son parcialmente apropiadas; El material visual está parcialmente organizado y es suficiente	Uso el lenguaje y terminología es inapropiado; El material visual no está organizado y es insuficiente
10	Exposición en Idioma Inglés	Pronunciation is clear so language is easily understood (2.5) Uses fluent connected speech, occasionally disrupted by search for correct form of expression (2.5) Uses topic related vocabulary without problems (2.5) Responds to questions using varied and descriptive vocabulary and language structures (2.5)	Pronunciation is understandable, but there are slight errors (2.25) Speech is connected but frequently disrupted by search for correct form of expression (2.25) Uses some topic related vocabulary sufficient to communicate ideas (2.25) Responds to questions using simple but accurate vocabulary and language structures (2.25)	Pronunciation is understandable most of the time, marked native accent and many errors (2) Speaks with simple sentences, sometimes not connected, but is understood (2) Uses basic vocabulary to communicate ideas (2) Partly responds to simple questions, with limited vocabulary and language structures (2)	Pronunciation makes language very difficult to understand (1) Uses one-word/two-word utterances (1) Unable to communicate ideas due to lack of vocabulary (1) Uses isolated words or sentence fragments to respond to questions (1)
5	Respuesta a los cuestionamientos de los evaluadores	Clara y satisfactoria	Clara y parcialmente satisfactoria	Clara e insuficiente	Confusa e insuficiente
5	Autorización de tesina en tiempo y forma	Presenta en tiempo y forma	Presenta en tiempo y forma con la mayoría de requerimientos solicitados	Presenta en tiempo y con algunas limitantes de los requerimientos solicitados.	Presenta fuera de tiempo y con los mínimos requerimientos solicitados.

Dr. Marco Aurelio Nuño Maganda  
ASESOR INSTITUCIONAL

Dr. Hiram Herrera Rivas  
EVALUADOR

EVALUADOR DE INGLÉS



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



**Tamaulipas**  
Gobierno del Estado



Secretaría  
de Educación



## REGISTRO DE EVALUACIÓN DE EXPOSICIÓN DE ESTADÍA

Siendo las 10:00 horas del día 11 de Agosto de 2021, el alumno **Jose Manuel Alonso Cepeda**, del programa académico **Ingeniería en Tecnologías de la Información**, con matrícula **1730505**, presentó la exposición de la estadía realizada durante el cuatrimestre **Mayo-agosto 2025**, en la **CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL (CINVESTAV) UNIDAD TAMAULIPAS**, con el proyecto titulado **Diseño e implementación de un middleware para la interoperabilidad entre la plataforma Nez y el concentrador de servicios Jub**.

Una vez concluido el proceso de evaluación, y con base a la rúbrica establecida para éste propósito, se determina que la calificación de la estadía es \_\_\_\_\_.

---

Dr. Marco Aurelio Nuño Maganda  
ASESOR INSTITUCIONAL

---

Dr. Hiram Herrera Rivas  
EVALUADOR

---

EVALUADOR DE INGLÉS

**UNIVERSIDAD POLITÉCNICA DE VICTORIA**

Av. Nuevas Tecnologías 5902  
Parque Científico y Tecnológico de Tamaulipas  
Carretera Victoria Soto La Marina Km. 5.5  
Cd. Victoria, Tamaulipas. C.P. 87138

Tel: (834) 1711100 al 10  
[www.upvictoria.edu.mx](http://www.upvictoria.edu.mx)



## Agradecimientos



## Resumen

El avance de la ciencia de datos y el aprendizaje profundo depende críticamente de la capacidad para integrar sistemas de software heterogéneos y especializados. Plataformas como *Nez*, un *framework* para el procesamiento de datos a gran escala basado en el modelo *PuzzleMesh*, y *Jub*, un concentrador de servicios para el monitoreo de fenómenos atmosféricos, ofrecen capacidades potentes pero operan en silos aislados, limitando su potencial sinérgico. Este trabajo aborda el desafío de la interoperabilidad entre estos dos sistemas mediante el diseño e implementación de un *middleware* de acoplamiento ligero.

La solución propuesta se basa en una arquitectura de API *RESTful*, que actúa como un puente de comunicación estandarizado, permitiendo que *Jub* invoque los servicios de procesamiento avanzado de *Nez* de manera transparente y eficiente. El *middleware* implementado no solo facilita el intercambio de datos y procesos en tiempo real, sino que también establece las bases para la creación de una malla de servicios de ciencia de datos unificada, gestionando las operaciones de almacenamiento a través del cliente *MictlanX*.

