Universidad de la República
Facultad de Ingeniería

Arquitectura para el Procesamiento, Modelado y Visualización de Datos de Redes Ópticas

Memoria de proyecto presentada a la Facultad de Ingeniería de la Universidad de la República por

Ignacio Faget, Joaquin Ramirez

en cumplimiento parcial de los requerimientos para la finalización de la carrera de Ingeniería en Sistemas de Comunicación e Ingeniería Eléctrica.

Dirección de Tesis

Claudina Rattaro ..... Universidad de la República
Alberto Castro ..... Universidad de la República

Tribunal

Adriana Marotta ..... Universidad de la República
Gabriel Gómez ..... Universidad de la República
Sebastián Maldonado ..... Universidad de la República
Alfredo Rodríguez ..... ANTEL

Montevideo
viernes 20 febrero, 2026

Arquitectura para el Procesamiento, Modelado y Visualización de Datos de Redes Ópticas, Ignacio Faget, Joaquin Ramirez.

Esta tesis fue preparada en LaTeX usando la clase iietesis (v1.2).
Contiene un total de 158 páginas.
Compilada el viernes 20 febrero, 2026.
http://iie.fing.edu.uy/

Agradecimientos

A nuestros tutores, Claudina Rattaro y Alberto Castro, por el acompañamiento, la guía y la disponibilidad durante todo el proyecto.

Al equipo de Red de Transporte de ANTEL, por su tiempo y disposición para responder consultas, compartir experiencia y facilitarnos información y material para el desarrollo del trabajo.

A la Universidad de la República, al Instituto de Ingeniería Eléctrica y al Instituto de Computación, por brindar el marco y los recursos necesarios para llevar adelante este trabajo.

Y a nuestras familias y amigos, por el apoyo y el acompañamiento a lo largo de todo el proceso.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

En el marco de este proyecto de grado se aborda el desafío de mejorar la gestión y el análisis de datos en redes ópticas, tomando como caso de estudio la red de transporte óptica de Administración Nacional de Telecomunicaciones (ANTEL) basada en tecnología del fabricante Infinera. El trabajo se desarrolla en el contexto de una colaboración entre el grupo de investigación Optical Mobile Network Integration (OMNI) (integrado por investigadores de los institutos de Computación e Ingeniería Eléctrica) y el área de Red de Transporte de ANTEL, con el objetivo de apoyar la operativa de la red mediante técnicas de modelado, procesamiento, visualización de datos y análisis de desempeño.

El proyecto se centra en el diseño e implementación de un sistema integral que organiza la información disponible en archivos de monitoreo y la transforma en estructuras de datos coherentes con el dominio de la red. A partir de este proceso, se construye una representación que permite integrar inventario, topología, servicios y métricas de rendimiento históricas en un entorno de consulta, habilitando la exploración y el análisis interactivo de la infraestructura y su comportamiento. Como resultado, se dispone de una herramienta que facilita la consulta operativa y el análisis técnico, reduciendo la dependencia de conocimiento experto y mejorando la accesibilidad a la información.

Como proyección, el trabajo sienta bases para extender el procesamiento hacia entornos distribuidos y avanzar hacia capacidades de análisis de mayor complejidad. En particular, establece un precedente para futuras investigaciones y desarrollos en analítica y aprendizaje automático aplicados a redes ópticas, y habilita la incorporación de funcionalidades como la automatización de diagnósticos, así como la optimización y la gestión dinámica de rutas de servicio, apoyadas en los modelos de datos implementados.

Palabras clave: Redes Ópticas, T-API, Modelo de Datos, Pipeline de Datos, ETL, Visualización de Datos, ANTEL, Infinera, Parámetros de Performance.

Esta página ha sido intencionalmente dejada en blanco.

Glosario

ANTEL Administración Nacional de Telecomunicaciones.

API Application Programming Interface.

BER Bit Error Rate.

BPSK Binary Phase Shift Keying.

BSON Binary JSON.

BVT Bandwidth Variable Transponder.

CC Cross-Connection.

CDC Colorless, Directionless and Contentionless.

CEP Connection End Point.

CIM Common Information Model.

CPU Central Processing Unit.

CSV Comma-Separated Values.

DP Dual Polarization.

DSR Digital Signal Rate.

DWDM Dense Wavelength Division Multiplexing.

ETL Extract, Transform and Load.

FEC Forward Error Correction.

FSM FlexROADM Switching Module.

HTTP HyperText Transfer Protocol.

I/Q In-phase/Quadrature.

Glosario

IETF Internet Engineering Task Force.

INFN Infinera.

IPv4 Internet Protocol version 4.

IPv6 Internet Protocol version 6.

ITU-T International Telecommunication Union - Telecommunication Standardization Sector.

JSON JavaScript Object Notation.

KPI Key Performance Indicator.

LC-PT Link Connection Physical Trail.

LC-SP Link Connection Server Path.

LO Local Oscilator.

LPDDR Low-Power Double Data Rate.

NBI Northbound Interface.

NE Network Element.

NEP Node Edge Point.

NETCONF Network Configuration Protocol.

OADM Optical Add-Drop Multiplexer.

OAM Operations, Administration and Maintenance.

OCC Optical Channel Carrier.

OCH Optical Channel.

ODU Optical channel Data Unit.

ODUCnI Optical Digital Unit-CnI.

OLA Optical Line Amplifier.

OMNI Optical Mobile Network Integration.

OMS Optical Multiplex Section.

ONF Open Networking Foundation.

OPSW Operational Switch.

OPU Optical Payload Unit.

OSNR Optical Signal-to-Noise Ratio.

OTH Optical Transport Hierarchy.

OTN Optical Transport Network.

OTS Optical Transmission Section.

OTSi Optical Tributary Signal.

OTSiG Optical Tributary Signal Group.

OTU Optical Channel Transport Unit.

OTUCnI Optical Channel Transport Unit-Cn.

PFC Proyecto Final de Carrera.

PI Performance Indicator.

PIG Performance Indicator Group.

PM Performance Management.

QAM Quadrature Amplitude Modulation.

QGIS Quantum Geographic Information System.

QoT Quality of Transmission.

QPSK Quadrature Phase Shift Keying.

RAM Random-Access Memory.

REST Representational State Transfer.

RESTCONF RESTful Configuration Protocol.

ROADM Reconfigurable Optical Add-Drop Multiplexer.

SAN Storage Area Network.

SBI Southbound Interface.

SBVT Sliceable Bandwidth Variable Transponder.

SCH Super-Channel.

SDH Synchronous Digital Hierarchy.

SDN Software-Defined Networking.

SGBD Sistema de Gestión de Bases de Datos.

SIP Service Interface Point.

SONET Synchronous Optical Network.

SQL Structured Query Language.

SSD Solid-State Drive.

T-API Transport Application Programming Interface.

TMF TM Forum (TeleManagement Forum).

TRNBI Transcend REST Northbound Interface.

UML Unified Modeling Language.

WDM Wavelength Division Multiplexing.

WSS Wavelength Selective Switch.

XML eXtensible Markup Language.

YANG Yet Another Next Generation.

Tabla de contenidos

Tabla de contenidos

Capítulo 1

Introducción

Las redes de telecomunicaciones actuales enfrentan importantes limitaciones en adaptabilidad y programabilidad, lo que dificulta su capacidad para satisfacer las crecientes demandas de servicios como el 5G, que requiere baja latencia y alta disponibilidad. La resolución de problemas relacionados con fallas, rendimiento y seguridad depende en gran medida de la intervención manual de operadores experimentados, quienes ajustan la red basándose en los resultados de los sistemas de monitoreo. Sin embargo, al no contar con un mecanismo automático de retroalimentación entre los sistemas de monitoreo y los componentes de configuración, la red no dispone de medios para realizar ajustes proactivos ni para refinar sus decisiones a partir de eventos previos.

El presente proyecto desarrolla una contribución basada en la extracción, el modelado y la visualización de datos para mejorar el análisis, la gestión y la comprensión del comportamiento de redes ópticas, tomando como caso de estudio la Red de Transporte de ANTEL, operador estatal de telecomunicaciones del Uruguay, basada en tecnología Infinera.

En este capítulo se presenta el marco general del proyecto. En la Sección 1.1 se describen la motivación y el contexto, exponiendo la problemática que origina este trabajo y su relevancia. La Sección 1.2 define los objetivos generales y específicos, mientras que la Sección 1.3 delimita el alcance e introduce los supuestos y restricciones considerados. Luego, la Sección 1.4 resume los antecedentes técnicos y teóricos que contextualizan la propuesta. Por último, la Sección 1.5 presenta las contribuciones del proyecto y la Sección 1.6 describe la organización del resto del documento.

1.1. Motivación y Contexto

El proyecto se lleva a cabo en el marco del grupo de investigación OMNI, liderado por los docentes e investigadores Claudina Rattaro y Alberto Castro, tutores del presente proyecto. El grupo se especializa en la gestión convergente de redes óptico-móviles de próxima generación (5G/6G). A lo largo de los años, OMNI ha acompañado a estudiantes en diversos ámbitos vinculados con redes ópticas, móvi-

les y multi-dominio, contribuyendo al desarrollo de tesis en temas de arquitectura, algoritmia e infraestructura. OMNI mantiene una colaboración activa con el área de Red de Transporte de ANTEL, respaldada por un convenio entre el ente estatal y la Facultad de Ingeniería de la Universidad de la República.

En este contexto, el presente proyecto se orienta a fortalecer la operación de la Red de Transporte de ANTEL, acompañando el despliegue de una nueva plataforma óptica basada en tecnología Infinera. Esta plataforma convive con la infraestructura existente de la red Nokia y su nivel de utilización crece de forma sostenida, lo que refuerza la necesidad de contar con herramientas que mejoren la capacidad de análisis y de monitoreo.

A partir del intercambio con el operador, se identificó que la operación de la red se apoyaba en tres elementos. En primer lugar, un visualizador alimentado por un único archivo diario, del que se obtenía información de manera puntual y orientada a casos específicos, sin un modelo de datos ni almacenamiento persistente. En segundo lugar, documentación operativa y, por último, el conocimiento experto del equipo, concentrado en un núcleo reducido de especialistas. Esto último generaba dependencia de personas clave y un riesgo para la continuidad y escalabilidad de las operaciones.

Frente a este escenario, el proyecto, por medio del procesamiento sistemático de información de la red Infinera, su estructuración en modelos de datos y el desarrollo de una herramienta de visualización, busca facilitar el análisis y el monitoreo, aportar soporte a la toma de decisiones operativas y reducir la dependencia de conocimiento concentrado en un núcleo acotado de especialistas.

1.2. Objetivos del Proyecto

El proyecto tiene como objetivo general diseñar e implementar un sistema de extremo a extremo que, a partir de datos de monitoreo de una red óptica, formalice un modelo de datos capaz de representarla y lo persista en esquemas de almacenamiento adecuados. Para ello, se diseñará un pipeline de procesamiento que alimente y mantenga actualizado dicho modelo, y se desarrollará una herramienta de visualización interactiva que facilite la comprensión de la red y apoye la toma de decisiones. Para alcanzar este propósito, se definen una serie de objetivos específicos:

- Estudiar los fundamentos teóricos y técnicos de redes ópticas pertinentes al proyecto.
- Comprender el caso de uso y el contexto operativo del área de Transporte de ANTEL.
- Analizar los datos de monitoreo disponibles y definir un modelo de datos que represente la red y sus métricas de interés.
- Diseñar e implementar un proceso de extracción, validación, transformación y carga de datos, alineado con el modelo definido.

- Definir e implementar esquemas de almacenamiento persistente acordes a los usos previstos.
- Desarrollar una herramienta de visualización interactiva que integre inventarios, topologías y métricas de desempeño.

1.3. Alcance, Supuestos y Restricciones

El proyecto se desarrolla sobre la Red de Transporte de ANTEL, con foco en la infraestructura óptica basada en tecnología Infinera, y se orienta al procesamiento y análisis de datos operativos obtenidos de la red, vinculados con inventario, topología y métricas de desempeño. El alcance se estructura en tres etapas consecutivas. En primer lugar, se define un modelo de datos que represente la red a partir de las distintas fuentes de datos. En segundo lugar, se diseña e implementa un pipeline que extraiga, transforme y persista la información conforme a ese modelo. Por último, se desarrolla una herramienta de visualización interactiva para la exploración y análisis de esta infraestructura, implementada en función de los requerimientos de indicadores de rendimiento definidos por ANTEL.

Como trabajo paralelo, fuera del alcance de este proyecto, se desarrolla una migración a entornos distribuidos del ecosistema Apache (Spark/Hive), alineada con la infraestructura de datos utilizada por ANTEL. Esta línea, a cargo de Bruno Tió, integrante del grupo de investigación OMNI, busca habilitar la ejecución del sistema desarrollado en este proyecto sobre dicha infraestructura. Profundizar en esta dirección habilita escalar el procesamiento, manejar mayores volúmenes de datos e incorporar flujos cercanos al tiempo real.

Para la ejecución del proyecto se asume la disponibilidad de tres fuentes de datos operativas provistas por el área de Transporte de ANTEL. Estas incluyen archivos diarios de monitoreo de la red y de métricas de rendimiento, así como archivos con las coordenadas geográficas de los nodos que la conforman, cuya recepción no sigue una periodicidad fija. Se asume además el acceso a documentación técnica y material de referencia del fabricante, con el fin de comprender la infraestructura y su funcionamiento. Por último, se prevén instancias de intercambio con el equipo de Transporte de ANTEL para resolver consultas, validar supuestos y afianzar la comprensión del dominio.

En lo que respecta a las restricciones, el proyecto está condicionado por factores que impactan en su diseño e implementación. En primer lugar, por requerimiento de ANTEL, el sistema debe operar en modalidad offline, sin conexión a Internet, lo que inhabilita el uso de servicios externos y exige una mayor autonomía operativa. En segundo lugar, los hitos del cronograma acordado con ANTEL delimitan el alcance por fase y establecen plazos específicos de trabajo, lo que acota el nivel de profundización posible por etapa. En tercer lugar, la generación y disponibilidad de los archivos de monitoreo no está bajo control del proyecto, por lo que el procesamiento se limita a los datos provistos por ANTEL. Por último, en lo que respecta a la documentación técnica, si bien puede encontrarse material del fabricante en fuentes no oficiales, la documentación de referencia utilizada en el proyecto es pro-

vista por ANTEL, lo que condiciona el nivel de detalle del análisis en función de la información que el operador ponga a disposición.

1.4. Antecedentes

El proyecto toma como referencia investigaciones previas que abordan aspectos relevantes para la gestión y optimización de redes ópticas. En particular, se consideran el trabajo de fin de carrera de Ignacio Bianchi y Franco Donnangelo [17] y el trabajo de fin de carrera de Santiago Olmedo [15]. Ambos estudios desarrollados en colaboración con ANTEL.

En términos generales, el primer trabajo propuso una metodología para procesar y analizar datos de monitoreo de la Red de Transporte de ANTEL sobre infraestructura Nokia, incorporando técnicas de aprendizaje automático para estimar métricas de interés y una aplicación interactiva para visualizar resultados y apoyar la operación. El segundo trabajo definió un modelo de datos y un proceso de integración y transformación orientado a la visualización de información de red. Además, empleó datos de Fraunhofer HHI [14] para predecir la calidad de transmisión mediante algoritmos de aprendizaje automático, comparando enfoques y criterios de optimización.

Estos antecedentes aportan de forma significativa al desarrollo del presente proyecto, en particular en el modelado de datos como base del análisis, en la construcción de pipelines de procesamiento y en la presentación de resultados mediante una herramienta de visualización. Además, al demostrar la viabilidad de aplicar algoritmos de aprendizaje automático sobre datos de redes ópticas de transporte, establecen un marco de referencia para una posible evolución de este trabajo. Si bien la optimización de la red queda fuera del alcance actual, el modelo de datos y las vistas generadas se diseñan de forma compatible con los enfoques de estos estudios, de modo de facilitar a futuro la incorporación de algoritmos de optimización y análisis avanzado.

No obstante, la aplicación de estos enfoques requiere considerar diferencias relevantes. Los trabajos previos se basan en infraestructura Nokia, mientras que el presente proyecto aborda una plataforma óptica basada en tecnología Infinera, con modelos de datos, nomenclaturas y convenciones de identificación distintas. Aun así, el contraste entre ambas infraestructuras aportó criterios útiles para interpretar la información de la plataforma Infinera. En particular, el solapamiento geográfico de algunos nodos permitió depurar coordenadas y consolidar la topología física. Por otra parte, la revisión de la definición de servicios utilizada por ANTEL en la infraestructura Nokia contribuyó a normalizar y comprender la nomenclatura empleada.

Si bien estos antecedentes constituyeron un punto de partida relevante, la ausencia de antecedentes sobre la plataforma Infinera exigió complementar el marco de referencia con un análisis del dominio basado en documentación del fabricante, intercambios con ANTEL e inspección directa de los datos provistos.

1.5. Contribución

El presente proyecto aporta un sistema integrado para el aprovechamiento de datos operativos de la Red de Transporte de ANTEL basada en tecnología Infinera. El eje de la contribución es un modelo de datos unificado que abstrae la representación de la red y normaliza la información de monitoreo proveniente de distintas fuentes, organizándola de forma estructurada para facilitar su análisis y exploración. El modelo brinda una visión consistente del dominio y permite representar la topología física y de servicios, así como las métricas de desempeño asociadas. Sobre esta base se diseña e implementa un pipeline de extracción, transformación y carga que procesa los datos conforme al modelo y los almacenan en bases de datos que aseguran persistencia.

Sobre estos componentes se desarrolla una herramienta de visualización interactiva orientada a la operación, que expone inventarios, topologías y métricas de interés para el equipo operativo, apoyando la toma de decisiones y mejorando la comprensión del estado de la red. La Figura 1.1 sintetiza la arquitectura propuesta y el encadenamiento entre modelo, pipeline y visualización. Como complemento, se entrega documentación técnica que describe el funcionamiento de cada etapa, detalla el modelo y sus parámetros, y establece lineamientos de uso, validación y mantenimiento para favorecer la adopción, la continuidad del sistema y su proyección hacia trabajos futuros.

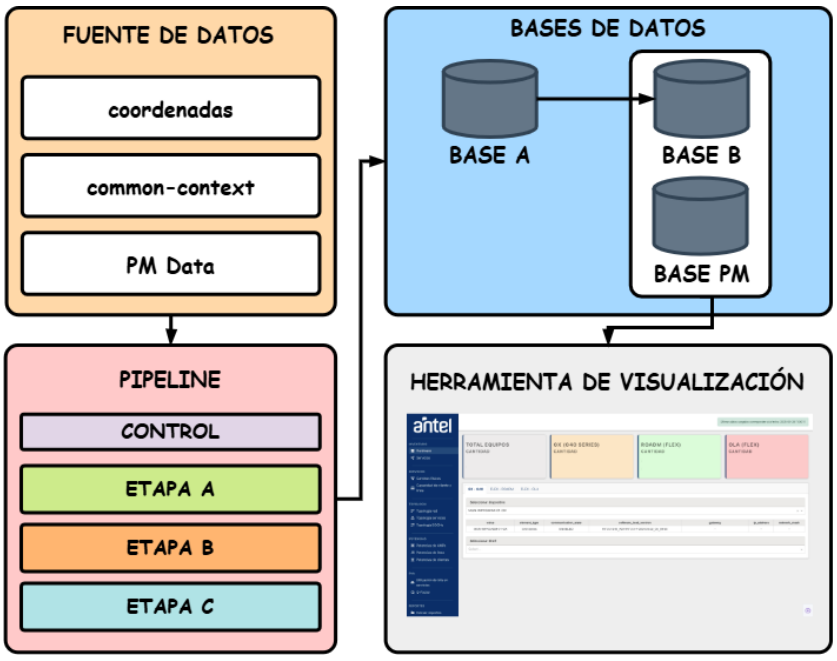

Figura 1.1: Diagrama de alto nivel de la arquitectura del sistema. Se ilustra el flujo de extracción, transformación y carga de los datos de monitoreo hacia los sistemas de almacenamiento y su posterior representación mediante una herramienta de visualización interactiva.

1.6. Estructura del Documento

Se resume a continuación el contenido de cada capítulo. El Capítulo 2 presenta el marco conceptual y los fundamentos teóricos del trabajo. El Capítulo 3 caracteriza el dominio y el contexto operativo del proyecto. El Capítulo 4 describe el diseño de los modelos de datos. El Capítulo 5 detalla la arquitectura e implementación del proceso de extracción, transformación y carga. El Capítulo 6 presenta la herramienta de visualización interactiva y sus funcionalidades principales. Finalmente, el Capítulo 7 resume los resultados y conclusiones y plantea líneas de trabajo futuro.

Capítulo 2

Marco Teórico

Este capítulo presenta los fundamentos teóricos que sustentan el proyecto y construye el contexto desde la red óptica de transporte hasta el uso de datos para su operación.