El prototipo fue desarrollado utilizando el *framework* *FastAPI* de Python, seleccionado por su alto rendimiento y sus capacidades para la rápida creación de APIs robustas. La validación del sistema se realizó a través de un caso de uso de procesamiento de imágenes, demostrando la viabilidad y eficacia del *middleware* como catalizador para la integración de sistemas complejos en entornos de investigación científica.

**Palabras clave:** Middleware, Interoperabilidad, Malla de Servicios, API REST, FastAPI, Nez, Jub, Sistemas Distribuidos.

## Summary

The advancement of data science and deep learning critically depends on the ability to integrate heterogeneous and specialized software systems. Platforms such as *Nez*, a framework for large-scale data processing based on the *PuzzleMesh* model, and *Jub*, a service hub for monitoring atmospheric phenomena, offer powerful capabilities but operate in isolated silos, limiting their synergistic potential. This work addresses the challenge of interoperability between these two systems through the design and implementation of a lightweight *middleware*.

The proposed solution is based on a RESTful API architecture, which acts as a standardized communication bridge, allowing *Jub* to invoke *Nez*'s advanced processing services transparently and efficiently. The implemented *middleware* not only facilitates the real-time exchange of data and processes but also lays the foundation for creating a unified data science service mesh, managing storage operations through the *MictlanX* client.

The prototype was developed using the Python *FastAPI* framework, chosen for its high performance and its capabilities for the rapid creation of robust APIs. The system's validation was conducted through an image processing use case, demonstrating the feasibility and effectiveness of the *middleware* as a catalyst for integrating complex systems in scientific research environments.

**Keywords:** Middleware, Interoperability, Service Mesh, REST API, FastAPI, Nez, Jub, Distributed Systems.

# Índice

<b>Agradecimientos</b>	<b>VII</b>
<b>Resumen</b>	<b>VIII</b>
<b>Summary</b>	<b>IX</b>
<b>Índice</b>	<b>X</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Definición del problema y justificación del proyecto . . . . .	1
1.2. Objetivo General . . . . .	3
1.3. Objetivos Particulares . . . . .	3
1.4. Alcances y limitaciones del Proyecto . . . . .	3
1.5. Organización del Documento de Tesina . . . . .	4
<b>2. Marco Teórico</b>	<b>5</b>
2.1. Arquitecturas Orientadas a Microservicios y Mallas de Servicios (Service Mesh)	5
2.2. El Modelo Conceptual de PuzzleMesh . . . . .	5
2.2.1. La Pieza de Software (P): Unidad Fundamental de Procesamiento . .	5
2.2.2. El Rompecabezas (R): Composición de Flujos de Trabajo . . . . .	5
2.2.3. El Metarompecabezas ( $\Omega$ ): Habilitando Flujos de Datos Inter-Sistemas	6
2.2.4. La Malla de Servicios ( $\Psi$ ) en PuzzleMesh . . . . .	6
2.3. Middleware como Catalizador de la Interoperabilidad en Sistemas Distribuidos	6
2.4. Plataformas Involucradas . . . . .	6
2.5. Aprendizaje Profundo como Servicio en el Procesamiento de Imágenes y Señales	7
<b>Índice de figuras</b>	<b>8</b>
<b>Índice de cuadros</b>	<b>9</b>
<b>Índice de algoritmos</b>	<b>10</b>
<b>Referencias</b>	<b>11</b>

# 1. Introducción

Vivimos en una era definida por la explosión de datos, comúnmente denominada la era del *Big Data*. Campos tan diversos como la genómica, la astrofísica, las finanzas y las ciencias de la Tierra están generando volúmenes de información a una escala sin precedentes [1]. En particular, el sector de la salud y la observación geoespacial han emergido como dos de los dominios más prolíficos en la generación de datos complejos, como imágenes médicas de alta resolución y datos satelitales multiespectrales. El análisis de esta vasta cantidad de información es fundamental para la toma de decisiones críticas, desde el diagnóstico temprano de enfermedades hasta el monitoreo del cambio climático. Para enfrentar este desafío, han surgido tecnologías transformadoras como el cómputo en la nube, que proporciona la infraestructura escalable necesaria, y el aprendizaje profundo (*deep learning*), que ofrece métodos potentes para extraer conocimiento de estos datos masivos [2].