En primer lugar, se introduce el concepto de red óptica de transporte como el dominio que provee conectividad de alta capacidad entre nodos principales, regiones y centros de datos. En este marco se describen sus dos componentes complementarios: la infraestructura fotónica, donde la capacidad se organiza en el espectro y se transporta sobre fibra mediante sistemas Wavelength Division Multiplexing (WDM)/Dense Wavelength Division Multiplexing (DWDM), y la capa digital, que estructura los servicios para su transmisión extremo a extremo. Sobre esta base se presentan los elementos que materializan la conectividad en el dominio óptico (Transponders, Reconfigurable Optical Add-Drop Multiplexer (ROADM) y amplificadores ópticos), junto con criterios asociados al uso del espectro y a los esquemas de transmisión.

A partir de esta infraestructura, se incorpora OTN como arquitectura de transporte digital, que estandariza la encapsulación y multiplexación de señales cliente e incorpora funciones de supervisión y operación extremo a extremo sobre la red óptica. Con ello se completa una visión integrada del transporte, desde el canal óptico que viaja por la fibra hasta las estructuras digitales que permiten gestionar servicios de forma consistente.

Luego, se presentan las arquitecturas de operación que permiten administrar la red y sus servicios. Primero se introduce la separación funcional por planos (transporte, control y gestión) para ubicar responsabilidades operativas y comprender cómo se establece, mantiene y supervisa la conectividad. Sobre esa organización se introduce SDN como un enfoque que refuerza la automatización y la interoperabilidad al habilitar el acceso a capacidades de red mediante interfaces estandarizadas. En este contexto, se adopta T-API como interfaz NBI para exponer, de forma uniforme, información y servicios de transporte (por ejemplo, topología y conectividad) consumibles por aplicaciones.

Finalmente, el capítulo define el marco conceptual de datos utilizado en el proyecto. Dado que la operación de redes ópticas se apoya en información diversa (inventario, topología, configuración, estado, métricas y eventos), se establecen de-

finiciones y criterios sobre fuentes, formatos, modelos, almacenamiento y pipelines Extract, Transform and Load (ETL). Este marco sirve de base para fundamentar, en capítulos posteriores, las decisiones de extracción, integración y preparación de datos orientadas al análisis y la visualización.

La Sección 2.1 presenta el concepto de red óptica de transporte y sus componentes principales. La Sección 2.2 introduce la arquitectura OTN y su separación en capa digital y de medios ópticos. La Sección 2.3 describe la arquitectura funcional por planos y su interacción en la operación de la red. La Sección 2.4 presenta la arquitectura SDN y el rol de la interfaz NBI, adoptando T-API. La Sección 2.5 establece los fundamentos de datos utilizados en el proyecto. Por último, la Sección 2.6 resume los principales conceptos abordados.

2.1. Redes Ópticas de Transporte

La infraestructura de telecomunicaciones de un proveedor de servicios se organiza en dominios para gestionar la complejidad y escalar la capacidad de forma controlada. Si bien la nomenclatura varía según el operador o el estándar, es habitual distinguir red de acceso, red de agregación (metro o distribución) y red de transporte (core o backbone). La Figura 2.7 ilustra, a alto nivel, la interacción entre estos dominios.

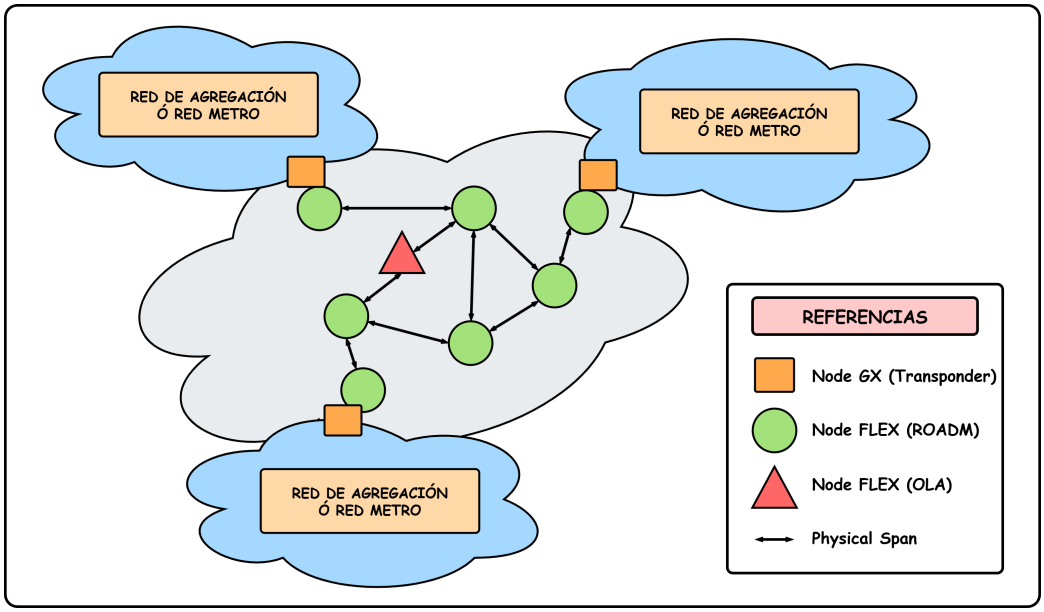

Figura 2.1: Diagrama de alto nivel que representa la interacción entre redes de agregación o metro a través de la red de transporte óptica. Estas pueden ser redes de tipo móvil, Ethernet o centros de datos. Figura basada en la Figura 2.8, sección 2.8 Metro/Core Network Architecture [25].

La red de acceso conecta a los usuarios finales con los primeros puntos de concentración y tiene un alcance predominantemente local. La red de agregación

concentra ese tráfico, lo encamina hacia nodos de mayor jerarquía e interconecta múltiples zonas, típicamente a escala urbana o regional. Finalmente, la red de transporte provee conectividad de alta capacidad entre nodos principales, regiones y centros de datos, priorizando disponibilidad, eficiencia y escalabilidad en un alcance regional o nacional.

El presente proyecto se centra en redes de transporte óptico. En este dominio, la arquitectura suele describirse mediante dos capas complementarias: la capa fotónica, responsable del transporte de señales ópticas sobre fibra mediante canales en longitudes de onda, y la capa digital, encargada de estructurar, encapsular y proteger los servicios para su transmisión extremo a extremo sobre la infraestructura óptica. En las secciones siguientes se presentan los fundamentos de ambas capas, los principales componentes de una red óptica de transporte y la organización funcional de la red en planos de datos, control y gestión, incluyendo el rol de arquitecturas SDN para habilitar automatización y programabilidad de estos sistemas.

2.1.1. Fundamentos del Medio Óptico

La capa fotónica constituye el nivel físico de una red óptica de transporte. Su función es transportar señales en forma de luz sobre enlaces de fibra y habilitar su encaminamiento a través de la red mediante el establecimiento de lightpaths y la asignación de recursos en el espectro disponible, manteniendo la calidad de transmisión requerida. En este contexto, un lightpath es una conexión óptica extremo a extremo entre dos nodos, establecida sobre la infraestructura de transmisión sin recurrir a conversiones al dominio eléctrico en los nodos intermedios.

Desde el punto de vista de infraestructura, esta capa se apoya en la fibra óptica como medio de transmisión y en equipos que generan, insertan y encaminan señales ópticas a lo largo de la red. A continuación se presenta la fibra óptica y se introduce la multiplexación por longitudes de onda (WDM/DWDM), tecnología implementada en las redes ópticas actuales y que permite transportar múltiples señales ópticas en una misma fibra asignando distintas longitudes de onda. Por último, se abordan criterios de planificación y asignación del espectro, que determinan cómo se organiza y aprovecha la capacidad disponible.

Fibra Óptica

La fibra óptica es el medio físico que sustenta las comunicaciones modernas, ya que permite transportar grandes volúmenes de información a largas distancias con bajas pérdidas y alta inmunidad a interferencias electromagnéticas, superando ampliamente las prestaciones del cableado de cobre. Consiste en un hilo delgado de material transparente, típicamente vidrio (aunque también puede ser plástico), por el que la información se transmite en forma de luz [37].

Su estructura incluye un núcleo por donde se propaga la señal, un revestimiento con índice de refracción ligeramente menor que mantiene la luz confinada por reflexión interna total, y una cubierta protectora externa. En comunicaciones ópticas

se utilizan principalmente longitudes de onda entre 1300 nm y 1600 nm, organizadas en bandas espectrales (ver Figura 2.2). Entre ellas, la banda C es la más empleada en enlaces de transporte por sus bajas pérdidas y por la disponibilidad de amplificación óptica eficiente, debido a que existen tecnologías de amplificación cuya ganancia se encuentra naturalmente centrada en esta región del espectro. Opera aproximadamente entre 1530 nm y 1565 nm (equivalente a 191,5 THz a 195,9 THz), habilitando transmisiones de larga distancia.

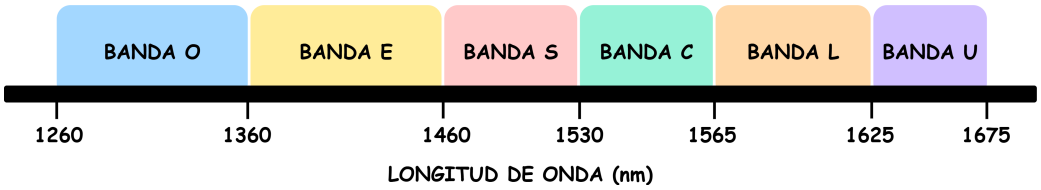

Figura 2.2: Esquema de las bandas espectrales utilizadas en comunicaciones por fibra óptica. Figura basada en la Figura 1.3, sección 1.2 Optical Spectral Bands [24].

Multiplexación por División de Longitud de Onda (WDM y DWDM)

La multiplexación por división de longitud de onda (WDM) es una tecnología que permite transmitir en simultáneo múltiples señales ópticas sobre una misma fibra, asignando a cada una una longitud de onda distinta. Cada longitud de onda constituye un canal independiente capaz de transportar diferentes demandas de cliente.

WDM surge para aprovechar mejor el ancho espectral de la fibra. En los primeros sistemas se utilizaba una única longitud de onda y, a medida que la demanda de tráfico creció, resultó más eficiente ampliar capacidad agregando canales ópticos en paralelo que escalar únicamente la tasa por canal.

En términos funcionales, un sistema WDM opera en tres etapas: los transmisores generan los canales y un multiplexor los combina en una señal compuesta, la señal se propaga por la fibra con múltiples longitudes de onda coexistiendo en el mismo enlace, y un demultiplexor separa los canales en el extremo receptor para su detección y recuperación.

Con la madurez de los componentes ópticos y la estandarización de la grilla de frecuencias (típicamente 100 GHz o 50 GHz), fue posible reducir el espaciamiento entre canales y transportar una mayor cantidad de longitudes de onda. Este enfoque se conoce como DWDM, y permite incrementar la capacidad al aumentar el número de canales por fibra (ver Figura 2.3). Al compartir la misma infraestructura (fibra, amplificación y equipos de multiplexación), se reduce el costo por canal y se habilita una expansión incremental incorporando nuevos canales en los extremos del enlace.

En redes conmutadas, la operación WDM/DWDM se apoya en equipos que encaminan canales ópticos a nivel de longitud de onda. Estos nodos determinan qué canales se agregan (add), cuáles se extraen (drop) y cuáles atraviesan el equipo en bypass sin conversión O-E-O. Los Optical Add-Drop Multiplexer (OADM) implementan estas funciones con selección esencialmente fija, mientras que los ROADM

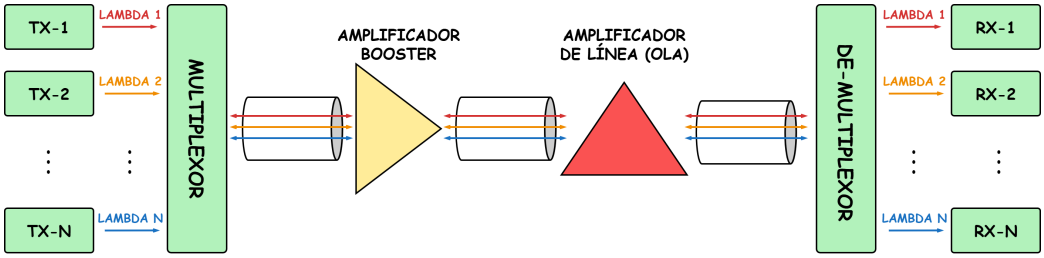

Figura 2.3: Esquema de un enlace óptico WDM con múltiples transmisores Lambda de 1 a N, donde las señales son multiplexadas, amplificadas mediante un booster y un amplificador de línea, y posteriormente demultiplexadas para obtener las Lambdas de salida de 1 a N. Figura basada en Figura 2.2, sección 2.3 Point-to-Point Fixed-Grid DWDM Architecture [25].

incorporan reconfiguración remota y dinámica, lo que habilita una operación más flexible y escalable en redes de transporte (ver Figura 2.4). Este último componente se describe en detalle más adelante.

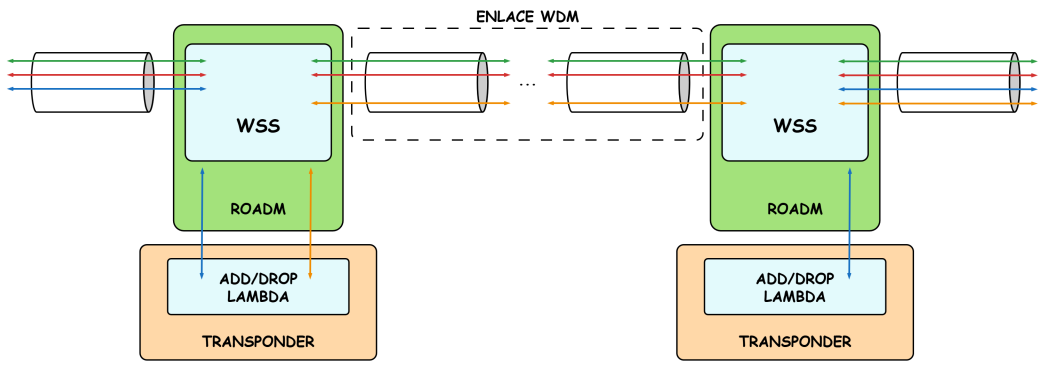

Figura 2.4: Esquema de una conexión óptica WDM entre dos ROADMs, donde los Transponders agregan y extraen Lambdas. Cada ROADM incorpora un bloque Wavelength Selective Switch (WSS) encargado de dirigir las Lambdas a través del enlace óptico. Figura basada en Figura 2.3, sección 2.3 Point-to-Point Fixed-Grid DWDM Architecture [25].

Planificación y Asignación del Espectro

La planificación y asignación del espectro en redes DWDM evolucionó desde un esquema rígido hacia esquemas flexibles y elásticos. Esta transición estuvo impulsada por el crecimiento del tráfico, la necesidad de aumentar capacidad sin tender nueva fibra y la coexistencia de señales con diferentes requerimientos espectrales, que volvió ineficiente y limitante el uso de canales estrictamente fijos.

La International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) definió en la recomendación G.694.1 [20] una grilla espectral de referencia para aplicaciones DWDM, con el propósito de ordenar el uso del espectro y asegurar la interoperabilidad entre equipos de distintos fabricantes. En este marco se establece un conjunto de frecuencias centrales nominales permitidas y se contemplan espaciamientos que van desde 100 GHz hasta 12,5 GHz e incluso

granularidades menores, obtenidas por subdivisión sucesiva por factores de 2. Sobre esta base se consideran tanto esquemas de grilla fija (con espaciamientos nominales típicos como 100 GHz y 50 GHz) como esquemas flexibles, donde la asignación se expresa en ranuras espectrales caracterizadas por una frecuencia central nominal y un ancho que puede definirse como múltiplo de una granularidad menor.

Fixed-Grid Inicialmente, los sistemas DWDM organizaron su espectro en canales de ancho fijo (fixed-grid) con espaciamientos de 100 GHz y 50 GHz. Este esquema simplifica la multiplexación y la conmutación, pero impone algunas restricciones. Si una señal requiere menos ancho de banda que el canal asignado, se desperdicia capacidad espectral, ademas, si requiere más, como puede ser el caso de señales con tasas de transferencia mas altas (400 Gb/s o superiores), puede no ajustarse adecuadamente a la ranura disponible o sufrir penalizaciones por filtrado en nodos intermedios. En particular, para una grilla de 100 GHz, la recomendación ITU-T G.694.1 [20] define las frecuencias centrales nominales permitidas en la banda C como

$$f = (193{,}1 + n \cdot 0{,}1)\,, \text{THz}, \tag{2.1}$$

donde 193,1 THz es la frecuencia de referencia y n es un entero que indica la posición en la grilla. El razonamiento es análogo para otras grillas derivadas por subdivisión, sustituyendo 0,1 por el espaciamiento correspondiente.

Flex-Grid Para mitigar estas limitaciones se adopta un enfoque de grilla flexible (flex-grid), que mantiene el marco de referencia de la grilla definido por ITU-T pero reemplaza el canal fijo por el concepto de ranura de frecuencia [20], el cual consiste en un rango contiguo definido por una frecuencia central nominal y un ancho ajustable en múltiplos de una granularidad definida. Su implementación fue posible a partir de la madurez de tecnologías ópticas que permiten generar, filtrar y conmutar señales con resolución más fina, de modo de operar con asignaciones espectrales variables sin perder selectividad ni estabilidad en la transmisión. En particular, el estándar G.694.1 [20] define para la grilla flexible se define la frecuencia central nominal con granularidad de 6,25 GHz como

$$f_c = (193{,}1 + n \cdot 0{,}00625)\,, THz, \tag{2.2}$$

donde n es un entero (positivo, negativo o cero). Además, el ancho de ranura se expresa con granularidad de 12,5 GHz como

$$\Delta f = 12{,}5 \cdot m, GHz, \tag{2.3}$$

donde m es un entero positivo. A partir de esta definición, el ancho asignado puede adaptarse al requerimiento espectral de cada señal, en función de la tasa de transmisión, el formato de modulación y la distancia, evitando desperdicio en demandas de menor ocupación y habilitando señales de mayor ancho sin quedar restringidas por un espaciamiento fijo (ver Figura 2.5). En la práctica, esto permite asignaciones más compactas y la conformación de superchannels, entendidos como un conjunto de subportadoras ópticas contiguas que se gestionan como una única conexión lógica para soportar tasas elevadas sobre espectro continuo.

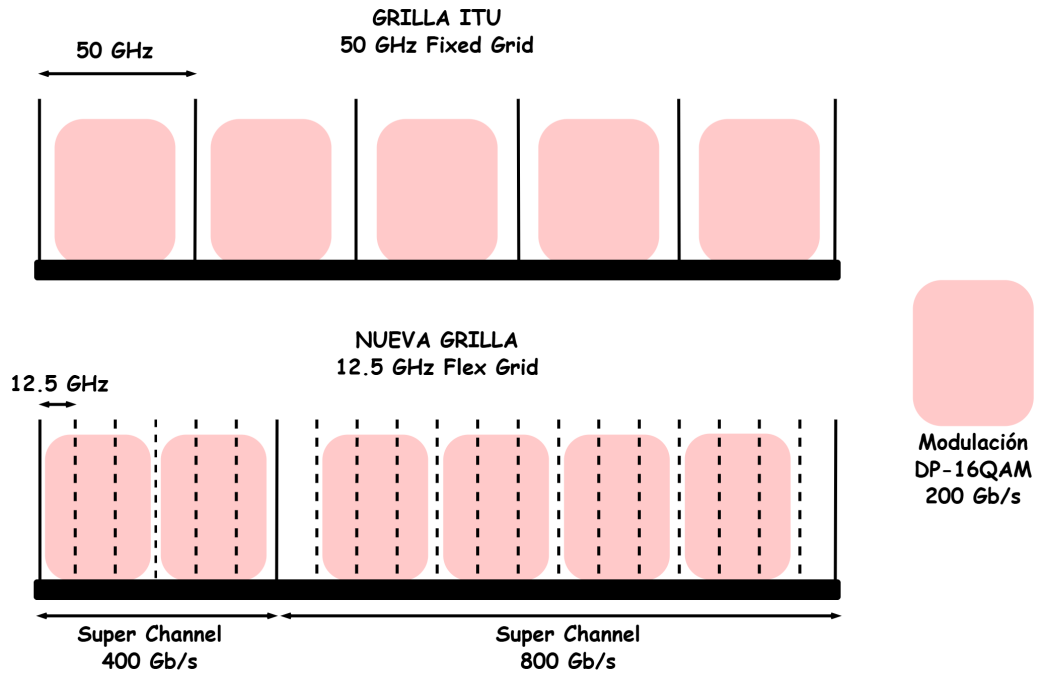

Figura 2.5: Comparación entre una red de rejilla fija de $50\ GHz$ y una red Flex-Grid de $12{,}5\ GHz$, en la que se conforman dos superchannels de $400\ Gb/s$ y $1\ Tb/s$ para transmitir una señal de $200\ Gb/s$ modulada con la técnica Dual Polarization (DP)-16Quadrature Amplitude Modulation (QAM). Figura basada en la Figura 2.1, sección 2.2 The History of ITU Grids and Development from Fixed to Flex-Grid [25].

Redes Ópticas Elásticas Sobre la base del flex-grid, las redes ópticas elásticas (EON) amplían la flexibilidad al incorporar adaptación no solo en la asignación espectral de la red, sino también en la forma en que cada conexión se genera en los extremos. Así, además de ajustar el ancho de la ranura espectral asignada, cada conexión puede variar su tasa y su ocupación espectral según la demanda del servicio y las condiciones del enlace. Esto se logra combinando nodos con flex-grid con Transponders avanzados en los extremos, encargados de convertir la señal cliente al dominio óptico y realizar la operación inversa en recepción. Mientras que en sistemas DWDM tradicionales estos equipos operaban típicamente con configuraciones fijas, en EON se emplean Transponders de ancho de banda variable (BVT), capaces de ajustar parámetros de transmisión para adaptar el compromiso alcance–eficiencia espectral de cada conexión.

2.1.2. Componentes Principales de la Infraestructura Óptica

Una red óptica de transporte se construye a partir de elementos con funciones bien definidos. Se consideran tres componentes principales: el Transponder, encargado de la transmisión y recepción de las señales ópticas que transportan el tráfico del cliente, el ROADM, encargado de conmutar y enrutar longitudes de onda en el dominio óptico, y los amplificadores ópticos de línea (Optical Line Amplifier

(OLA)), cuya funcion es compensar las pérdidas de transmisión a lo largo de los enlaces de fibra. Una posible configuración se presenta en la Figura 2.6.

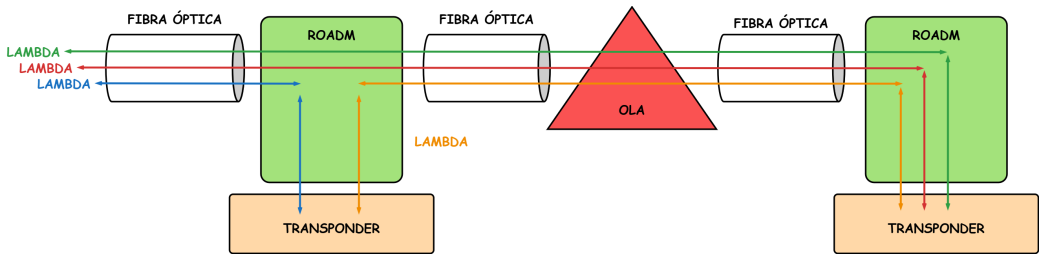

Figura 2.6: Relación entre los principales elementos de una red óptica en una configuración compuesta por Transponder-ROADM-OLA-ROADM-Transponder, donde se visualizan los Transponder agregando y quitando las longitudes de onda (Lambdas) establecidas entre los extremos.

A continuación se describen estos componentes con mayor detalle: se presenta el Transponder y sus variantes Bandwidth Variable Transponder (BVT) y Sliceable Bandwidth Variable Transponder (SBVT), que permiten adaptar formato de modulación, tasa y ocupación espectral según los requerimientos del servicio; los ROADM y sus bloques funcionales (Colorless, Directionless and Contentionless (CDC), Add/Drop, Express y WSS); los amplificadores de línea óptica (OLA).

Transponder

El Transponder es el equipo ubicado en los extremos de un servicio. Está encargado de adaptar la señal del cliente y realizar la conversión entre el dominio eléctrico y el óptico. En el extremo emisor recibe las señales eléctricas del cliente y las transforma en una señal óptica modulada para su transmisión sobre la fibra. En el extremo receptor, otro Transponder detecta la señal óptica recibida y la convierte nuevamente a señal eléctrica para entregarla al cliente.

En su forma estándar, el Transponder opera con parámetros de transmisión predefinidos, por lo que su capacidad y su ocupación espectral quedan esencialmente fijadas por diseño. En particular, suele trabajar con una tasa de símbolos constante, la señal se ajusta a un ancho espectral fijo asociado a grillas WDM tradicionales y el formato de modulación no se modifica. Esto limita la posibilidad de ajustar, en forma significativa, la capacidad ofrecida o el espectro ocupado según la demanda o las condiciones del enlace.

En redes modernas, donde la demanda varía y el espectro es un recurso crítico, resulta ventajoso adaptar estos parámetros para seleccionar el compromiso capacidad–alcance más conveniente. Bajo este objetivo surgen los Transponders Bandwidth-Variable (BVT), que permiten reconfigurar parámetros de transmisión para ajustar capacidad, alcance y ocupación espectral. Sobre esta base, los Sliceable BVT (SBVT) incorporan la posibilidad de particionar capacidad en múltiples slices asignables a distintos servicios. A continuación se formalizan ambos conceptos.

Bandwidth-Variable Transponder (BVT) Un BVT es un Transponder capaz de ajustar, en operación, la tasa y el ancho espectral ocupado mediante la selec-

ción del formato de modulación, la tasa de símbolos y la tasa de código de Forward Error Correction (FEC). Esta flexibilidad permite operar distintos compromisos entre la capacidad y el alcance, eligiendo configuraciones más robustas para distancias mayores o configuraciones más eficientes cuando el enlace ofrece mayor margen [25]. Además, los BVT pueden generar superchannels, es decir, conjuntos de subportadoras contiguas en frecuencia que se tratan como una única entidad lógica para asignación y operación. En este caso, el formato de modulación condiciona la eficiencia espectral alcanzable, mientras que el número de subportadoras y su tasa de símbolos determinan la ocupación espectral total y la tasa de datos resultante. A nivel de red, estas capacidades se aprovechan plenamente cuando el dominio de conmutación óptica, por ejemplo un ROADM, permite establecer conexiones con ancho espectral variable, habilitando asignación de espectro acorde a los requerimientos del servicio.

Sliceable Bandwidth-Variable Transponder (SBVT) Un SBVT extiende el concepto de BVT al permitir particionar la capacidad del transmisor en múltiples slices que pueden operar como flujos relativamente independientes. En la práctica, esto se materializa generando múltiples portadoras o subportadoras dentro de un mismo equipo y asignándolas de forma independiente a distintos servicios, facilitando una asignación más flexible de capacidad. En escenarios donde la arquitectura de red lo soporta, esta partición también permite distribuir capacidad hacia diferentes destinos. En combinación con una red elástica, el SBVT habilita ajustar tanto el tamaño de la conexión (espectro asignado) como la cantidad de slices en función de la demanda.

ROADM

Un ROADM permite agregar, extraer y encaminar longitudes de onda de forma dinámica, sin convertir la señal al dominio eléctrico. En comparación con un OADM, donde las longitudes de onda a agregar/extraer suelen estar fijas o requieren intervención manual para modificarse, un ROADM habilita reconfiguración remota y automatizada. Esta capacidad permite operar redes más flexibles y escalables, reduciendo cambios físicos ante variaciones de demanda o topología. En un ROADM, los bloques funcionales típicos son el subsistema Add/Drop con capacidades CDC, el camino Express y el bloque WSS (ver Figura 3.4).

Bloque CDC El bloque CDC describe un conjunto de propiedades que aumentan la flexibilidad del subsistema Add/Drop:

- **Colorless:** cualquier longitud de onda puede agregarse o retirarse en un puerto Add/Drop sin quedar asociada a un "color" preasignado.
- **Directionless:** una longitud de onda agregada puede encaminarse hacia cualquiera de las direcciones (grados) del ROADM sin estar atada a una salida fija. La selección se realiza por software.

- **Contentionless:** evita bloqueos internos durante la operación de Add/Drop, habilitando manejar en simultáneo múltiples canales, incluyendo el uso de una misma longitud de onda en puertos distintos, sin competir por los mismos recursos internos.

Bloque Add/Drop (A/D) Este bloque cumple dos funciones: inyectar en la fibra de línea del ROADM una longitud de onda generada por el Transponder (Add) y extraer de la fibra una longitud de onda entrante para entregarla al Transponder (Drop) (ver Figura 2.7).

Bloque Express El bloque Express proporciona el camino de paso directo (bypass) para longitudes de onda que atraviesan el ROADM sin ser agregadas ni extraídas, permitiendo que continúen hacia otro ROADM sin conversión a eléctrica.

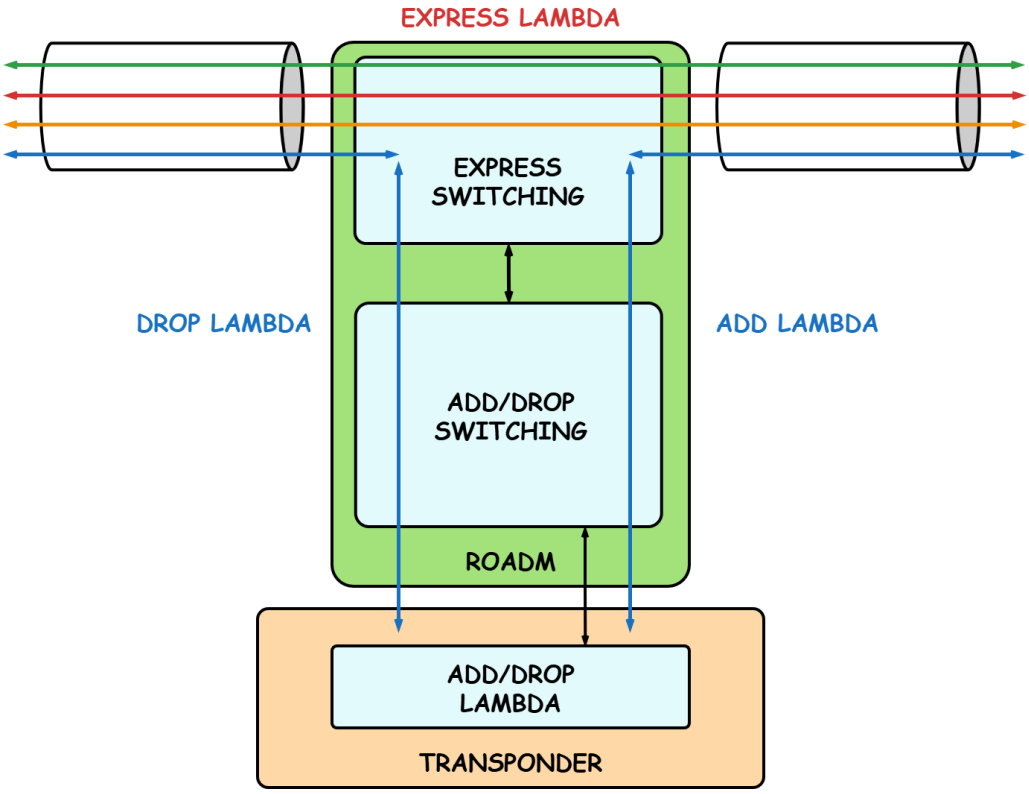

Figura 2.7: Interacción entre un Transponder y un ROADM: Add/Drop de una longitud de onda (azul) y paso por Express de longitudes de onda (verde, rojo y amarillo). Figura basada en la Figura 2.5, sección ROADM Architecture, de [25].

Bloque WSS El WSS permite dirigir selectivamente longitudes de onda hacia distintos puertos de salida del equipo, sin conversión al dominio eléctrico (ver

Figura 2.8). Opera como un conmutador/filtro óptico programable: recibe un conjunto de canales por un puerto común, selecciona cada longitud de onda y la envía al puerto correspondiente. En un ROADM, esto habilita encaminar canales hacia Add/Drop o hacia el camino Express según la configuración.

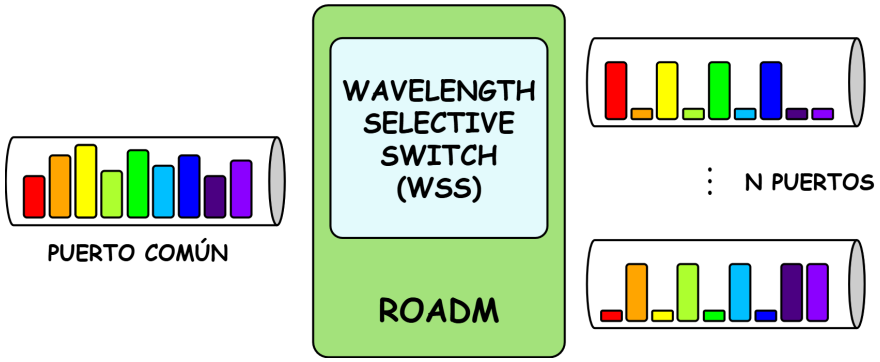

Figura 2.8: Funcionamiento conceptual de un WSS: las longitudes de onda que llegan por un puerto común se filtran y se dirigen hacia N puertos de salida según su destino. Figura basada en [13].

Amplificadores

Los amplificadores ópticos se utilizan para compensar la atenuación que sufre la señal al propagarse por la fibra y al atravesar elementos de la red. Su objetivo operativo es mantener niveles de potencia adecuados a lo largo del enlace y preservar la calidad de transmisión dentro de márgenes aceptables. En redes WDM, el amplificador actúa sobre un conjunto de longitudes de onda en simultáneo, por este motivo su dimensionamiento no depende solo de la distancia, sino también de la pérdida del tramo (span) y de la potencia objetivo por canal. En este trabajo se denomina OLA al amplificador instalado en línea, ubicado entre nodos para compensar la pérdida del tramo de fibra.

Además de elevar potencia, el comportamiento del amplificador se describe por dos aspectos operativos. Por un lado, su ganancia determina cuánto compensa la pérdida del tramo y se ajusta para mantener la potencia de salida dentro del rango objetivo de la red. Por otro lado, el amplificador introduce ruido (principalmente ASE), lo que degrada la relación señal-ruido óptica (Optical Signal-to-Noise Ratio (OSNR)) a medida que se concatenan tramos. En consecuencia, un amplificador no "recupera" la señal: la refuerza, pero también amplifica el ruido presente e incorpora ruido adicional, por lo que la calidad de la transmisión (Quality of Transmission (QoT)) del servicio queda limitada por el ruido acumulado y el presupuesto óptico del enlace.

2.1.3. Transmisión Óptica

La transmisión es un concepto fundamental en redes ópticas de transporte. Consiste en llevar información a través de fibra óptica mediante una portadora

modulada, garantizando que el receptor pueda recuperar los datos con la calidad requerida. A diferencia de otras tecnologías de red, el desempeño no depende únicamente de la conectividad lógica, si no que la señal se ve además afectada por fenómenos físicos durante su propagación (pérdidas, ruido, dispersión, filtrado, entre otros). Por ello, la transmisión se diseña y opera en términos de calidad de transmisión (QoT) y de márgenes asociados.

De forma general, el proceso comprende tres etapas. En el transmisor, un Transponder adapta y modula la información del cliente sobre una portadora óptica. Luego, la señal se propaga por la infraestructura (fibra, amplificadores y ROADM); finalmente, en el receptor se detecta la señal óptica y se recupera la información, aplicando los mecanismos de compensación y corrección que correspondan. En este marco, el esquema de transmisión se diferencia, en esencia, por el tipo de modulación empleada y por qué magnitudes de la señal óptica se aprovechan en la recepción. En redes de transporte se distinguen dos enfoques principales: transmisión con detección directa y transmisión coherente.

Transmisión Directa En la transmisión con detección directa, la información se transporta modulando la intensidad de la portadora óptica. Tras propagarse por la fibra, el receptor convierte directamente la potencia óptica en una señal eléctrica mediante un fotodiodo, cuya corriente se amplifica y se acondiciona mediante filtrado antes de la decisión binaria. En forma simplificada, la amplitud resultante se compara con un umbral para decidir el valor de bit [23]. Este enfoque es conceptualmente simple y, en el marco de redes WDM, suele desplegarse sobre canales de ancho fijo definidos por una grilla espectral fija (fixed-grid).

Transmisión Coherente En la transmisión coherente, la información se codifica en amplitud y fase de la portadora óptica y, típicamente, se aprovechan dos polarizaciones ortogonales de la luz. Esto implica que, dentro del mismo canal óptico, se transmiten dos flujos independientes utilizando estados de polarización mutuamente ortogonales (por ejemplo, dos orientaciones perpendiculares), que el receptor puede separar y procesar por separado. En el transmisor, un modulador In-phase/Quadrature (I/Q) genera la señal compleja. En el receptor, un oscilador local (Local Oscilator (LO)) permite recuperar las componentes en cuadratura y, mediante procesamiento digital, compensar degradaciones y decidir los símbolos transmitidos [3, 23]. Este enfoque habilita formatos de modulación de mayor eficiencia (por ejemplo, Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK) y familias QAM) y el uso de polarización dual (DP), lo que permite transportar dos señales en paralelo dentro del mismo canal óptico y, por tanto, aumentar la tasa de transmisión sin incrementar el ancho de banda ocupado (ver Figura 2.9).

```
bitrate  =  simbolos/seg × bits/símbolo × N_pol,
```

Figura 2.9: Cálculo del bitrate como producto de la tasa de símbolos, los bits por símbolo y el número de polarizaciones ($N_{\text{pol}} \in \{1, 2\}$).

La elección del formato de modulación conlleva un compromiso alcance-capacidad. Órdenes de modulación superiores aumentan los bits por símbolo y la capacidad, pero exigen mayor OSNR y reducen el alcance. En cambio, formatos más robustos (por ejemplo, QPSK o BPSK) toleran OSNR más bajas y favorecen distancias mayores a costa de eficiencia espectral [3]. En consecuencia, la transmisión coherente se alinea con redes flexibles/elásticas, donde la ocupación espectral y la capacidad pueden ajustarse en función del servicio y de las condiciones del enlace.

2.2. Arquitectura OTN

Hasta este punto se introdujeron varios conceptos necesarios para comprender las redes ópticas de transporte. Sin embargo, para operar servicios de extremo a extremo se requiere un marco que describa la red en términos de capas funcionales y cómo se combinan para entregar un servicio completo. En este contexto se introduce la arquitectura OTN, que brinda una descripción funcional independiente de la implementación física. Su aporte es definir una estructura común para transportar y supervisar señales cliente de distintos formatos y tasas de manera interoperable y consistente sobre la infraestructura subyacente.

La Recomendación ITU-T G.872 define la arquitectura funcional de la Red de Transporte Óptico (OTN) como un modelo lógico para describir, a nivel de red, cómo se proveen capacidades de transporte extremo a extremo, agregación/multiplexación, encaminamiento/conmutación, supervisión y supervivencia de señales de cliente sobre infraestructura óptica. El valor de este enfoque es separar el qué (capas y funciones de transporte) del cómo (implementación física), permitiendo representar la red de forma consistente en escenarios multi-servicio y multi-vendor, donde distintos tipos de señales cliente deben transportarse y supervisarse de manera interoperable.

Desde el punto de vista estructural, G.872 organiza la OTN en dos componentes funcionales complementarios: la capa digital y la capa de medios ópticos. La capa digital describe cómo se encapsula la información del cliente en unidades estandarizadas y cómo se agrega/multiplexa para su transporte, incorporando además funciones de operación, administración y mantenimiento (Operations, Administration and Maintenance (OAM)) que permiten supervisar el trayecto extremo a extremo. La capa de medios ópticos describe el soporte sobre el cual viajan esas señales digitales: la conectividad óptica disponible en la infraestructura (es decir, los recursos que permiten establecer un canal óptico entre puntos) y los mecanismos de supervisión asociados al propio medio. Esta separación es funcional y no implica dos redes distintas. En la práctica, ambos componentes se complementan. La capa digital define qué señales se transportan y cómo se estructuran y supervisan, mientras que la capa de medios ópticos provee dónde y cómo se materializa su transporte sobre la infraestructura.

En las Subsecciones siguientes se profundiza en cada uno de estos componentes, describiendo las capas y entidades funcionales que los conforman, con el objetivo de comprender cómo se estructura la arquitectura OTN y cómo se materializa el transporte sobre la infraestructura. La Subsección 2.2.1 presenta la componente

digital, mientras que la Subsección 2.2.2 aborda la componente de medios ópticos.

2.2.1. Capa Digital

La Recomendación ITU-T G.872 define, a nivel arquitectónico, las capas digitales de la OTN y su relación cliente/servidor. En este contexto, una capa cliente es la información que se desea transportar, mientras que una capa servidor es la que le provee la capacidad de transporte e incorpora funciones de supervisión asociadas. En complemento, ITU-T G.709 especifica cómo se materializan estas capas mediante estructuras de señal estandarizadas, sus encabezados (overhead) y los mecanismos de mapeo/adaptación de señales cliente. [21, 22]

Dentro del dominio digital, las estructuras fundamentales son Optical Payload Unit (OPU), Optical channel Data Unit (ODU) y Optical Channel Transport Unit (OTU). La OPU contiene la información del cliente una vez adaptada a OTN y provee un contenedor común para transportar señales cliente de distinto tipo y tasa, incorporando la información necesaria para dicha adaptación.

La ODU representa el camino digital por el cual se transporta la información del cliente a través de la OTN. A esta capa se asocian funciones de mantenimiento orientadas al transporte extremo a extremo. Desde el punto de vista de la señal, G.709 describe a la ODU como una estructura compuesta por la OPU como carga útil (donde viaja el cliente adaptado) y el overhead de ODU, con campos de operación, administración y mantenimiento asociados al camino.