Sin embargo, la creciente especialización tecnológica ha llevado a una fragmentación significativa. Las organizaciones, tanto en el ámbito académico como en el industrial, desarrollan o adoptan sistemas de software altamente especializados, optimizados para tareas muy concretas. Si bien esta especialización es beneficiosa para resolver problemas específicos, a menudo conduce a la creación de “silos de datos y procesamiento”. Estos silos son ecosistemas tecnológicos aislados que, aunque potentes individualmente, carecen de la capacidad de comunicarse e interactuar entre sí de manera fluida. Esta falta de interoperabilidad se ha convertido en uno de los desafíos más significativos de la ingeniería de software moderna en sistemas distribuidos [3]. La heterogeneidad en lenguajes de programación, protocolos de comunicación y formatos de datos impide la creación de flujos de trabajo integrados, limitando el potencial científico y operativo que podría surgir de la combinación sinérgica de estas herramientas.

Para superar esta barrera, la industria ha adoptado el uso de *middleware* como una solución estratégica. El *middleware* actúa como un “pegamento de software”, una capa de abstracción que se sitúa entre aplicaciones dispares y les permite comunicarse de manera estandarizada, ocultando la complejidad subyacente de la red y los sistemas operativos. En el contexto de la investigación científica, el *middleware* es crucial para la construcción de flujos de trabajo científicos (*scientific workflows*), que permiten automatizar y orquestar secuencias complejas de procesamiento y análisis de datos a través de múltiples sistemas distribuidos [4]. Este enfoque no solo mejora la eficiencia y la reproducibilidad de la investigación, sino que también fomenta la colaboración al permitir que diferentes herramientas y servicios se combinen de formas novedosas.

## 1.1. Definición del problema y justificación del proyecto

En el Centro de Investigación y de Estudios Avanzados (CINVESTAV) Unidad Tamaulipas, el desafío de la interoperabilidad se manifiesta de manera concreta en la interacción (o la falta de ella) entre dos plataformas de software estratégicas. Por un lado, se encuentra **Nez**, un *framework* avanzado para el procesamiento de datos a gran escala. *Nez* no es solo una herramienta, sino la implementación del innovador modelo arquitectónico *PuzzleMesh*, que concibe las aplicaciones como “piezas de rompecabezas” modulares y autocontenidas que

pueden ser ensambladas para construir estructuras de procesamiento complejas y dinámicas. Esta arquitectura ha sido validada con éxito en dominios de alto impacto como el análisis de tomografías para diagnóstico médico y el procesamiento de imágenes satelitales para la observación de la Tierra.

La plataforma *Nez* es, de hecho, un componente clave dentro de un ecosistema tecnológico más amplio y ambicioso denominado **Muyal-Ilal**, un servicio en la nube diseñado por investigadores de la Universidad Autónoma Metropolitana para la gestión, aseguramiento, intercambio y procesamiento de grandes volúmenes de datos, con un enfoque particular en el sector salud [5]. Dentro de este ecosistema, *Nez* funciona como la plataforma de construcción de sistemas, guiando al personal de salud y de tecnologías de la información en la creación de flujos de trabajo de manera intuitiva, utilizando un enfoque de “bloques de Lego” para ensamblar herramientas de inteligencia artificial. Este ecosistema se complementa con otros servicios especializados como *Muyal-Painal*, dedicado al almacenamiento y distribución de datos, y *Muyal-Xelhua*, enfocado en la analítica de datos masivos. La filosofía de *Nez* es, por tanto, facilitar la creación de sistemas complejos sin requerir conocimientos avanzados de programación, resolviendo dependencias tecnológicas a través de su arquitectura modular.

Por otro lado, se encuentra **Jub**, un concentrador y distribuidor de datos diseñado específicamente para el monitoreo de fenómenos atmosféricos, actuando como un punto central para la ingesta y distribución de flujos de datos en tiempo real. Adicionalmente, un tercer componente, *MictlanX*, gestiona las operaciones de almacenamiento subyacentes para este ecosistema.

El problema central que aborda esta tesina es la ausencia de un mecanismo de comunicación nativo y estandarizado entre *Nez* y *Jub*. A pesar de sus capacidades evidentemente complementarias, estas dos plataformas operan en silos tecnológicos. Esta desconexión impone una barrera significativa: los usuarios de *Jub* no pueden invocar de manera programática y automatizada las potentes capacidades de análisis y aprendizaje profundo de *Nez*, y este último no puede ser alimentado directamente por los flujos de datos curados y distribuidos por *Jub*. Este aislamiento tecnológico no solo representa una subutilización de los recursos existentes, sino que impide activamente la formación de una malla de servicios de ciencia de datos (*Service Mesh*) cohesiva y eficiente, un paradigma arquitectónico moderno para gestionar la comunicación entre microservicios de manera fiable y segura [6].