La OTU, por su parte, representa la sección digital que transporta una ODU como cliente, aportando supervisión a nivel de tramo y robustez de transmisión. En términos de señal, G.709 define la OTU como el portador de una ODU, incorporando overhead de OTU (orientado al mantenimiento de tramo) y, cuando corresponde, FEC para incrementar la tolerancia a errores en transmisión.

Estas estructuras existen en distintos órdenes (OPUk, ODUk, OTUk), asociados a capacidades nominales, lo que permite transportar y multiplexar servicios de diferentes tasas dentro de un marco estandarizado.

2.2.2. Capa de Medios Ópticos

La capa de medios ópticos describe cómo se materializa el transporte sobre la infraestructura fotónica: qué conectividad existe en el medio y qué mecanismos permiten supervisar su estado. En términos cliente/servidor, este dominio actúa como servidor de las capas digitales, aportando conectividad óptica para transportar señales digitales a través de la red.

Para describir esta conectividad sin depender de un vendor o de una implementación puntual, G.872 utiliza un conjunto de abstracciones funcionales encadenadas. Los puertos de medios identifican los puntos de origen y terminación entre los que se desea establecer conectividad óptica. Sobre esos extremos, los canales de medios representan el recurso del medio que habilita la conexión, típicamente asociado a una porción del espectro (por ejemplo, una longitud de onda). Los elementos de medios modelan las funciones intermedias del dominio óptico que permiten sostener

esa conectividad a lo largo de la infraestructura (por ejemplo, conmutación/filtrado y amplificación), sin describir su implementación interna. Finalmente, las funciones de mantenimiento óptico representan la supervisión del propio transporte óptico (estado, degradación y alarmas) asociada al recurso y a sus secciones dentro del dominio.

Sobre esta conectividad modelada, la arquitectura distingue tres capas ópticas principales, con alcances distintos: Optical Channel (OCH), Optical Multiplex Section (OMS) y Optical Transmission Section (OTS). La capa OCH representa un canal óptico individual asociado a un servicio. Operativamente se interpreta como la conectividad óptica a nivel de canal que transporta una señal óptica específica (por ejemplo, la portadora que soporta una señal digital). La capa OMS representa el transporte conjunto de múltiples canales OCH sobre una misma infraestructura, es decir, la sección multiplexada (por ejemplo, un conjunto WDM), y permite describir y supervisar el comportamiento del transporte considerando el agregado de canales. La capa OTS representa el tramo de transmisión más cercano a lo físico, típicamente asociado a la fibra entre puntos de terminación/gestión, donde se observan los efectos del medio y del tramo como tal.

En términos de uso, el modelo parte de puertos de origen y destino, asocia un canal de medios como recurso espectral para establecer la conectividad entre ellos, considera los elementos de medios intermedios que la hacen posible en la infraestructura y vincula funciones de mantenimiento para supervisar el estado del transporte. Desde la perspectiva cliente/servidor, una señal digital (por ejemplo, una OTU como cliente) requiere un OCH como soporte óptico, múltiples OCH se agrupan en una OMS y todo ello se transmite sobre tramos OTS de fibra.

2.3. Arquitectura Funcional por Planos

En redes ópticas de transporte, la provisión y operación de servicios extremo a extremo requiere coordinar tres tipos de funciones: transportar señales sobre la infraestructura, establecer y mantener conexiones, y operar y administrar la red (configuración, fallas y desempeño). Para ordenar esta complejidad se adopta una arquitectura funcional basada en la separación en tres planos: plano de transporte, responsable de la transferencia de la información, plano de control, responsable del establecimiento y mantenimiento de conexiones y plano de gestión, responsable de la administración del ciclo de vida de la red y sus servicios.

Esta separación aporta ventajas directas en la operación. En primer lugar, introduce modularidad al delimitar responsabilidades y reducir acoplamientos entre funciones. En segundo lugar, facilita la evolución e interoperabilidad, ya que cada plano puede ampliarse o modificarse con menor impacto sobre los demás. Por último, simplifica el diagnóstico y la operación al permitir ubicar fallas, métricas y decisiones en el plano correspondiente. En las siguientes Subsecciones se presenta el rol de cada plano y su interacción dentro de una red óptica de transporte. La Subsección 2.3.1 describe el plano de transporte, la Subsección 2.3.2 el plano de control y la Subsección 2.3.3 el plano de gestión.

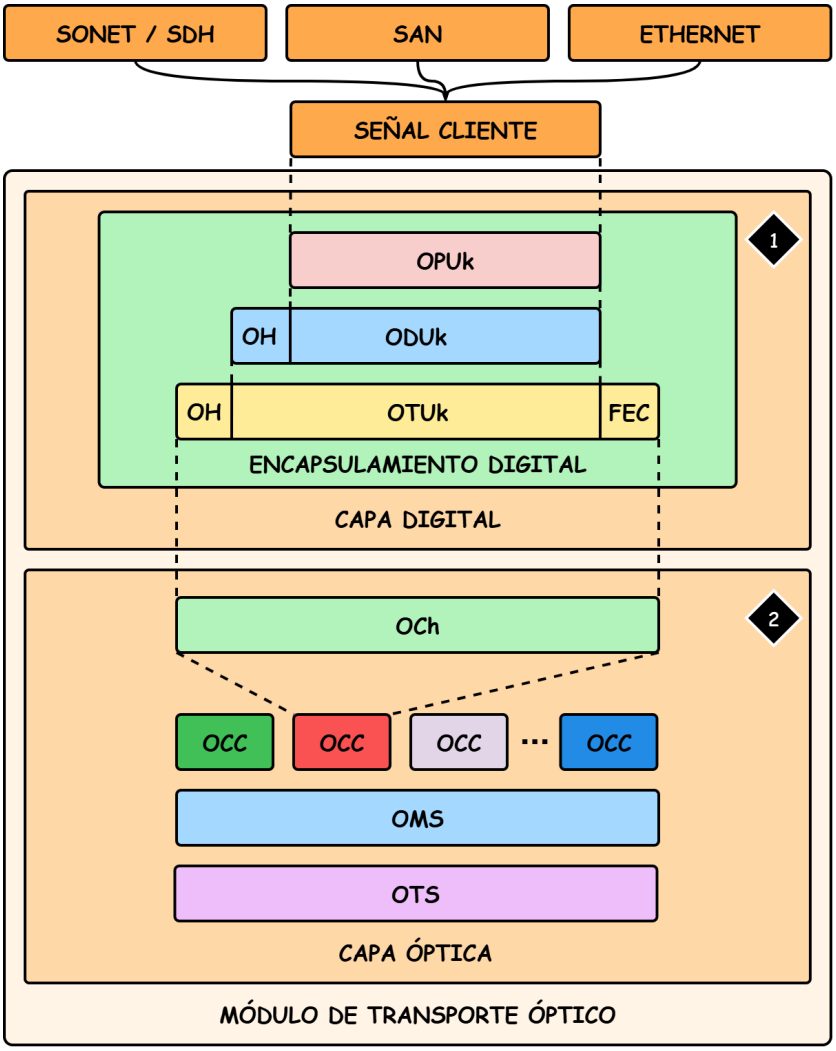

Figura 2.10: Representación de la jerarquía de capas en una red óptica de transporte, desde los servicios de cliente (Synchronous Optical Network (SONET)/Synchronous Digital Hierarchy (SDH), Storage Area Network (SAN), Ethernet) hasta las capas digital (indicación 1) y óptica (indicación 2), mostrando las entidades OPU, ODU, OTU, OCH, Optical Channel Carrier (OCC), OMS y OTS. La notación con subíndice k indica la capacidad de la señal OTN. Los valores válidos de k y sus bitrates asociados se establecen en G.709, y la arquitectura de cómo se relacionan estas capas se describe en G.872. Figura basada en Figura 6.3, sección 6.2 Information Structure for OTN Interfaces [21].

2.3.1. Plano de Transporte/Datos

El plano de transporte, también conocido como plano de datos, reúne las funciones y recursos que mueven la información a través de la red una vez que el servicio está establecido. En redes ópticas, esto abarca tanto el dominio fotónico (fibra, amplificación, conmutación óptica y multiplexación WDM) como el dominio digital OTN (encapsulado, multiplexación y conmutación de contenedores).

En términos funcionales, este plano se encarga de la transferencia de señales,

es decir, de la propagación y conmutación de canales ópticos y/o flujos OTN sobre la infraestructura. También incluye multiplexación y agregación, combinando múltiples señales sobre recursos compartidos: en el dominio óptico esto se logra multiplexando varias longitudes de onda sobre una misma fibra (WDM), mientras que en el dominio digital se agrupan varios servicios de menor tasa dentro de contenedores de mayor capacidad (grooming).

Además, incorpora funciones de OAM asociadas al transporte, que permiten supervisar el estado del servicio mediante mediciones y alarmas (por ejemplo, niveles de potencia, degradación y errores). Finalmente, contempla mecanismos de protección y resiliencia a nivel óptico y/o digital, orientados a mantener el servicio ante fallas, cuya activación puede estar coordinada por el plano de control o por el plano de gestión.

2.3.2. Plano de Control

El plano de control es el componente funcional que establece, mantiene y libera conexiones a través de la red óptica. Mientras el plano de transporte se encarga de mover efectivamente la información, el plano de control aporta la lógica operativa necesaria para crear y administrar servicios sobre los recursos disponibles. En este sentido, actúa como el "cerebro operativo" de la red, decidiendo cómo se construye una conexión y como se coordina su activación a lo largo de los nodos involucrados.

Estas funciones se materializan mediante protocolos y procedimientos que suelen agruparse en tres familias. En primer lugar, descubrimiento, orientado a conocer vecinos, recursos y topología (quién está conectado con quién y qué capacidades existen). En segundo lugar, cálculo de rutas, que determina caminos viables según restricciones del servicio y del dominio óptico (capacidad disponible, continuidad espectral, diversidad, entre otras). En tercer lugar, señalización, que permite reservar recursos, activar la conexión extremo a extremo y liberarla cuando corresponda, manteniendo además el estado del servicio.

Estas capacidades pueden utilizarse para reducir intervención manual en dos procesos centrales: el aprovisionamiento de servicios y la recuperación ante fallas. En la práctica, el aprovisionamiento suele ser planificado y disparado desde sistemas de gestión, mientras que el plano de control (cuando está presente) aporta mecanismos para reservar recursos y activar conexiones de forma consistente a lo largo del camino. Ante fallas, estas mismas funciones permiten calcular alternativas y coordinar la restauración del servicio, en complemento con mecanismos de protección que pueden existir en el propio plano de transporte.

2.3.3. Plano de Gestión

El plano de gestión comprende los sistemas y procesos utilizados para operar y administrar la red y sus servicios a lo largo de su ciclo de vida. En redes ópticas de transporte, esto incluye inventario de recursos (equipos, módulos, puertos y capacidades), configuración y provisión planificada de servicios, supervisión de desempeño, manejo de alarmas y fallas, auditorías, aplicación de políticas operativas

y generación de reportes. En la práctica, estas funciones se implementan mediante plataformas de operación que centralizan la visibilidad de la red y coordinan acciones sobre los elementos.

En relación con el plano de control, el plano de gestión actúa a un nivel más alto, traduciendo objetivos operativos en acciones de configuración y supervisión, y coordinando cambios planificados. Puede disparar la creación o modificación de un servicio, pero la materialización de la conexión a través de los nodos (asignación de recursos y establecimiento del camino) queda típicamente a cargo del plano de control o de mecanismos locales del plano de transporte.

La interacción con los elementos de red se apoya en interfaces y protocolos de gestión basados en modelos de datos. En este enfoque, YANG [8] define la estructura de la información que el equipo expone (objetos, atributos y relaciones). Sobre estos modelos, Network Configuration Protocol (NETCONF) [12] provee un protocolo para leer estado operativo y aplicar cambios de configuración de forma estructurada y consistente. En complemento, RESTful Configuration Protocol (RESTCONF) [7] ofrece acceso a esos mismos modelos mediante una interfaz Representational State Transfer (REST) sobre HyperText Transfer Protocol (HTTP), facilitando la integración con aplicaciones y el intercambio de datos en formatos como JavaScript Object Notation (JSON).

2.4. Arquitectura SDN

La arquitectura SDN puede entenderse como una evolución de la separación por planos, orientada a mejorar la operación mediante programabilidad. Mantiene la distinción entre el plano que transporta la información y la lógica que decide cómo se usan los recursos, pero refuerza su desacople al concentrar la función de control en software y al exponer interfaces estandarizadas para interactuar con la red. En redes ópticas de transporte, este enfoque favorece la automatización del aprovisionamiento y la recuperación de servicios, mejora la interoperabilidad en entornos multi-vendor y habilita una operación más dinámica en escenarios flexibles/elásticos.

El objetivo de SDN es reducir el acoplamiento entre la operación de la red y las herramientas propietarias de cada fabricante. En lugar de gestionar el aprovisionamiento y el monitoreo con sistemas específicos por plataforma, SDN propone abstraer capacidades de red y exponerlas mediante interfaces bien definidas. De esta forma, las aplicaciones pueden solicitar conectividad, consultar estado y automatizar procesos operativos sin interactuar equipo por equipo.

En SDN se distinguen tres planos funcionales: transporte/datos, control y aplicación. Como se ilustra en la Figura 2.11, la comunicación entre aplicación y control se realiza mediante la interfaz NBI, típicamente expuesta como APIs. A su vez, la interfaz Southbound Interface (SBI) conecta el plano de control con el de transporte y permite consultar estado y aplicar configuraciones sobre los elementos de red. Según el dominio y la tecnología, puede implementarse con protocolos como OpenFlow u otras interfaces orientadas a configuración y modelado (por ejemplo, NETCONF/RESTCONF), además de mecanismos específicos de proveedor.

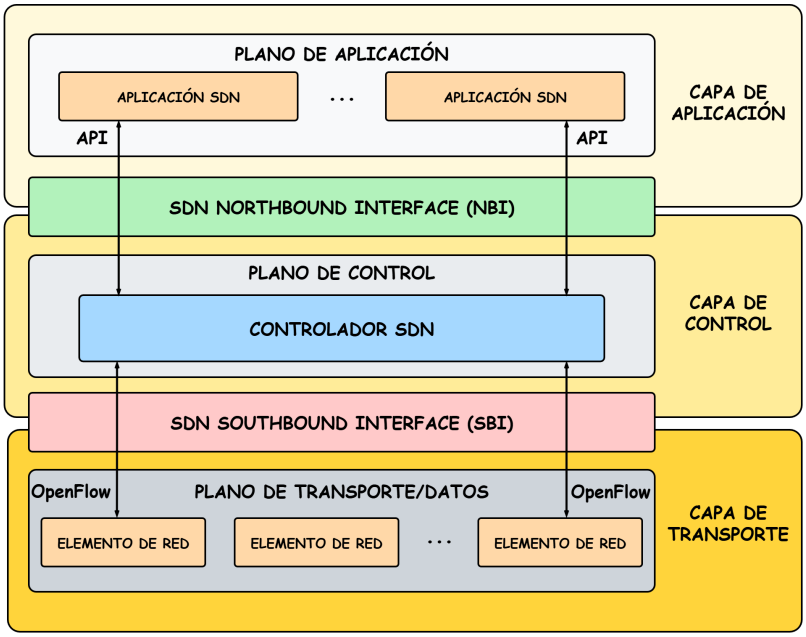

Figura 2.11: Arquitectura SDN compuesta por los planos de infraestructura, control y aplicación, mostrando su interacción mediante las interfaces NBI y SBI. La NBI se expone típicamente como Application Programming Interface (API)s, mientras que la SBI puede materializarse con protocolos de control o de gestión/configuración (según el dominio). Figura basada en la Figura 3.1, sección SDN overview [30] y la Figura 1, sección Introducing Software Defined Networking [29].

La separación funcional entre planos se sostiene, por tanto, en interfaces bien definidas. La SBI habilita que el controlador programe y supervise la infraestructura (por ejemplo, crear o modificar conectividad, configurar recursos y obtener métricas/estado). Por su parte, la NBI expone una vista más abstracta de la red (topología, conectividad y servicios) y permite que las aplicaciones consuman capacidades de red sin operar directamente sobre los equipos. Si bien ambas interfaces son necesarias para cerrar el ciclo operativo, el foco de este trabajo se centra en la NBI, ya que habilita el acceso programático a la información y a los servicios utilizados por la herramienta desarrollada.

En las Subsecciones siguientes se describe el rol de cada plano en la arquitectura SDN. La Subsección 2.4.1 presenta el plano de transporte/datos, la Subsección 2.4.2 el plano de control y la Subsección 2.4.3 el plano de aplicación. Además, se profundiza en el concepto de interfaz NBI, en este trabajo se adopta T-API como interfaz, la misma se describe en la Subsección 2.4.4.

2.4.1. Plano de Transporte/Datos

En SDN, el plano de transporte/datos corresponde al conjunto de recursos que materializan el transporte óptico y, cuando aplica, el transporte digital OTN. En este plano residen los recursos que se asignan al crear servicios (por ejemplo,

longitudes de onda, slots en flex-grid, puertos y contenedores OTN). Una vez establecido el servicio, este plano ejecuta la transferencia efectiva de señales y aplica las configuraciones necesarias en los elementos de red.

La diferencia respecto al enfoque por planos "clásico" es que estos recursos se exponen al plano de control mediante interfaces Southbound para su configuración y supervisión de forma programable. Esto incluye, por un lado, la capacidad de aplicar cambios de configuración (por ejemplo, habilitar un canal, asociar puertos o ajustar parámetros) y, por otro, la disponibilidad de información de estado y desempeño (ocupación de recursos, alarmas y métricas) que el controlador utiliza para tomar decisiones de provisión y recuperación.

2.4.2. Plano de Control

En SDN, el plano de control se implementa mediante un controlador SDN, una entidad lógicamente centralizada que mantiene una vista coherente de la red (topología, recursos, estados y restricciones) y coordina la provisión y operación de servicios. Se dice lógicamente centralizada porque la lógica se presenta como un único punto de control, aunque su implementación pueda ser distribuida por razones de escalabilidad y resiliencia.

El controlador se ubica entre el plano de aplicación y el plano de transporte, y su función es intermediar entre ambos mediante interfaces bien definidas. Hacia arriba, expone interfaces Northbound para que las aplicaciones soliciten servicios y consulten el estado de la red de forma abstracta. Hacia abajo, utiliza interfaces Southbound para configurar y supervisar los elementos del plano de transporte, es decir, para traducir una solicitud de conectividad en acciones concretas sobre la infraestructura.

En redes ópticas, esta traducción suele involucrar decisiones sobre el camino a utilizar y la asignación de recursos (por ejemplo, espectro/longitud de onda y puertos) y, cuando aplica, parámetros de transmisión asociados. Entre sus funciones típicas se incluyen el descubrimiento y actualización de topología y capacidades, el cálculo de caminos considerando restricciones (capacidad disponible, continuidad espectral y políticas), la coordinación del establecimiento, modificación y liberación de conexiones y la coordinación de acciones de recuperación ante fallas o degradaciones, ya sea restableciendo el servicio por una alternativa o integrándose con mecanismos de protección existentes en el plano de transporte.

2.4.3. Plano de Aplicación

El plano de aplicación agrupa los sistemas que consumen capacidades de red para cumplir objetivos operativos o de negocio. En una arquitectura SDN, estas aplicaciones no interactúan directamente con los equipos del plano de transporte. En su lugar, se comunican con elcontrolador SDN mediante interfaces NBI, a través de las cuales solicitan servicios (por ejemplo, conectividad entre extremos con determinados requisitos) y consultan información de red (topología, estado, disponibilidad y métricas).

Dentro de este plano suelen ubicarse componentes como orquestadores, sistemas de analítica, plataformas de visualización y herramientas de automatización. Su valor es operar sobre una abstracción de la red, delegando en el controlador la traducción de esos requerimientos a acciones concretas sobre el plano de transporte.En el contexto de este proyecto, el plano de aplicación es central porque constituye el punto de consumo de la información expuesta por la red: habilita el acceso estandarizado a datos de topología, servicios y desempeño para procesos de extracción y visualización, evitando acoplarse a mecanismos propietarios o a detalles específicos de proveedor.

2.4.4. T-API como Interfaz NBI

La interfaz NBI es el punto de interacción entre el plano de aplicación y el controlador SDN. Su propósito es exponer a las aplicaciones una vista abstracta y consistente de la red, evitando que deban operar con detalles de bajo nivel del plano de transporte o con particularidades de cada fabricante. En términos prácticos, una NBI permite consultar topología e inventario lógico (nodos, enlaces, puntos de terminación y capacidades), acceder a estado operativo y desempeño (estados, alarmas, utilización y métricas) y solicitar o administrar servicios de conectividad (creación, modificación y liberación), incluyendo parámetros y restricciones relevantes para el servicio.

En redes de transporte, estas interfaces suelen definirse mediante modelos de información y APIs orientadas a servicios, con el objetivo de favorecer interoperabilidad en entornos multi-vendor. En este contexto, una opción ampliamente utilizada es T-API de Open Networking Foundation (ONF), que estandariza entidades e interfaces para acceder a topología, conectividad y servicios de transporte.

T-API es un estándar de ONF que define un modelo e interfaces para interactuar con redes de transporte a nivel de servicio [31]. Su objetivo es desacoplar a las aplicaciones de las particularidades de cada proveedor, ofreciendo una forma uniforme de acceder a información de red y de gestionar conectividad. En este esquema, un cliente T-API (por ejemplo, un orquestador o una aplicación de operación) consume la interfaz para consultar topología y capacidades, y para solicitar o administrar servicios de conectividad. Del lado opuesto, un servidor T-API expone dichas capacidades y se ubica típicamente en el plano de control, asociado a un controlador SDN o a una función equivalente de control de dominio.

En el marco de SDN, T-API se utiliza como interfaz estándar en la NBI, habilitando la comunicación entre aplicaciones y el plano de control. Fue diseñado para facilitar la adopción de SDN en infraestructuras de transporte multi-capa, multi-dominio y multi-vendor, extendiendo la programabilidad de la red mediante entidades y operaciones comunes.

Las especificaciones del modelo de T-API se basan en el Common Information Model (CIM) de ONF y se definen en Unified Modeling Language (UML), de forma independiente del protocolo. A partir de ese modelo existen mapeos a lenguajes de definición como YANG, lo que permite exponer T-API mediante mecanismos basados en modelos y APIs de gestión [8, 32]. Los módulos YANG que componen

el modelo de información de T-API se presentan en la tabla 2.1 [28].

Tabla 2.1: Módulos YANG del modelo T-API [28].

Módulo
`tapi-common.yang`
`tapi-connectivity.yang`
`tapi-dsr.yang`
`tapi-equipment.yang`
`tapi-eth.yang`
`tapi-notification.yang`
`tapi-oam.yang`
`tapi-odu.yang`
`tapi-path-computation.yang`
`tapi-photonic-media.yang`
`tapi-streaming.yang`
`tapi-topology.yang`
`tapi-virtual-network.yang`

En implementaciones basadas en YANG, la comunicación entre el cliente y el servidor T-API puede materializarse mediante RESTCONF [7]. RESTCONF es un protocolo sobre HTTP que provee una interfaz programática estandarizada para acceder y operar datos definidos en YANG. En este enfoque, YANG define la estructura de los objetos de información (por ejemplo, topologías, nodos, puntos de terminación y servicios), mientras que RESTCONF define el mecanismo para consultarlos y modificarlos a través de recursos web.

Según el RFC-8040, el punto de entrada de RESTCONF se identifica mediante el recurso raíz `+restconf`, a partir del cual se accede a tres recursos principales: `data`, `operations` y `yang-library-version` [7]. El recurso `data` expone el árbol de datos conforme a los modelos YANG disponibles (incluyendo los módulos asociados a T-API). El recurso `operations` expone las operaciones soportadas por el servidor sobre esos modelos. Por su parte, `yang-library-version` permite identificar las versiones de los módulos YANG publicados, lo que resulta relevante para asegurar compatibilidad entre cliente y servidor. De esta forma, RESTCONF habilita el acceso web a los datos modelados y permite intercambiarlos en formatos de serialización como JSON [11] o eXtensible Markup Language (XML), según la preferencia o compatibilidad de la aplicación.

La principal ventaja de T-API es que permite acceder a dominios con distintas tecnologías mediante un marco común independiente del proveedor. Esto facilita la integración tanto en entornos nuevos como existentes sin necesidad de reemplazar infraestructura, y contribuye a una adopción más simple de SDN en redes de transporte multi-vendor.

2.5. Fundamentos de Datos para Operación de Redes

La operación de redes ópticas de transporte depende, en forma creciente, de la disponibilidad y del uso sistemático de datos. A diferencia de otros dominios, el desempeño de un servicio óptico no queda determinado únicamente por la conectividad lógica: la señal está condicionada por fenómenos físicos de propagación (atenuación, ruido, filtrado, dispersión y no linealidades) que pueden degradar la calidad de transmisión (QoT) aun cuando no se observen fallas evidentes a nivel de conectividad. En este contexto, disponer de datos operativos, como por ejemplo, inventario y topología de recursos, configuración, estado operativo, métricas de desempeño y alarmas/eventos, es clave para supervisar la salud de la red, diagnosticar degradaciones, planificar capacidad y sostener procesos de provisión y recuperación.

Con este objetivo, esta sección presenta el marco conceptual de datos utilizado en el proyecto, estableciendo definiciones y criterios que se emplearán en los capítulos posteriores. En la Subsección 2.5.1 se describen las fuentes de datos y su modo de generación, y en la Subsección 2.5.2 se clasifica la naturaleza de los datos según su formato. Luego, la Subsección 2.5.3 introduce los modelos de datos considerados y sus implicancias de diseño, mientras que la Subsección 2.5.4 aborda su materialización en sistemas de almacenamiento. Finalmente, la Subsección 2.5.5 define el concepto de pipeline de datos con foco en ETL, sus etapas principales y consideraciones operativas asociadas.

2.5.1. Fuentes de Datos

Las fuentes pueden originarse en APIs, bases de datos, dispositivos o directorios de archivos, con contenidos sujetos a variaciones de calidad y a diferentes ritmos de generación (flujos continuos o conjuntos finitos). Su evolución en el tiempo puede estar bajo control propio o de terceros. Antes de diseñar la solución, es imprescindible caracterizar estas fuentes: validar el propósito de uso, la frecuencia de recepción, la finitud o continuidad del conjunto, el volumen esperado y el nivel de calidad [33].

2.5.2. Formato de Datos

Según su naturaleza, los datos pueden ser no estructurados, semiestructurados o estructurados. Los no estructurados carecen de un esquema predefinido, como por ejemplo texto, audio, video o imágenes. Los semiestructurados no siguen un esquema rígido, pero incluyen metadatos que inducen jerarquías implícitas, como JSON o XML. Los estructurados se rigen por un esquema definido, típicamente en modelos relacionales.

2.5.3. Modelos de Datos

Los modelos de datos describen cómo se representan y organizan los datos. El diseño adoptado condiciona su capacidad de expresión y las operaciones que

habilita, por lo que debe definirse en función del propósito de uso. En esta sección se abordan dos familias: modelos relacionales y modelos NoSQL.

Modelos Relacionales

El modelo relacional organiza los datos en relaciones (tablas) que son conjuntos de tuplas (filas) descritas por atributos (columnas). Al ser conjuntos, ni el orden de las filas ni el de las columnas importa, pueden intercambiarse manteniendo el mismo resultado. Una tabla suele representar una entidad y cada fila, una instancia.

Cada fila se identifica de forma única mediante una clave primaria. Las claves externas son atributos cuyo valor debe coincidir con una clave primaria existente en otra tabla; su verificación constituye la integridad referencial, que evita referencias a filas inexistentes y mantiene la coherencia entre tablas [10]. Para minimizar la redundancia y preservar la coherencia del modelo, el diseño se normaliza con base en dependencias funcionales. La normalización descompone en formas normales de modo que cada hecho se represente una sola vez y los atributos dependan de la clave, del conjunto completo de la clave y de nada más.

Modelos NoSQL

Para ciertos casos de uso, el modelo relacional puede resultar restrictivo, especialmente cuando los datos carecen de un esquema completamente definido o este cambia con frecuencia, dificultando su representación en tablas normalizadas. En estos escenarios cobran relevancia los modelos no relacionales, entre los cuales destacan los documentales y los de grafos. En este enfoque se aborda el primero.

En el modelo documental, la unidad es el documento: una estructura jerárquica (JSON, Binary JSON (BSON) o XML) que reúne la información de una entidad y se identifica con una clave única. A diferencia del enfoque relacional, una colección no impone un esquema rígido, por lo que pueden coexistir documentos con estructuras distintas. Esta flexibilidad, junto con la mayor localidad de los datos (información de una entidad en un único documento), simplifica ciertas lecturas frente a diseños relacionales normalizados que suelen requerir múltiples uniones.

2.5.4. Almacenamiento de Datos

El almacenamiento se elige según el propósito de uso, el volumen y la cadencia de escritura, el formato y el modelo de organización de los datos. Dependiendo del esquema adoptado, suelen emplearse enfoques diferentes. A efectos del proyecto, se consideran dos: el relacional y el documental. Ambos se materializan a través de sistemas de gestión de bases de datos (Sistema de Gestión de Bases de Datos (SGBD)), que permiten definir el esquema, cargar la información y consultarla de forma controlada.

Esquemas relacionales

Organizan los datos en tablas (filas y columnas) con un esquema definido. Las relaciones entre tablas permiten representar vínculos entre entidades y consultar información combinada. Se utilizan cuando la estructura es estable y las reglas de integridad entre datos son importantes. Las consultas se expresan, en general, en lenguaje Structured Query Language (SQL).

Esquemas documentales

Representan cada entidad como un documento jerárquico (por ejemplo, similar a JSON). Ofrecen mayor flexibilidad de estructura y facilitan leer la entidad completa en una sola operación. Se prefieren cuando los datos varían con frecuencia en forma o profundidad, o cuando el modelo es naturalmente jerárquico.

2.5.5. Pipeline de Datos

Un pipeline de datos es un flujo que permite mover, extraer y transformar información entre una o más fuentes y destinos. El flujo y las etapas varían según las necesidades específicas del problema.

En esta Subsección se adopta el enfoque ETL, un patrón común que organiza el proceso en tres etapas diferenciadas: extracción, desde una o más fuentes; transformación para adaptar y enriquecer los datos al esquema y uso objetivo; y carga hacia los destinos definidos. El diseño de cada etapa depende del problema: tipo y calidad de las fuentes, requisitos de rendimiento y objetivo de uso. Este enfoque permite procesar datos crudos y dejarlos listos para su consumo en una fuente de almacenamiento. La Figura 2.12 ilustra el proceso de punta a punta.

Extracción de Datos

La extracción captura los datos desde las fuentes disponibles y define el ritmo y la forma de adquisición. Puede ejecutarse de manera periódica (time-driven), impulsada por eventos (event-driven) o de forma manual (ad-hoc). Respecto del modo de procesamiento, puede ser por lotes (batch), donde se acumulan y procesan bloques de datos, o en flujo continuo (streaming), donde se leen y procesan los datos a medida que se reciben. La elección condiciona la latencia inicial y el volumen que recibirán las etapas posteriores del pipeline, y se alinea con los requisitos específicos del problema. En esta fase también se aplican verificaciones básicas de integridad y calidad: se aíslan o descartan registros mal formados y se controlan, por ejemplo, esquema y tipos, rangos válidos, consistencia y deduplicación, para asegurar que ingresen datos aptos para el procesamiento y el almacenamiento posteriores.

Transformación de Datos

La transformación adapta y enriquece la información para satisfacer los requerimientos del problema y el esquema objetivo. Abarca desde acciones simples

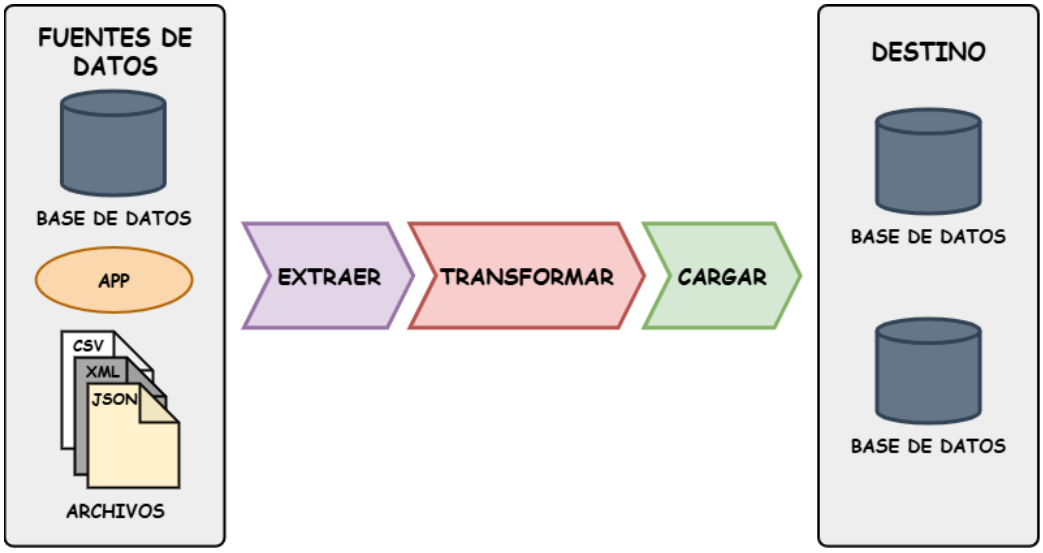

Figura 2.12: Esquema representativo de un proceso ETL, mostrando las fuentes de datos, las etapas de extracción, transformación y carga, y su destino en bases de datos. Fuente basada en Figura 3.7, Cap.3 Data Engineering Fundamentals [16] .

(filtrado, eliminación de duplicados, conversión de tipos) hasta procesos que reestructuran los datos (uniones/fusiones, mapeo al esquema destino, agregaciones y, cuando corresponde, particionado o descomposición de estructuras complejas). También puede incluir enriquecimiento con atributos provenientes de fuentes internas o externas.

Carga de Datos

La carga almacena los datos transformados en los destinos definidos y define el ritmo y la granularidad de escritura. Puede ejecutarse de forma periódica, impulsada por eventos o manual, y materializarse en lotes o en flujo continuo. En esta fase se establecen el particionado y la organización del destino (por tiempo o por claves) y se aplican controles para asegurar la consistencia de los datos y la compatibilidad de esquema.

Aspectos Transversales

La operación del pipeline de extremo a extremo se apoya en principios que atraviesan sus etapas. La observabilidad permite visibilizar el comportamiento de las ejecuciones mediante registros estructurados, métricas e indicadores de estado. La trazabilidad identifica el linaje de datos (origen, transformaciones y destinos), facilitando auditorías y reconstrucción del recorrido. El versionado y la reproducibilidad gestionan versiones de código, configuraciones y artefactos de datos para repetir resultados de manera controlada ante los mismos insumos. Finalmente, latencia y throughput caracterizan, respectivamente, el tiempo hasta disponibilidad de los datos y la capacidad sostenida de procesamiento por unidad de tiempo,

guiando el dimensionamiento operativo. Aplicar estos principios de forma consistente en extracción, transformación y carga favorece una operación predecible y facilita depuración y reprocesos. En particular, la persistencia en zonas intermedias, temporales o persistentes, puede actuar como punto de control entre etapas, desacoplando componentes y mejorando la trazabilidad [33].

Orquestación de Datos

La orquestación de datos es la coordinación automatizada de un conjunto de tareas para componer un flujo de trabajo. Un orquestador define dependencias entre tareas, establece disparadores (por tiempo o por eventos) y procura que cada etapa se ejecute en el orden previsto, con manejo de errores, reintentos y notificaciones. También centraliza el seguimiento operativo (registros de ejecución, estados y métricas) y facilita la integración con sistemas externos donde se realizan las ingestas, transformaciones y cargas [33].

2.6. Resumen

Este capítulo estableció los fundamentos teóricos del proyecto, construyendo el contexto desde el transporte óptico sobre fibra hasta el uso de datos para la operación de la red. Se introdujeron la fibra óptica y el uso del espectro en sistemas WDM/DWDM, junto con las bandas más empleadas para transmisión. Se abordó la planificación y asignación espectral, partiendo de esquemas Fixed-Grid y su evolución hacia Flex-Grid y redes ópticas elásticas, destacando cómo estas tecnologías mejoran la eficiencia en el aprovechamiento del espectro.

De igual forma, se caracterizaron los principales elementos de una red óptica de transporte: Transponders (incluyendo variantes BVT/SBVT), ROADMs y sus bloques funcionales (CDC, Add/Drop, Express y WSS), y los amplificadores ópticos de línea (OLA). En complemento, se presentaron los enfoques de transmisión directa y coherente, enfatizando el compromiso alcance–capacidad y su relación con el uso del espectro: la detección directa se asocia típicamente a canales de ancho fijo en esquemas Fixed-Grid, mientras que la transmisión coherente habilita configuraciones de mayor eficiencia espectral y se alinea con esquemas Flex-Grid y redes ópticas elásticas, donde el ancho de ranura puede ajustarse al formato de modulación, la tasa y el margen disponible (OSNR/QoT).

Sobre esta base, se incorporó OTN como arquitectura de transporte digital, describiendo sus componentes de capa digital y de medios ópticos y las entidades que estructuran y supervisan servicios extremo a extremo. Luego, se introdujo la organización funcional por planos (transporte, control y gestión) para comprender cómo se establecen, mantienen y administran conexiones y servicios en la red. En este marco se presentó SDN como enfoque de automatización e interoperabilidad, y se adoptó T-API como interfaz NBI para habilitar la comunicación entre el plano de aplicación y el plano de control. Al exponer de forma estandarizada información y servicios de transporte, T-API abstrae a las aplicaciones de los detalles y mecanismos de interacción con la red subyacente, evitando la operación directa equipo

por equipo.

Finalmente, se estableció el marco conceptual de datos del proyecto. Se definieron fuentes, formatos y modelos de datos, se discutieron alternativas de almacenamiento relacional y documental, y se introdujo el enfoque de pipelines ETL con sus etapas y consideraciones operativas. Con esta base conceptual, el capítulo siguiente caracteriza el dominio del caso de estudio y prepara el diseño del modelo de datos y los procesos de integración que sustentan la herramienta de visualización.

Capítulo 3

Caso de Estudio: ANTEL

En este capítulo se presenta el caso de estudio sobre el dominio de ANTEL, con foco en comprender la estructura y el comportamiento de su red óptica de transporte operada con equipamiento Infinera. El análisis se apoya en los archivos de monitoreo disponibles y en la información que estos exponen tanto desde la perspectiva del dominio de ANTEL como desde el dominio propio del fabricante.

El capítulo se organiza de la siguiente manera. En la Sección 3.1 se describe el dominio considerado, sus componentes principales y el marco de gestión mediante el cual se expone la información operativa. En la Sección 3.2 se detallan las fuentes utilizadas para el análisis, incluyendo los archivos `common-context` y `PM Data`. Luego, la Sección 3.3 desarrolla el análisis de la red, distinguiendo entre la visión de hardware y servicios de bajo nivel, y la visión de topología y servicios de alto nivel. Finalmente, la Sección 3.4 resume los resultados obtenidos y presenta las conclusiones principales del estudio.

3.1. Visión General de la Red

El caso de estudio corresponde a un dominio de red óptica de transporte de ANTEL provisto por Infinera. Se trata de una infraestructura fotónica en la que los servicios se establecen como canales ópticos entre extremos, se encaminan mediante conmutación óptica en nodos intermedios y se sostienen mediante amplificación en los tramos de fibra. En el dominio analizado, estas funciones se materializan en Transponders, nodos ROADM y amplificadores ópticos de línea (OLA).

El estudio del dominio se apoya en dos insumos. Por un lado, la documentación Transcend REST Northbound Interface (TRNBI) provista por ANTEL [19], que describe la interfaz NBI de la plataforma Transcend para acceder a información y servicios del dominio de transporte. Por otro, los archivos de monitoreo de la red, incluyendo el `common-context` del dominio y las métricas de desempeño (`PM Data`). En conjunto, estos materiales permiten analizar el dominio desde una perspectiva operacional a partir de inventario lógico, topología, conectividad y desempeño.

3.1.1. Alcance del Dominio Considerado

Este capítulo se limita al dominio de transporte óptico de ANTEL provisto por Infinera, tal como es observado a través de la interfaz Northbound de Transcend y de los archivos de monitoreo disponibles. En consecuencia, el análisis se restringe a los recursos, relaciones y servicios expuestos por esa vista (inventario lógico, topología, conectividad y métricas PM), quedando fuera de alcance cualquier información no publicada por la o perteneciente a otros dominios o tecnologías.

3.1.2. Componentes de la Infraestructura Óptica

En este dominio, un servicio extremo a extremo se interpreta como un canal óptico entre dos Transponders que atraviesa, o no, una secuencia de nodos ROADM y tramos amplificados por OLA. En particular:

- **Transponder:** termina el servicio en los extremos y realiza la adaptación entre señales cliente y señales ópticas transportadas sobre fibra, definiendo los puntos de terminación y parámetros de transmisión asociados.
- **ROADM:** conmuta longitudes de onda en nodos intermedios, permitiendo agregar, extraer o encaminar canales ópticos sin conversión O-E-O para construir conectividad a través de múltiples tramos.
- **OLA:** compensa la atenuación acumulada en los tramos de fibra y elementos pasivos, manteniendo niveles de potencia dentro de rangos objetivo e impactando en la calidad de transmisión y el margen operativo.

Esta abstracción es suficiente para contextualizar cómo se representan, en los datos, los recursos del dominio, su topología y los servicios analizados en las secciones siguientes.