La justificación de este proyecto es, por tanto, clara y apremiante. El desarrollo de un *middleware* que actúe como un puente de comunicación estandarizado es un paso indispensable para romper estos silos. Al permitir una integración transparente y ligera, este *middleware* no solo conectará dos sistemas, sino que sentará las bases para una arquitectura de servicios distribuidos mucho más ambiciosa y escalable. Permitirá desbloquear nuevas y valiosas líneas de investigación y aplicación al combinar el análisis de datos atmosféricos con el procesamiento avanzado de imágenes y otras formas de análisis computacional. En resumen, este trabajo es un paso fundamental y estratégico para maximizar el valor y el impacto de los activos tecnológicos existentes en la institución, alineándolos con las mejores prácticas de la ingeniería de software contemporánea.

## 1.2. Objetivo General

Diseñar e implementar un *middleware* de acoplamiento ligero que permita la interoperabilidad entre la plataforma *Nez* y el concentrador de servicios *Jub*, habilitando la creación de una malla de servicios de ciencia de datos y aprendizaje profundo para el procesamiento distribuido de datos e imágenes de distintos dominios.

## 1.3. Objetivos Particulares

- Diseñar la arquitectura del *middleware* para permitir la comunicación eficiente entre las plataformas *Nez* y *Jub*, basándose en principios de arquitecturas orientadas a servicios.
- Implementar el *middleware* para lograr un acoplamiento ligero que habilite el intercambio de datos y la invocación de procesos en tiempo real.
- Integrar y validar el flujo de trabajo completo mediante la implementación de servicios de procesamiento distribuido de datos e imágenes, utilizando algoritmos de aprendizaje profundo.
- Mejorar las interfaces gráficas de usuario existentes en la plataforma *Jub* para incorporar la nueva funcionalidad, facilitando la gestión y visualización de los datos procesados.
- Elaborar la documentación técnica detallada y los manuales de usuario del sistema para garantizar su correcto uso, mantenimiento y extensibilidad futura.

## 1.4. Alcances y limitaciones del Proyecto

El alcance de este proyecto se centra en la entrega de un prototipo funcional y bien documentado del *middleware* de interoperabilidad. Este prototipo será capaz de recibir solicitudes desde *Jub*, orquestar la ejecución de procesos de análisis en *Nez* y gestionar el flujo de los resultados de vuelta al sistema de origen. El proyecto incluye la validación de la solución completa a través de un caso de uso específico de procesamiento de imágenes, así como las mejoras necesarias a la interfaz de usuario de *Jub* para integrar esta nueva funcionalidad de manera intuitiva.

No obstante, el proyecto presenta las siguientes limitaciones en su fase actual:

- El prototipo se validará con un conjunto limitado y bien definido de casos de uso, sin abarcar la totalidad de las capacidades potenciales de *Nez* y *Jub*.
- El manejo de errores en el *middleware* se centrará en la notificación de fallos y el registro de eventos, sin implementar mecanismos avanzados de resiliencia como reintentos automáticos con retroceso exponencial (*exponential backoff*) o patrones de *circuit breaking*.
- La solución no incluirá una integración nativa con herramientas de monitoreo y observabilidad de nivel de producción, como *Prometheus* para métricas o *Grafana* para

visualización, aunque su arquitectura permitirá dicha integración en el futuro.

- El modelo para consultar el estado de los trabajos de procesamiento se basará en un mecanismo de sondeo (*polling*) iniciado por el cliente, en lugar de un sistema de notificaciones proactivas basadas en eventos (como *webhooks*), que podría ser más eficiente en ciertos escenarios.

## 1.5. Organización del Documento de Tesina

Este documento se organiza en seis capítulos para presentar de manera clara, lógica y estructurada el desarrollo completo del proyecto.

El **Capítulo 2** establece el **Marco Teórico**, donde se exploran y definen los conceptos fundamentales que sustentan este trabajo. Se abordan en profundidad temas como las arquitecturas de mallas de servicios (*Service Mesh*), la teoría y práctica del *middleware*, los desafíos de la interoperabilidad en sistemas heterogéneos, y se describen las tecnologías clave involucradas, con especial énfasis en el modelo conceptual *PuzzleMesh* de la plataforma *Nez*.