3.1.3. Gestión del Dominio y Exposición de Información

La red se gestiona mediante la plataforma Transcend de Infinera, que expone una interfaz REST Northbound orientada a automatización e integración con aplicaciones externas [1]. En este trabajo, la NBI se utiliza como punto de acceso a información operativa del dominio (entidades, relaciones de conectividad, vistas de topología y servicios), complementada con los archivos de monitoreo.

La interfaz se documenta mediante TRNBI e implementa un subconjunto de T-API v2.1.3 [5]. Esto permite interpretar el dominio bajo un modelo estandarizado y trabajar sobre una capa de abstracción que expone objetos y relaciones de más alto nivel (topología, puntos de terminación, conectividad y servicios), evitando depender de interfaces de bajo nivel por equipo. Sobre esta base se construyen las vistas exploradas en el resto del capítulo.

3.2. Análisis de Fuentes de Datos

Esta sección describe las dos fuentes de datos utilizadas para analizar el dominio de transporte óptico de ANTEL provisto por Infinera. Ambas fuentes se complementan. Los archivos `common-context` aportan la representación estructural del dominio (inventario lógico, topología y conectividad), mientras que los archivos `PM Data` aportan la dimensión de desempeño, es decir, mediciones periódicas asociadas a entidades operativas del mismo dominio. En conjunto, estas fuentes permiten estudiar la red desde una perspectiva operacional, permitiendo conocer qué elementos existen, cómo se relacionan y qué métricas se observan sobre ellos.

En la Subsección 3.2.1 se presenta el `common-context` y sus subcontextos principales. Luego, la Subsección 3.2.2 introduce los archivos `PM Data`, explicando cómo se generan a través de la PM API, qué entidades se pueden monitorear y cómo se estructura un registro de mediciones.

3.2.1. common-context

Uno de los pilares del análisis son los archivos de monitoreo de tipo `common-context`. Estos archivos organizan la información en distintos `context`, entendidos como espacios lógicos que agrupan representaciones de recursos de red para su intercambio a través de una interfaz. En T-API, un `context` se apoya en la relación cliente/servidor y permite aislar y estructurar la información expuesta a través de la interfaz [5]. En el caso estudiado, el `common-context` corresponde al espacio compartido entre el cliente T-API (operación en ANTEL) y el servidor T-API (plataforma Infinera), donde se publican estructuras y parámetros del dominio asociados a la provisión y supervisión de servicios.

Para soportar estos servicios se consideran tres subcontextos principales: `equipment-context`, `topology-context` y `connectivity-context`. El `equipment-context` describe el equipamiento desplegado y expone entidades con granularidad física/lógica del fabricante. En este subcontexto, las entidades centrales son Device, Access-Port y Physical-Span. Según la documentación TRNBI, un Device corresponde a un elemento de red (Network Element (NE)); el dispositivo se compone por módulos (Equipment) alojados en contenedores (`holders`). Un Access-Port representa un puerto físico del NE. Por su parte, un Physical-Span representa el par de fibras que conecta dos Access-Port pertenecientes a NE distintos.

El `topology-context` abstrae el inventario anterior para representar la red como un grafo de nodos y enlaces. En este subcontexto existe una correspondencia uno a uno con entidades del `equipment-context`: Node abstrae a Device, Node Edge Point (NEP) abstrae a Access-Port y Link abstrae a Physical-Span (ver indicación 1, Figura 3.1). Esta abstracción permite describir conectividad sin depender del detalle interno de los equipos. A nivel físico, la transmisión se realiza sobre dos fibras, una por cada sentido. En cambio, en el `topology-context` ambos sentidos se representan como un único Link bidireccional. Por ejemplo, entre el Device A-GX y el Device A-FLEX existen dos Physical-Span (uno por cada dirección), pero en la topología se observan como un solo Link (ver indicación 2, Figura 3.1), simplificando el modelo para su consumo por aplicaciones.

Finalmente, el `connectivity-context` modela la conectividad provista sobre la topología mediante la entidad Connection. Una Connection representa una capacidad de transporte habilitada entre dos o más NEP pertenecientes a uno o varios Node, y constituye la base para describir conectividad a bajo nivel dentro del dominio.

Además, el `common-context` define Service Interface Point (SIP) como punto de acceso de servicio. Un SIP puede representar un requerimiento de cliente o, alternativamente, un punto expuesto hacia aplicaciones T-API para solicitar servicios o consultar capacidades. En este trabajo, estas entidades se utilizan para reconstruir la vista estructural del dominio (equipos, puertos, enlaces y conectividad) y vincularla con las métricas de desempeño.

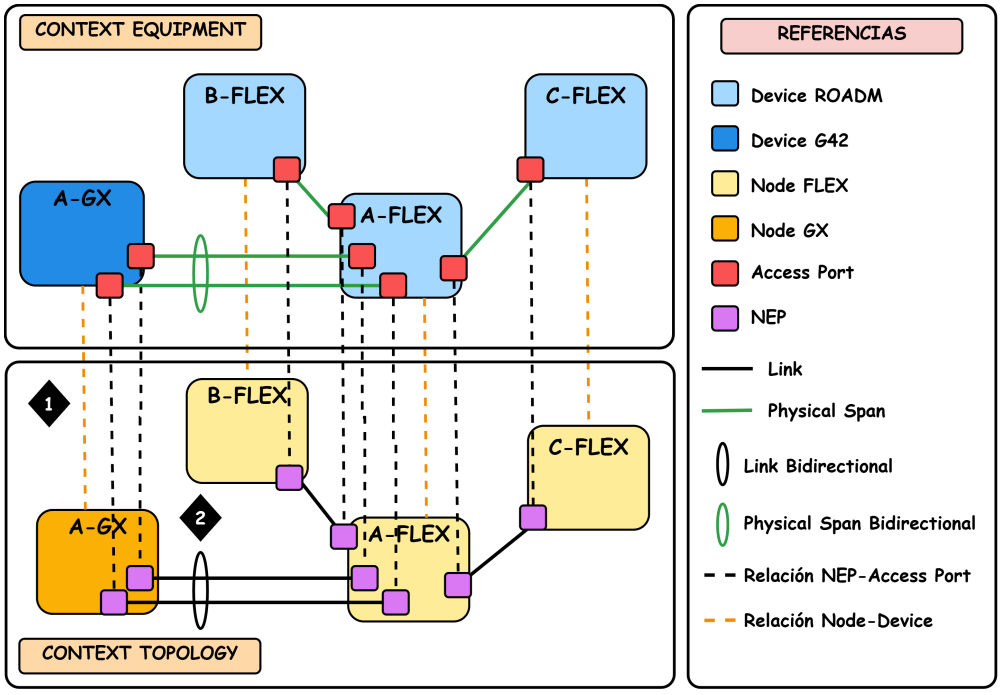

Figura 3.1: Relación entre `equipment-context`, `topology-context` y sus entidades, mostrando cómo los Node son vinculados a los Devices, los NEP a los Access-Port y los Link a los Physical-Span, incluyendo la representación de Node de tipo GX y FLEX.

3.2.2. PM Data

La segunda fuente de datos corresponde a los archivos `PM Data`, que contienen mediciones de desempeño de la red y permiten observar, de forma cuantitativa, el comportamiento del dominio a lo largo del tiempo. En el entorno Infinera, estos archivos se generan mediante la PM API, utilizada para solicitar colecciones de mediciones al plano de control y obtenerlas en formato JSON.

La PM API está basada en el estándar TMF628 [38] de TM Forum (TeleManagement Forum) (TMF) Open APIs y soporta operaciones para crear, consultar,

actualizar y eliminar colecciones de métricas. En el caso de extracción, el mecanismo central son las solicitudes *On Demand Collections*, que definen qué métricas se desea medir, sobre qué entidades de red y para qué intervalo temporal. Para describir este proceso se introducen tres conceptos:

- **Performance Indicator (Performance Indicator (PI)):** métrica individual de desempeño (por ejemplo, temperatura o consumo de corriente de un módulo).
- **Performance Indicator Group (Performance Indicator Group (PIG)):** agrupación de PIs asociadas a un mismo aspecto o entidad (por ejemplo, un conjunto de métricas eléctricas y térmicas de hardware).
- **Entidades:** objetos del dominio sobre los que se solicitan mediciones. En el caso analizado se utilizan Device, Equipment, Access-Port y `connection`.

Una *On Demand Collection* especifica uno o más PIG y el conjunto de entidades sobre las que se medirá, junto con una granularidad temporal (15 minutos o 1 hora) y una ventana de observación. Como resultado, el sistema genera archivos `PM Data` que contienen las mediciones correspondientes a cada PIG solicitado.

Cada archivo `PM Data` se compone de registros estructurados según un esquema propietario de Infinera (Figura 3.2). En cada registro, el campo `name` identifica el PIG representado. Los campos `pmpLocation` y `pmpDirection` indican, respectivamente, la ubicación lógica y el sentido sobre el cual se realizan las mediciones. El campo `pmpGranularity` fija la periodicidad con la que se reportan los valores. Las mediciones se agrupan en `values`, una lista donde cada elemento corresponde a un intervalo temporal determinado por la granularidad configurada.

Para cada elemento de `values`, se incluyen marcas temporales que permiten interpretar el significado de la medición. El campo `periodEndTime` representa el instante final del intervalo lógico al que corresponde el conjunto de mediciones reportadas. El campo `retrievalTime` indica el instante en que el sistema extrajo efectivamente el objeto de datos desde el plano de control. El campo `monitoredTime` expresa la duración del intervalo sobre el cual se realizó la observación, lo que permite distinguir mediciones instantáneas de mediciones promediadas. Finalmente, dentro de cada elemento de `values`† se incluye la lista de mediciones por PI: `name`† identifica la métrica, `indicatorUnit` su unidad, `value` el resultado reportado y `status` indica si la medición es válida.

Los PIG se reportan para distintas entidades del dominio. Si bien una consulta puede referir a Equipment, Access-Port o Connection, los registros siempre incluyen el Device asociado, lo que permite vincular cada medición con el elemento de red correspondiente (Transponder, ROADM u OLA). En el conjunto de archivos analizados se observan PIG orientados a recursos de hardware (por ejemplo, temperatura, consumo y utilización), y otros vinculados al transporte óptico y digital, alineados con la estructura de capas de OTN. En este último grupo aparecen métricas como niveles de potencia, OSNR, Bit Error Rate (BER), Q-Factor y métricas asociadas a FEC (pre/post-FEC), entre otras.

```
[
    {
        pmpMonitoredObject
        name
        pmpLocation
        pmpDirection
        pmpGranularity
        values  [
                    {
                        retrievalTime
                        periodEndTime
                        monitoredTime
                        nrOfPeriods
                        values† [
                                    {
                                        name†
                                        indicatorUnit
                                        value
                                        status
                                    }
                                ]
                    }
                ]
    }
]
```

Figura 3.2: Estructura de un registro de `PM Data`. Se omiten los campos `pmpRelatedPaths`, `nrOfPeriods` y `pmpRelatedSuscribers` para simplificar la visualización, ya que no están presentes en los registros obtenidos a partir de los archivos proporcionados por ANTEL.

En capítulos posteriores estas mediciones se estudian con mayor detalle. En esta instancia interesa destacar su rol en el caso de estudio. Los archivos `PM Data` aportan la serie temporal de desempeño del dominio y, al poder asociarse a entidades del `common-context`, habilitan combinar estructura (inventario/topología/conectividad) con estas métricas para el diseño del modelo de datos y del pipeline de integración.

3.3. Análisis de la Red

En esta sección se presenta un análisis completo de la red de transporte, abordando tanto la topología física como su representación abstracta y cómo los servicios se estructuran y transportan a través de esta red.

Primero, se detalla la topología física de la red en la Subsección 3.3.1, donde se describe la infraestructura de hardware y sus componentes clave, como los NE de las familias GX y FLEX, y cómo estos elementos permiten el transporte de datos. Posteriormente, en la Subsección 3.3.2, se aborda la topología abstracta, explicando

cómo la red es representada a través del `topology-context` y cómo se construyen los servicios sobre esta representación abstracta.

Este enfoque proporciona una visión integral, comenzando desde los componentes físicos, pasando por su modelado lógico, y culminando en la construcción y el transporte de los servicios. El análisis incluye ejemplos detallados que ilustran cómo las abstracciones lógicas y físicas se interrelacionan en la práctica.

3.3.1. Topología Física

La red está conformada por un conjunto de NE interconectados mediante fibra óptica. Estos equipos, según la nomenclatura de Infinera, se agrupan en dos familias: GX y FLEX. Los equipos de la familia GX, concentra los Transponders y Muxponders, que son responsables de terminar señales cliente y adaptarlas al transporte óptico. Por otro lado, los nodos FLEX, que incluyen ROADM y OLA, gestionan el enrutamiento de longitudes de onda en tránsito y compensan las pérdidas acumuladas en los enlaces.

GX Series

Para Infinera, un GX es una plataforma modular perteneciente a la serie GX. En el dominio de ANTEL, el modelo utilizado es el G42, subdividido en siete Slots, cada uno con una función específica:

- **Slots 1 y 3:** módulos de administración.
- **Slot 2:** módulo de reporte de alarmas.
- **Slots 4 a 7:** módulos de servicio (placas CHM6) [2], que actúan como Transponder o Muxponder (ver indicación 1, Figura 3.3).

Tal como se muestra en la indicación 2 de la Figura 3.3, cada placa CHM6 integra:

- **16 puertos Tributary (cliente):** admiten QSFP-DD (sólo en puertos 1, 8, 9 y 16, hasta 400 *Gb*) y QSFP28 (en todos los puertos, hasta 100 *Gb*).
- **2 puertos Line (Lambda):** configurables de forma independiente entre 100 *Gb* y 800 *Gb* cada uno. La capacidad agregada puede alcanzar hasta 1,6 *Tbps* en transmisión/recepción.
- **1 puerto lógico Containment:** orquesta la asociación cliente-lambda.

El puerto Containment es una entidad lógica que gestiona la vinculación entre puertos Tributary y puertos Line. Operativamente, el cliente ingresa por un módulo QSFP28/QSFP-DD en un puerto Tributary y el Containment lo asocia a uno de los dos puertos Line, de acuerdo con la provisión del servicio.

La salida hacia la red se realiza por un puerto Line (Lambda). Estos puertos se configuran por frecuencia central, ancho de banda, formato de modulación, baud

rate, potencia y el extremo remoto (otro puerto Line en una CHM6 de un NE GX). La conectividad óptica punto a punto resultante se denomina Lightpath, y es el recurso que transporta el servicio atravesando nodos FLEX (ROADM y/o OLA).

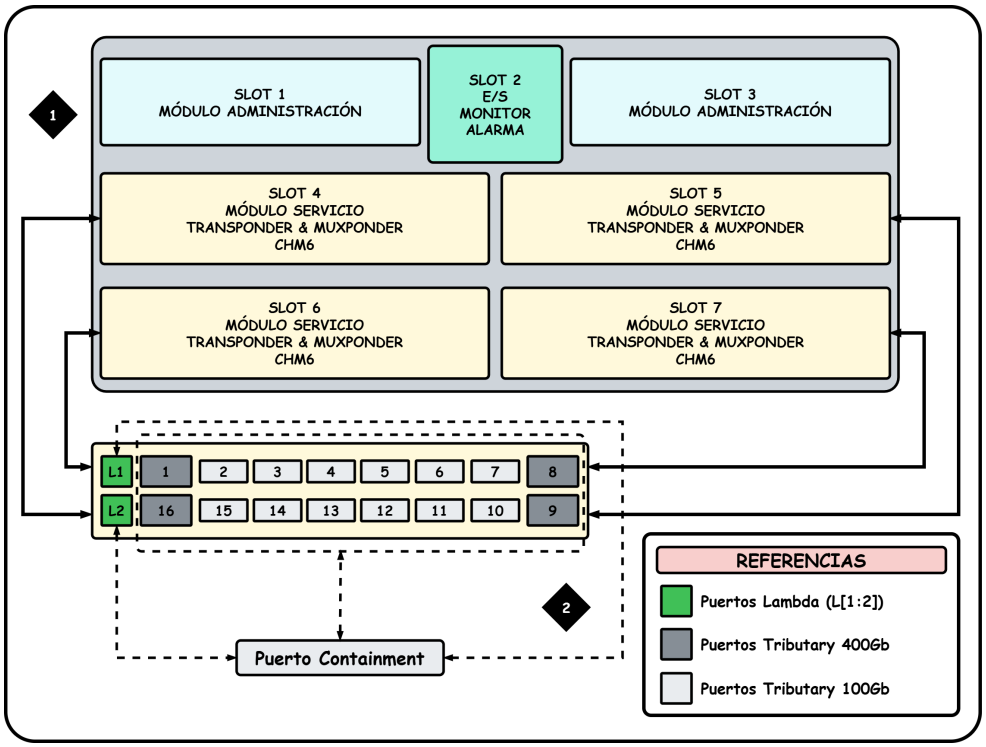

Figura 3.3: Esquema de un chasis G42 con sus siete Slots, incluyendo los módulo de administración y monitor de alarma, mostrando en detalle en una placa CHM6 los tipos de puertos: Lambda, Tributary y el puerto lógico Containment. Figura basada en Figura 1, sección 1.3, G42 Chassis [2].

FLEX: ROADM y OLA

Los nodos FLEX constituyen el dominio fotónico de tránsito del Lightpath. En este dominio, los ROADM realizan conmutación óptica de longitudes de onda (sin conversión O-E-O) y los OLA compensan la atenuación acumulada en los tramos de fibra y elementos pasivos, manteniendo la potencia dentro de rangos operativos y preservando la calidad de transmisión.

En la interfaz entre un GX y un ROADM se utiliza una placa Operational Switch (OPSW), ubicada entre el puerto Line de la CHM6 y el ROADM (ver indicación 1, Figura 3.4). Esta placa permite implementar protección 1+1: desde cada puerto Line se derivan dos salidas ópticas (L1A y L1B) que se conectan a placas FlexROADM Switching Module (FSM) diferentes del ROADM. Bajo este esquema, la señal se duplica y circula simultáneamente por ambos trayectos, habilitando conmutación ante fallas sin conversión O-E-O, reduciendo el tiempo de recuperación y asegurando la continuidad del servicio.

La Figura 3.4 resume la estructura funcional de un nodo ROADM. En el bloque CDC, la placa FSM (indicación 2) habilita la distribución y el encaminamiento óptico mediante la funcionalidad Multicast Switch, permitiendo duplicar y distribuir señales ópticas que llegan al nodo. Sobre esa base, el ROADM realiza tres operaciones sobre un Lightpath: Add/Drop y Express. Add inserta una longitud de onda proveniente de un Transponder hacia la fibra de línea, Drop extrae una longitud de onda desde la fibra y la dirige hacia un Transponder, y Express permite que la longitud de onda atraviese el nodo sin ser modificada ni convertida, lo cual ocurre típicamente en nodos intermedios de tránsito. En estos casos, el Lightpath se mantiene transparente extremo a extremo (sin conversión O-E-O), lo que permite cursar múltiples longitudes de onda con baja latencia y menor complejidad en los nodos de paso. Finalmente, la señal egresa hacia el siguiente tramo por el puerto Line asociado al bloque WSS (indicación 5), responsable de seleccionar qué longitudes de onda se dirigen a cada salida.

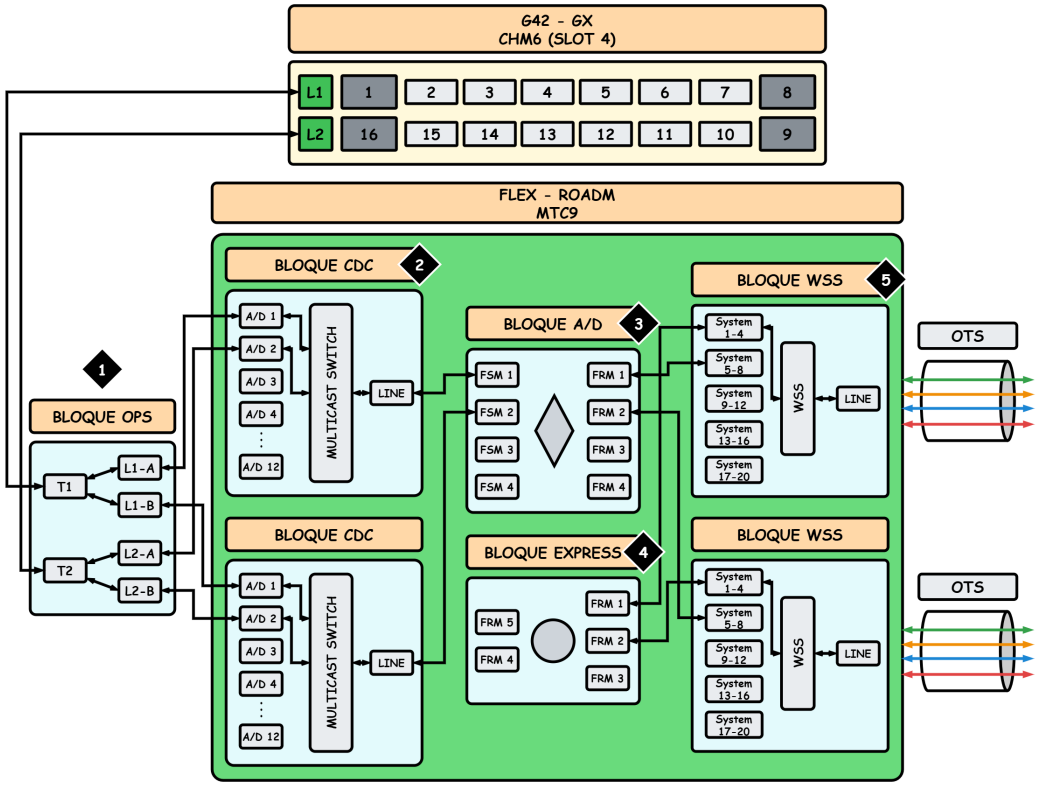

Figura 3.4: Topología lógica de un nodo tipo A, mostrando una placa CHM6 conectada al bloque OPS, enlace hacia el ROADM a través del bloque CDC, seguido del bloque A/D, el WSS y las lambdas transmitidas por la fibra, incluyendo también el bloque Express.