El **Capítulo 3** se dedica por completo al **Diseño Arquitectónico** del *middleware*. En esta sección se detallan los requisitos funcionales y no funcionales, se justifican las decisiones de diseño cruciales, como la elección de una API REST como interfaz de comunicación, y se especifica formalmente el contrato de la API, incluyendo *endpoints*, formatos de datos y códigos de estado.

El **Capítulo 4** describe la **Implementación** del prototipo. Este capítulo abarca la selección del *stack* tecnológico (lenguajes, *frameworks*), la estructura del código fuente, las decisiones de diseño a bajo nivel y las estrategias de empaquetado y despliegue mediante contenerización con *Docker*.

En el **Capítulo 5** se presentan las **Pruebas y Validación** del sistema. Aquí se detalla el escenario de prueba configurado, se exponen los resultados de las pruebas de integración que demuestran la correcta comunicación entre *Jub* y *Nez*, y se realiza un análisis de rendimiento preliminar para evaluar la latencia y el *throughput* del *middleware*.

Finalmente, el **Capítulo 6** expone las **Conclusiones y el Trabajo Futuro**. En esta última sección se resumen las contribuciones más importantes del proyecto, se reflexiona sobre los resultados obtenidos y se delinean posibles líneas de investigación y desarrollo para extender y mejorar el trabajo realizado, como la implementación de mecanismos de seguridad más robustos o la integración con sistemas de orquestación avanzados.

## 2. Marco Teórico

Este capítulo establece el marco conceptual y tecnológico sobre el cual se fundamenta el proyecto. Se definen los principios de arquitecturas de software modernas, se formaliza el modelo PuzzleMesh, se introduce el rol del middleware como catalizador de la interoperabilidad y se describen las plataformas tecnológicas involucradas.

### 2.1. Arquitecturas Orientadas a Microservicios y Mallas de Servicios (Service Mesh)

Las arquitecturas de microservicios estructuran una aplicación como una colección de servicios pequeños, autónomos y débilmente acoplados. Para gestionar la complejidad en la comunicación entre estos servicios, ha surgido el concepto de Malla de Servicios (Service Mesh). Una malla de servicios es una capa de infraestructura de software dedicada que facilita la comunicación entre microservicios en una arquitectura distribuida. Actúa como un intermediario que gestiona el tráfico de red, proporcionando funcionalidades críticas como el descubrimiento de servicios, el balanceo de carga, la encriptación, la observabilidad y la resiliencia. Su arquitectura se compone de un **Plano de Datos**, formado por proxies ligeros (sidecars) que interceptan el tráfico, y un **Plano de Control**, que administra y configura dichos proxies para aplicar políticas de enrutamiento y seguridad.

### 2.2. El Modelo Conceptual de PuzzleMesh

El modelo PuzzleMesh es un marco formal para construir estructuras de procesamiento agnósticas a la infraestructura, utilizando una metáfora de rompecabezas. Este modelo se alinea con los principios de las mallas de servicios al promover la modularidad y la gestión centralizada de componentes reutilizables.

#### 2.2.1. La Pieza de Software (P): Unidad Fundamental de Procesamiento

La unidad básica es la Pieza de Software (P), un artefacto autocontenido que encapsula una aplicación con todos sus componentes necesarios para su despliegue y ejecución. Una pieza funciona como una «caja negra» con capas definidas para el acceso, interfaces de entrada/-salida (E/S), la aplicación, metadatos y dependencias.

#### 2.2.2. El Rompecabezas (R): Composición de Flujos de Trabajo

Un Rompecabezas (R) es una estructura de procesamiento creada al unir un conjunto de Piezas. El orden de ejecución y el flujo de datos se definen mediante un **Grafo Acíclico Dirigido (DAG)**, donde los nodos son las piezas y los vértices representan las dependencias de datos entre ellas. Un rompecabezas representa un flujo de trabajo completo, como un pipeline de procesamiento de datos.



### 2.2.3. El Metarompecabezas ( $\Omega$ ): Habilitando Flujos de Datos Inter-Sistemas

PuzzleMesh introduce el concepto de Metarompecabezas ( $\Omega$ ) al encadenar múltiples rompecabezas. Esto permite la creación de flujos de datos complejos que pueden abarcar diferentes departamentos o incluso distintas organizaciones (flujos interinstitucionales).

### 2.2.4. La Malla de Servicios ( $\Psi$ ) en PuzzleMesh

Finalmente, todas las Piezas, Rompecabezas y Metarompecabezas se incorporan en una Malla de Servicios ( $\Psi$ ). En este contexto, la malla actúa como un repositorio o catálogo centralizado desde el cual las organizaciones pueden seleccionar, componer y reutilizar estos bloques de construcción para crear nuevos servicios.

## 2.3. Middleware como Catalizador de la Interoperabilidad en Sistemas Distribuidos

El objetivo central de este proyecto es lograr la interoperabilidad entre Nez y Jub, meta que se alcanza a través de un middleware. El middleware es una capa de software que se sitúa entre diferentes aplicaciones, funcionando como un puente o una «capa de traducción oculta» para facilitar su comunicación y el intercambio de datos. Su función es abstraer la complejidad de la comunicación en un entorno distribuido.

La interoperabilidad es la capacidad de dos o más sistemas de intercambiar información y utilizarla de manera efectiva. Se distinguen varios niveles, desde el **fundacional** (intercambio básico de datos) hasta el **semántico** (comprensión compartida del significado de los datos). El middleware es el mecanismo técnico que permite alcanzar la interoperabilidad, proporcionando servicios estandarizados como la transformación de datos y el enrutamiento de mensajes.

## 2.4. Plataformas Involucradas

El ecosistema del proyecto se compone de tres plataformas desarrolladas en el **CINVESTAV Tamaulipas**:

- **Nez:** Es el framework para el procesamiento de datos a gran escala que implementa el modelo conceptual de PuzzleMesh. Ha sido utilizado con éxito en dominios como el análisis de imágenes de tomografía y la observación de la Tierra.
- **Jub:** Es un concentrador y distribuidor de datos diseñado para el monitoreo de fenómenos atmosféricos, actuando como un gestor de datos y punto de acceso para servicios de ciencia de datos.
- **MictlanX:** Es la plataforma designada para gestionar las operaciones de almacenamiento dentro del ecosistema integrado, manejando la persistencia de los datos que fluyen entre Jub y Nez.

## 2.5. Aprendizaje Profundo como Servicio en el Procesamiento de Imágenes y Señales

La motivación final para integrar estas plataformas es habilitar el uso de algoritmos avanzados de Aprendizaje Profundo (*Deep Learning*). Este subcampo del aprendizaje automático utiliza redes neuronales con múltiples capas para aprender representaciones de datos con altos niveles de abstracción. Esta tecnología es altamente eficaz en el reconocimiento de patrones complejos, especialmente en el procesamiento de imágenes y señales. La integración de Nez y Jub a través del middleware permitirá ofrecer estas capacidades como un servicio (*Deep Learning as a Service – DLaaS*), donde los usuarios de Jub pueden solicitar análisis complejos que se ejecutarán en la infraestructura de alto rendimiento de Nez.

## Índice de figuras

## Índice de cuadros

## Índice de algoritmos

## Referencias

- [1] Martin Sudmanns et al. «Big Earth Observation Data-A Review of the Challenges and Opportunities for Big Data Processing in the Copernicus Era». En: *Remote Sensing* 11.21 (2019), pág. 2577. DOI: [10.3390/rs11212577](https://doi.org/10.3390/rs11212577).
- [2] Geert Litjens et al. «A survey on deep learning in medical image analysis». En: *Medical Image Analysis* 42 (2017), págs. 60-88. DOI: [10.1016/j.media.2017.07.005](https://doi.org/10.1016/j.media.2017.07.005).
- [3] Kennedy O. Ondimu y Geoffrey M. Muketha. «Challenges in Achieving Interoperability in Distributed Systems: a Survey of Literature». En: *International Journal of Computer Applications* 168.10 (2017), págs. 33-39. DOI: [10.5120/ijca2017914539](https://doi.org/10.5120/ijca2017914539).
- [4] Carole A. Goble y David De Roure. «Scientific workflow systems—for whom and for what?» En: *Journal of Grid Computing* 5.3 (2007), págs. 249-255. DOI: [10.1007/s10723-007-9078-8](https://doi.org/10.1007/s10723-007-9078-8).
- [5] J. L. González-Compean et al. *Muyal-Nez, una plataforma de construcción de sistemas de ciencias de datos médicos para procesos de toma de decisiones en el sector salud*. <https://contactos.izt.uam.mx/index.php/contactos/article/view/297/169>. 2020.
- [6] Lee Calcote y Zack Butcher. *Istio: Up and Running: Using a Service Mesh to Connect, Secure, Control, and Observe*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2019.