Ejemplo (recorrido físico de un servicio a través de la infraestructura GX y FLEX). El siguiente ejemplo ilustra el recorrido de un servicio de cliente entre dos nodos adyacentes, Node A y Node B, sin nodos intermedios (ver indicaciones 1, 2, 4 y 5 en la Figura 3.5).

1. **Ingreso del cliente:** El cliente se conecta al Tributary 11 de la CHM6 en el Slot 4 del GX-A.
2. **Asociación interna:** El puerto Containment vincula este puerto al Lambda 1 de la misma CHM6.
3. **Emisión por Line:** La señal sale por el puerto Line 1 y viaja hacia el dominio FLEX (ROADM A).
4. **Tránsito óptico:** El Lightpath recorre el tramo de fibra que conecta FLEX-A con FLEX-B.
5. **Terminación:** En FLEX-B, el Lightpath ingresa al extremo remoto y se entrega al Tributary 10 del Slot 4.

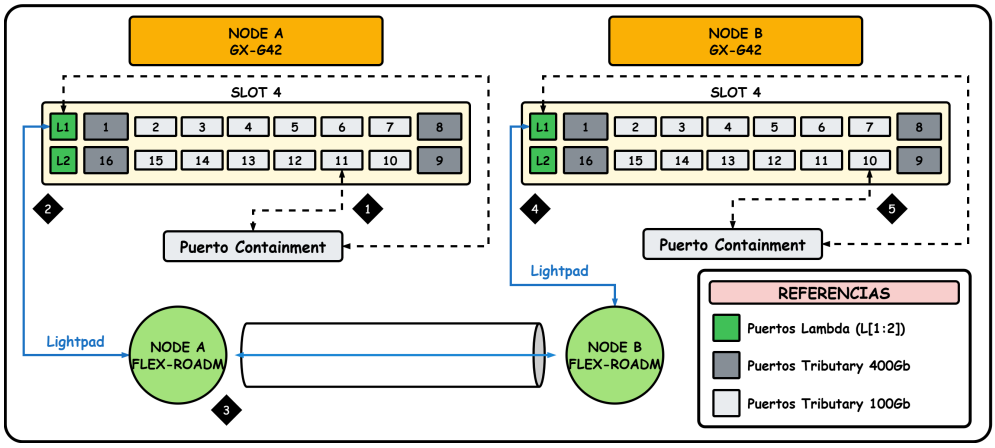

Figura 3.5: Recorrido de un servicio desde una placa CHM6 de origen hasta la placa CHM6 de destino.

3.3.2. Abstracción de la Topología

La subsección anterior describió el dominio desde el hardware, los puertos y la interconexión física. Sin embargo, para operar la red e integrarla con aplicaciones, ese mismo dominio se expone mediante abstracciones T-API que reorganizan la información en objetos lógicos navegables. En esta sección se presentan tres de esas abstracciones y su relación entre sí.

En primer lugar, el `topology-context` describe la red como un grafo de Node, NEP y Link, proporcionando una vista estructural independiente del detalle interno de cada equipo. Sobre esa topología, el `connectivity-context` modela capacidades de transporte mediante Connection, cuyos extremos se expresan como Connection End Point (CEP) soportados jerárquicamente por NEP. Finalmente, los SIP permiten ubicar los puntos de inicio y fin de servicio desde la perspectiva de provisión, vinculando el acceso al servicio con la topología.

Los ejemplos incluidos en las Figuras 3.6 a 3.9 se utilizan como apoyo para fijar estas relaciones. Primero se muestra cómo se conectan Node, NEP y Link. Luego cómo una Connection se apoya en CEP y su jerarquía respecto a NEP. Por último cómo un servicio extremo a extremo entre nodos GX se construye sobre estas abstracciones, atravesando nodos intermedios del dominio FLEX. A partir de ese mismo ejemplo, la Figura 3.10 resume cómo el servicio se organiza en capas OTN desde la señal cliente hasta el canal óptico, lo que resulta clave para interpretar los objetos de conectividad y vincularlos posteriormente con métricas de desempeño.

`topology-context`

En el `topology-context` la red se representa como un grafo lógico compuesto por Node, NEP y Link. Un Node modela un nodo de la red desde la perspectiva de topología (por ejemplo, un equipo o un bloque funcional), mientras que un NEP representa un punto de terminación dentro de ese Node utilizado para materializar conectividad. Los Link describen adyacencias entre Node y se conectan a través de NEP: cada Link tiene exactamente un NEP en cada extremo, y cada NEP pertenece a un único Node.

Esta abstracción permite navegar la estructura de la red y razonar sobre conectividad sin depender del detalle interno de cada equipo. En particular, aunque a nivel físico existan dos fibras (una por sentido), en la vista de topología la relación entre nodos se expresa como un único Link bidireccional.

Ejemplo (relación Node-NEP-Link). El siguiente ejemplo fija la relación Node-NEP-Link. Si dos nodos están conectados en la topología, el Link se apoya en un NEP de cada extremo (ver Figura 3.6). En particular, para Node A-FLEX y Node B-FLEX sin nodos intermedios, se cumple que:

1. Node A posee un NEP que referencia como propio.
2. Ese NEP se relaciona de forma única con un Link.
3. En el otro extremo del Link existe otro NEP, distinto del anterior, referenciado como propio por Node B.

La relación es: Node A $\leftrightarrow$ NEP de A $\leftrightarrow$ Link $\leftrightarrow$ NEP de B $\leftrightarrow$ Node B.

`connectivity-context`

El `connectivity-context` se define sobre el `topology-context` y describe la conectividad habilitada en la red como capacidades de transporte. Su entidad principal es la Connection, que representa una capacidad entre Node (o dentro del mismo Node). Toda Connection comienza y termina en extremos abstractos denominados CEP.

En esta vista, los extremos CEP se soportan jerárquicamente sobre NEP de la topología (ver Figura 3.7). La relación NEP-CEP se expresa mediante dos roles:

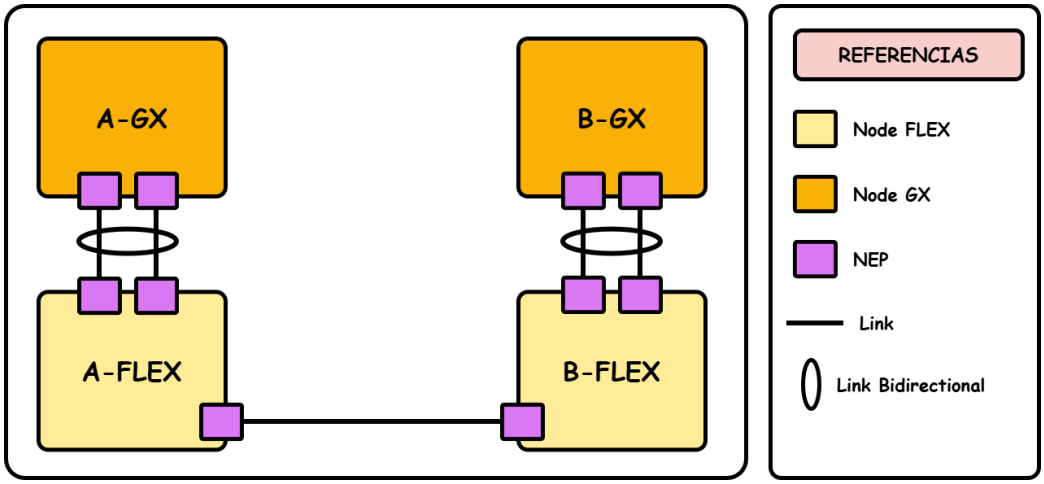

Figura 3.6: Relación en la perspectiva del `topology-context`, mostrando un Node A-GX conectado a un Node A-FLEX, luego mediante un Link hacia un Node B-FLEX y finalmente a un Node B-GX, incluyendo la visualización de Node, NEP y Link.

- **Client-NEP**: el NEP soporta un CEP cuya Connection está en la misma capa.
- **Parent-NEP**: el NEP es referenciado por un CEP cuya Connection pertenece a una capa inferior.

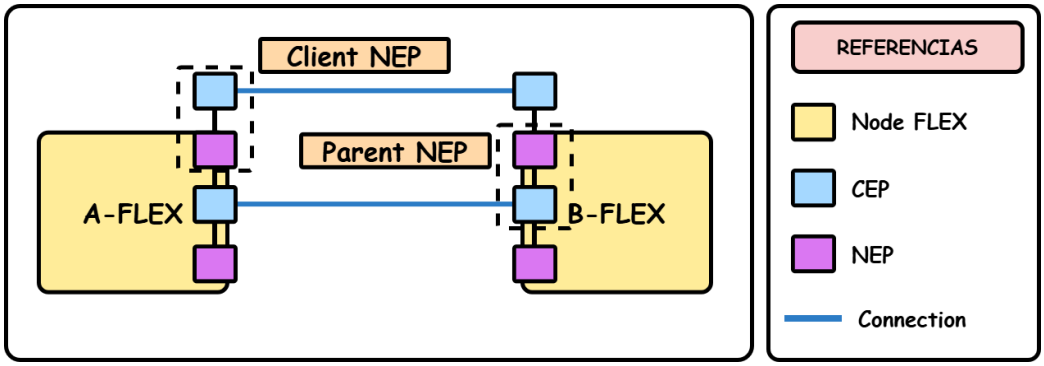

Figura 3.7: Relación entre Node A-FLEX y Node B-FLEX desde la perspectiva del `connectivity-context`, mostrando NEP y CEP, y la asociación Client-NEP y Parent-NEP.

Según la porción de red representada, una Connection puede clasificarse como:

- **PATH:** nivel jerárquico superior. Representa el Lightpath extremo a extremo (Node A ↔ Node B), con posibles nodos intermedios.
- **Link Connection Server Path (LC-SP):** sub-conexión de un PATH entre ROADM↔ROADM, independientemente de OLA intermedios.

- **Link Connection Physical Trail (LC-PT):** sub-conexión entre NE adyacentes (ROADM↔ROADM, OLA↔ROADM, G42↔ROADM); en el dominio óptico, corresponde a tramos físicos de fibra.
- **Cross-Connection (CC):** Cross-Connection. Conectividad intra Node (inicia y termina en el mismo Node).

En consecuencia, un PATH se materializa mediante CC en cada Node de tránsito y por LC-SP/LC-PT según la porción de red atravesada.

Ejemplo (relación Connection-CEP soportada por NEP). El siguiente ejemplo fija cómo la conectividad se expresa como una Connection entre extremos CEP, soportados por NEP del `topology-context`. Para Node A-FLEX y Node B-FLEX sin nodos intermedios (ver Figura 3.8):

1. Node A posee un NEP propio.
2. Ese NEP soporta el extremo CEP A de la Connection A↔B.
3. La Connection une CEP A y CEP B.
4. El extremo CEP B se soporta en un NEP propio de Node B.

La relación queda: Node A ↔ NEP A ↔ CEP A ↔ Connection ⟺ CEP B ↔ NEP B ↔ Node B.

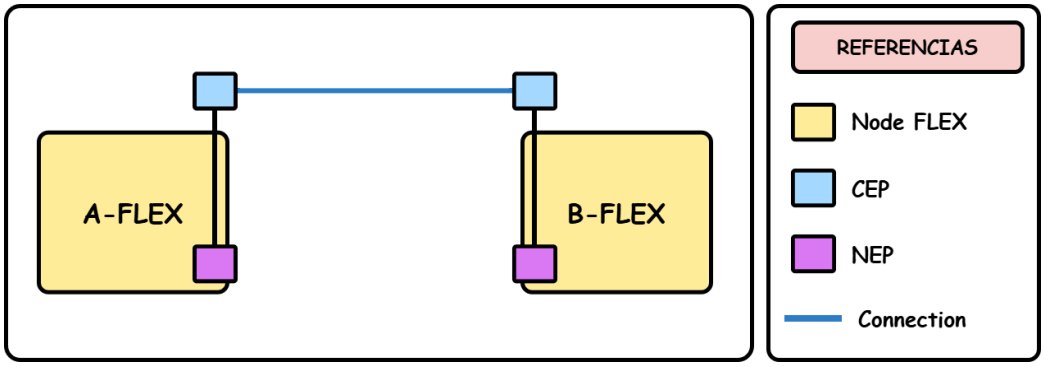

Figura 3.8: Relación entre Node FLEX a través de una Connection desde la perspectiva del `connectivity-context`, mostrando NEP y CEP.

`service-interface-point`

En `topology-context`, un servicio comienza o termina en un SIP, que se relaciona uno a uno con un NEP. Este NEP pertenece a un Node y es de tipo Access-NEP, lo que indica que su función es la de ingreso o egreso del Node. En cambio, los NEP que soportan Link no son considerados puntos de acceso a servicio.

El cambio entre `topology-context` y `connectivity-context` es crucial. En `topology-context`, las relaciones de conectividad se dan entre nodos y enlaces,

mientras que en `connectivity-context`, las relaciones se trasladan a un nivel superior, iniciando o terminando en CEP soportados jerárquicamente por los NEP.

El SIP está asociado a cada puerto Tributary (Access-Port en `equipment-context`) de una placa CHM6 en un G42, y se vincula con un Access-NEP (abstracción del Access-Port en `equipment-context`). De esta forma, cada uno de los 16 puertos Tributary en una CHM6 tiene su Access-NEP y SIP correspondiente.

Ejemplo (relación `service-interface-point` y Access-NEP). El siguiente ejemplo muestra cómo el servicio se inicia en un SIP asociado a un puerto Tributary de un G42, el cual está vinculado a un Access-NEP en el `topology-context`. Dados Node A-GX y Node D-GX, con nodos intermedios (Node A-GX ↔ Node A-FLEX ↔ Node B-FLEX ↔ Node C-FLEX ↔ Node D-FLEX ↔ Node D-GX):

Suposiciones:

- El cliente requiere 100 *Gb* desde Node A hasta Node D y es el único conectado a la placa.
- En Node A-GX, se conecta un QSFP28 en el puerto Tributary 5 del Slot 4 (ver indicación 3, Figura 3.9).
- Por simplicidad, se asume que tanto en A como en D hay un único Shelf.
- En Node D-GX, la salida es también por el Slot 4, puerto Tributary 5 con QSFP28.

El proceso de provisión de servicio sigue estos pasos:

1. El SIP asociado al puerto Tributary 5 en Node A-GX inicia el servicio, vinculado a su Access-NEP correspondiente en el `topology-context`.
2. El puerto Containment establece la conexión jerárquica con las capas de transporte Digital Signal Rate (DSR), ODU, y PHOTONIC MEDIA, encapsulando la señal desde el nivel digital hasta el óptico, lo que da lugar a un Lightpath.
3. La señal viaja por la red, atravesando diferentes nodos intermedios, hasta llegar a Node D-GX, donde se entrega al puerto Tributary 5 en el Slot 4, a través de su respectivo SIP.

Este ejemplo ilustra la relación entre los elementos de la red física, el SIP y el Access-NEP, mostrando cómo el servicio fluye desde un Node de origen hasta su destino final.

- PATH extremo a extremo entre el puerto Tributary de A-GX y el puerto Tributary de D-GX (ver indicación 3, Figura 3.9).
- PATH entre puertos Line de A-FLEX y D-FLEX (ver indicación 3 y 4, Figura 3.9), con:
    - CC en cada Node del PATH.

    - LC-SP entre ROADM↔ROADM: A-FLEX ↔ C-FLEX y C-FLEX ↔ D-FLEX.
    - LC-PT entre cada par de Node adyacentes: A-FLEX ↔ B-FLEX, B-FLEX ↔ C-FLEX, C-FLEX ↔ D-FLEX (ver indicación 2, Figura 3.9).
- PATH extremo a extremo entre el puerto Containment de A-GX y el puerto Containment de D-GX (ver indicación 5, Figura 3.9).
- PATH extremo a extremo entre puertos Line de A-GX y D-GX (ver indicación 6, Figura 3.9).

Con esto, se ha establecido el Lightpath y se ha identificado qué Node de la topología recorre, junto con las conexiones (Connection) internas que lo soportan.

Representación de un Servicio en OTN

En el ejemplo anterior, hemos identificado cómo se configura un servicio en términos de SIP, Node y Connection. Sin embargo, para una comprensión completa del comportamiento del servicio en la red, es necesario conocer cómo la señal cliente se encapsula en las capas del modelo OTN y cómo se transporta a través del dominio fotónico. A continuación, presentamos un ejemplo que ilustra este proceso de mapeo, desde la terminación del cliente hasta la portadora óptica, abarcando las entidades de multiplexación en el espectro.

Ejemplo (mapeo de capas en un servicio A-GX↔D-GX) Este ejemplo busca ilustrar cómo se distribuyen las capas DSR, ODU, PHOTONIC MEDIA y sus transformaciones asociadas en un servicio real. Consideramos un servicio de 100 Gb/s entre Node A y Node D (ver Figura 3.10), donde el cliente es el único conectado a la placa.

Suposiciones:

- El cliente requiere 100 Gb de ancho de banda desde Node A hasta Node D, siendo el único cliente en esa placa.
- En Node A-GX, se conecta un módulo QSFP28 en el puerto Tributary 5 del Slot 4 (ver indicación 3, Figura 3.9).
- Por simplicidad, en ambos nodos (A y D) hay un único Shelf.
- En Node D-GX, la salida es también por el Slot 4, puerto Tributary 5 con QSFP28.

En Node A-GX, la gestión de los puertos de la CHM6 en el Slot 4 es manejada por el puerto Containment. En una primera fase, se establece una Connection extremo a extremo entre los puertos Containment en la capa DSR (ver indicación 1, Figura 3.10), lo que define la capacidad de línea, en este caso, 100 Gb/s.

Luego, en la capa ODU, el puerto se encarga de organizar y controlar los recursos de transporte disponibles. Esta función se realiza mediante la asignación de

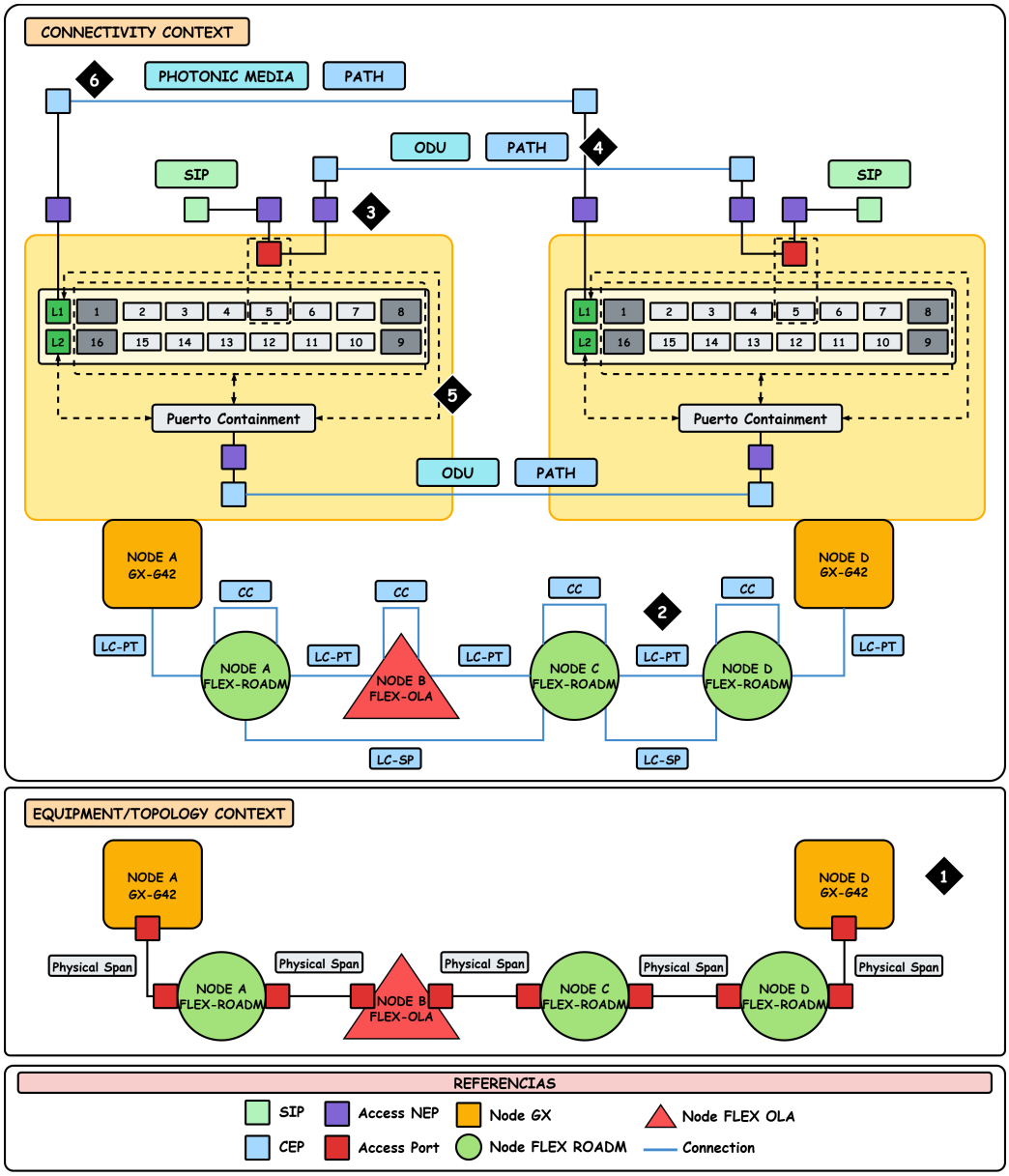

Figura 3.9: Servicio entre los Node A-GX y Node D-GX, evidenciando cómo el SIP asociado al puerto Tributary de A-GX se vincula con el Access-NEP correspondiente. En `connectivity-context` y `topology-context` se muestran las relaciones de los elementos, y en `connectivity-context` se detalla la conexión desde el SIP asociado a A-GX hasta el SIP en D-GX.

ODU de acuerdo con la recomendación ITU-T G.709 [21]. Los ODU se encapsulan en OTU, sobre los cuales se añaden encabezados de transmisión y mecanismos de corrección de errores (FEC), conforme a la estructura definida por OTN. En esta fase, existe una Connection de la capa ODU entre los puertos Tributary (ver indicación 2, Figura 3.10).

A continuación, las transformaciones se realizan en la capa PHOTONIC MEDIA. Cada OTU se transporta en el dominio óptico a través de una Optical Tributary Signal (OTSi). Un conjunto de OTSi puede agruparse para formar una Optical Tributary Signal Group (OTSiG), que representa la entidad de transporte en esta capa.

Para su transmisión, un OTSiG requiere una portadora óptica llamada OCH o Lightpath (ver indicación 3, Figura 3.10), definida por parámetros como frecuencia central y ancho de banda espectral asignado. Un conjunto de OCH puede tratarse como una única entidad por los filtros de la red, lo que se conoce como supercanal (Super-Channel (SCH)). Los supercanales (SCH) y las portadoras ópticas (OCH) multiplexadas en el espectro forman una OMS (ver indicación 4, Figura 3.10). Finalmente, esta OMS se transporta sobre OTS (ver indicación 5, Figura 3.10), que soporta la propagación de la señal a través de la fibra óptica.

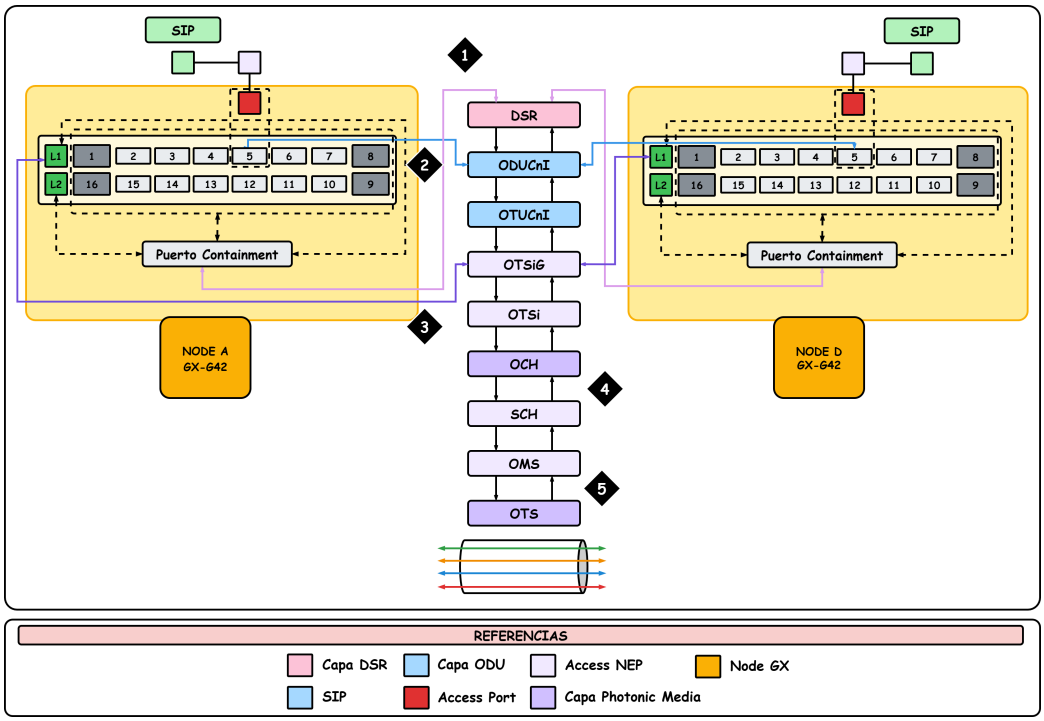

Figura 3.10: Capas de un servicio entre los Node A-GX y Node D-GX y su correspondencia con la estructura OTN, mostrando las placas de los Node A y Node D y destacando las relaciones entre los puertos de dichas placas y cada una de las capas: DSR, Optical Digital Unit-CnI (ODUCnI), Optical Channel Transport Unit-Cn (OTUCnI), OTSiG, OTSi, OCH, SCH, OMS y OTS.

3.4. Resumen y Conclusiones

En este capítulo se presentó un análisis detallado de la red óptica de ANTEL, basado en el manual TRNBI de Infinera y los archivos `common-context` y

`PM Data` alineados con T-API 2.1.3. A través de las vistas de `equipment-context`, `topology-context` y `connectivity-context`, se organizó la información y se modelaron los recursos y servicios de la red.

Se describió la infraestructura física de la red, que incluye los equipos GX (Transponders y Muxponders) y FLEX (ROADM y OLA), y se explicó su relación con la topología abstracta. Se ilustró el recorrido de un servicio desde los puertos Tributary hasta las capas fotónicas de la red, mostrando cómo se encapsulan y transportan las señales.

En la parte de conectividad, se detallaron las relaciones entre Node, NEP y Link, así como las conexiones Connection que permiten el transporte de servicios en la red. También se describió la función de los `service-interface-points` (SIP) y su vinculación con los Access-NEP.

Finalmente, se explicó cómo las señales se encapsulan y transportan a través de las capas de OTN, desde la capa DSR hasta las portadoras ópticas (OCH, OTSiG, OMS, OTS), completando el mapeo de capas necesario para interpretar el transporte de servicios en la red.

Este análisis permite una comprensión clara de cómo se modelan los recursos y servicios en la red óptica de ANTEL, proporcionando las bases para la gestión y planificación de los recursos en el proyecto. En el siguiente capítulo, se definirá el modelo de datos para apoyar esta gestión.

Capítulo 4

Modelado de Datos

En este capítulo se describe el proceso y las decisiones adoptadas para el modelado de los datos de la red de fibra óptica de ANTEL. El trabajo parte de las fuentes disponibles (coordenadas geográficas, `common-context` y `PM Data`) y de las especificaciones formales de los modelos YANG de T-API y de sus extensiones vendor-specific de Infinera. A partir de estos insumos se identifican las entidades y relaciones relevantes del dominio y se determina qué información está efectivamente expuesta por la red en los archivos de monitoreo.

Sobre esa base se construye un modelo de datos general organizado por niveles, que permite representar de forma coherente la topología física, los lightpaths y los servicios. A partir de este modelo se definen tres modelos concretos con fines complementarios: un modelo relacional de alcance amplio (Modelo A), un modelo derivado y optimizado para visualización y consulta operativa (Modelo B) y un modelo documental orientado a métricas de desempeño (Modelo de PMs). Finalmente, se analizan cuantitativamente estos modelos, con énfasis en el Modelo A, su correspondencia con el `common-context` y el grado de utilización de estructuras T-API/Infinera en la red observada, así como en la cobertura efectiva de métricas en el Modelo de PMs.

Los objetivos del modelado de datos son:

- Construir un modelo de datos que represente la red, incluyendo topología física, lightpaths y servicios.
- Facilitar el estudio de la red mediante una estructura mnemotécnica y didáctica, que favorezca la navegabilidad entre vistas y niveles de abstracción.
- Dar soporte a la herramienta de visualización y al cálculo de indicadores de desempeño, proporcionando modelos adecuados tanto para consultas estructuradas como para el análisis de métricas de PM.

El capítulo se organiza de la siguiente manera. En primer lugar, la Sección 4.1 presenta el estudio del problema, describe las fuentes de datos disponibles y analiza los modelos YANG de T-API junto con las extensiones de Infinera que sirven de base para el diseño. Luego, la Sección 4.2 introduce el modelo de datos general, su

organización por niveles y los tres modelos derivados definidos para fines complementarios (Modelo A, Modelo B y Modelo de PMs). A continuación, la Sección 4.3 aborda el análisis del modelo general, con foco en el Modelo A y en el Modelo de PMs , evaluando su cobertura y alcance respecto de las fuentes disponibles. Por último, la Sección 4.4 resume los principales resultados del capítulo y presenta las conclusiones.

4.1. Estudio del Problema

En esta sección se abordan los aspectos clave del problema que se analiza, comenzando con la identificación y descripción de las fuentes de datos disponibles, y luego profundizando en el estudio de los modelos YANG utilizados para estructurar y representar los datos de la red.

En la Subsección 4.1.1 se discute sobre las fuentes de datos disponibles, detallando los archivos JSON que contienen información de la red (coordenadas geográficas de los dispositivos de la red (NE), el archivo `common-context` y los archivos `PM Data`). En esta subsección se describen los diferentes tipos de archivos, su frecuencia de actualización y la granularidad de los datos.

Por otro lado, en la Subsección 4.1.2 se centra el estudio de los modelos YANG de T-API e Infinera, fundamentales para la representación de la red. Se explica cómo T-API establece un marco común para el monitoreo de redes de transporte y cómo Infinera extiende y adapta estos modelos para su propia implementación. Además, se detallan las dependencias entre los modelos YANG utilizados por T-API y las especificaciones del fabricante, Infinera, así como las estructuras clave que permiten la interoperabilidad y la representación de los diferentes subcontextos dentro del modelo de la red.

4.1.1. Fuentes de Datos Disponibles

El análisis del problema se inicia con la identificación de las fuentes de datos utilizadas para la creación del modelo. Las fuentes disponibles son tres tipos de archivos en formato JSON: las coordenadas geográficas de los dispositivos de la red (NE), el `common-context` y los archivos `PM Data`. A continuación, se detallan las características de cada una de estas fuentes:

- **Coordenadas Geográficas**: Se dispone de un único archivo JSON que indica la ubicación geográfica de los NE de la red. Este archivo es actualizado manualmente cada vez que se agrega un nuevo dispositivo.

- **Archivo** `common-context`: Este archivo contiene 529 registros recopilados desde el 21 de enero de 2024 hasta el 6 de agosto de 2025, con una periodicidad diaria. El archivo captura diferentes subcontextos y entidades relacionadas con la red, tales como `service-interface-point`, `topology-context`, `connectivity-context`, y `equipment-context`.

- **Archivo** `PM Data`: Compuesto por 61 archivos que cubren el período entre el 18 de marzo de 2025 y el 4 de agosto de 2025, con periodicidad diaria. Este archivo contiene parámetros detallados sobre diversas entidades, como Access-Port, Connection, Device y Equipment, con granularidad de reporte configurable a 15 minutos o 24 horas.

Cada archivo aporta información sobre la red desde diferentes niveles de abstracción, lo que implica la necesidad de tratarlos de manera separada. El archivo de coordenadas geográficas es único y de actualización manual, mientras que los archivos `common-context` y `PM Data` presentan una periodicidad de captura diaria, pero con diferentes granularidades para los parámetros.

4.1.2. Estudio de Modelos YANG de T-API e Infinera

Los archivos `common-context` están basados en modelos YANG, los cuales proporcionan un marco estándar para el monitoreo de redes de transporte. El objetivo principal de T-API es unificar la vista lógica y operativa de la red, facilitando la interoperabilidad entre equipos de distintos fabricantes. Aunque T-API establece una base común para la definición de modelos, Infinera adapta estas especificaciones para crear sus propios modelos YANG personalizados, lo que se conoce como vendor-specific o especificaciones del fabricante (ver las definiciones de Infinera en Figura 4.1).

En la indicación 1 de la Figura 4.1, se muestra cómo T-API define el modelo `common-context` en el archivo `tapi-common`. Infinera importa todas las estructuras definidas en `tapi-common`, así como otras definiciones clave de T-API como `tapi-photonic-media`, `tapi-connectivity` y `tapi-topology`. Con todas estas estructuras importadas, se define el `common-context` propio, denominado `infn-tapi-common`. Aunque la organización de los modelos de Infinera difiere de la estructura estándar propuesta por T-API, es evidente que su desarrollo tiene como base sus definiciones,

En la indicación 2 de la Figura 4.1, T-API también define varios subcontextos dentro del `common-context`, tales como `tapi-connectivity`, `tapi-equipment` y `tapi-topology`. Cada uno de estos subcontextos aborda aspectos específicos de la red:

- `tapi-connectivity` modela la conectividad de la red, e incluye Connection y `connectivity-services`. Estos elementos permiten representar los caminos de extremo a extremo en la red, sin necesidad de conocer la topología exacta. Por ejemplo, para conectar un servicio entre el punto A y el punto B, se utiliza una Connection. Este archivo también incluye toda la información relacionada con una Connection, como los CEP y las jerarquías entre ellos.

- `tapi-topology` describe las estructuras que conforman la `topology-context`, es decir, la topología de la red. Aquí se encuentran las entidades Node, NEP y Link, y sus características asociadas.

- `tapi-equipment` define las estructuras que componen la `equipment-context`, modelando los elementos físicos de la red. Estas estructuras incluyen Devices, Access-Port y Physical Span, así como la jerarquía dentro de un Device, Equipment y Container Holder.

En la indicación 3 de la Figura 4.1, se observa cómo se modelan las capas de un servicio. Estas capas están representadas por los modelos `tapi-dsr`, `tapi-odu` y `tapi-photonic-media`, los cuales definen las conexiones que componen un servicio, especificando las características de cada capa.

Además de importar los modelos de T-API y adaptar los suyos propios, Infinera incorpora otros modelos adicionales. En la indicación 4 de la Figura 4.1, se muestra cómo el modelo `infn-tapi-equipment` importa el modelo `ietf-inet-type`, que ha sido definido por Internet Engineering Task Force (IETF). Este modelo tiene como objetivo estandarizar la representación de direcciones Internet Protocol version 4 (IPv4), Internet Protocol version 6 (IPv6), y los números de puerto, lo cual facilita la configuración de los NE de la red óptica a través de una red IP mediante el plano de control y gestión.

El modelo `infn-nrp-interface` importa otros modelos adicionales, como `nrp-interface` y `mef-type`, tal como se muestra en la indicación 5 de la Figura 4.1. Con ello, Infinera aprovecha las definiciones estandarizadas del Metro Ethernet Forum (MEF) para describir interfaces de red y tipos básicos, asegurando la compatibilidad con los estándares de redes Metro y de Transporte.

Como se mencionó en el Capítulo 3, la Connection, que representa un servicio en su máxima abstracción, está compuesta por diferentes Connection, y estas Connection pertenecen a determinadas capas. Estas capas son DSR, ODU y PHOTONIC MEDIA. Mientras que T-API define tres modelos separados para estas capas (ver indicación 3, Figura 4.1), Infinera agrupa `tapi-dsr` y `tapi-odu` en un único modelo llamado `infn-tapi-digital`, mientras que en otro modelo, `infn-tapi-photonic-media`, importa `tapi-photonic-media`. De esta forma, en lugar de tener tres modelos por separado, Infinera opta por crear dos modelos bien definidos, separando el servicio a nivel digital y óptico, siguiendo el esquema Optical Transport Hierarchy (OTH) de OTN.

Finalmente, en la indicación 7 de la Figura 4.1, se puede ver cómo Infinera define los modelos `infn-tapi-topology` y `infn-tapi-photonic-media`, los cuales parten de las bases definidas por T-API.

Infinera, además de importar los modelos de T-API e IETF, realiza modificaciones y adiciones propias en los modelos. Estas modificaciones están sujetas a la visión del fabricante e incluyen desde cambios de nombres hasta la creación de nuevas estructuras que permiten definir relaciones específicas entre los elementos de la red. Por ejemplo, `tapi-topology` define las estructuras Node y Link, las cuales son importadas por Infinera en su modelo `infn-tapi-equipment`. Es en este modelo donde se definen nuevas estructuras como `supporting-device` y `supporting-physical-span`, las cuales permiten vincular directamente los elementos del `topology-context` con el `equipment-context`.

Existen otras estructuras definidas en `tapi-topology` que Infinera importa en su modelo `infn-tapi-topology`, como `mapped-service-interface-point` (ma-

pped-sip), que permite relacionar un SIP con un NEP cuyo Node lo referencia como propio. Esta vinculación facilita la relación entre el topology-context y los servicios dentro de la red.

Además de adoptar estas estructuras, Infinera realiza otras modificaciones, como la adición de la estructura protection-role en infn-tapi-common. Esta adición se hace sobre la estructura route del modelo tapi-connectivity, y permite identificar el rol de protección de cada ruta, como si es activa (working o nominal), de respaldo (protection o emergencia) u otra ruta de soporte, facilitando así la monitorización y la resiliencia de la red.

4.2. Modelo de Datos General

El modelo de datos desarrollado se basa en los modelos YANG de T-API, que estructuran los datos de manera definida y consistente. Al igual que en Infinera, los datos se organizan en un esquema claro, donde cada atributo tiene un tipo específico. La estructura del modelo se organiza en tres niveles, basados en la periodicidad de las fuentes de archivos JSON, que presentan diferentes temporalidades: coordenadas, common-context y PM Data. Estos niveles se clasifican como bajo, medio y alto.

- **Nivel bajo:** incluye entidades con atributos estáticos, como la posición de los Node en la red, extraída de un archivo JSON cuya estructura difiere de la de common-context.
- **Nivel medio:** comprende las entidades provenientes de common-context.
- **Nivel alto:** engloba las entidades provenientes de PM Data.

El modelo de datos general se presenta mediante tres modelos complementarios, cada uno enfocado en aspectos específicos de la red y con objetivos distintos:

- **Modelo A:** agrupa los datos provenientes de common-context e integra los niveles bajo y medio. Su objetivo es ofrecer una representación amplia de las entidades y relaciones de la red, capturando su evolución temporal, y operar como base canónica para el análisis. Dado que busca cubrir la mayor parte posible del common-context, el modelo incluye un número elevado de tablas y el volumen de datos crece con el tiempo, lo que dificulta la obtención de información específica para uso operativo. Por este motivo, una vez construido el Modelo A se evaluó la complejidad de acceso a la información requerida por ANTEL. En particular, se analizaron los Key Performance Indicator (KPI) de interés, que pueden estar representados directamente en el modelo como entidades o atributos, o bien derivarse a partir de cálculos sobre atributos distribuidos en múltiples entidades. Esta evaluación motivó el diseño del Modelo B, definido como fuente principal de consulta para el visualizador.

- **Modelo B:** se centra en los últimos seis meses de las instancias contenidas en el Modelo A. Optimiza el acceso a los datos mediante su consolidación en nuevas estructuras (tablas) que reducen operaciones y aceleran consultas desde el visualizador. Su objetivo es ofrecer una visión temporalmente restringida de la red y presentar tablas con datos procesados y transformados, facilitando la visualización y el análisis operativo.

- **Modelo de PMs:** está orientado al análisis de rendimiento y a la medición del desempeño de la red. Integra métricas de PMs y permite analizar tendencias, alertas y agregaciones a gran escala. Actúa como complemento, aportando información de desempeño que se integra con las estructuras del Modelo B.

Estos tres modelos trabajan de manera complementaria. El Modelo A representa la visión más detallada y exhaustiva de la red, mientras que el Modelo B ofrece una versión simplificada y optimizada para el análisis operativo. El Modelo de PMs, por su parte, se enfoca en métricas de desempeño y permite profundizar el análisis del estado de la red.

A lo largo de este capítulo se describen en detalle los tres modelos. En la Subsección 4.2.1 se presenta el Modelo A, en la Subsección 4.2.2 el Modelo B, y en la Subsección 4.2.3 el Modelo de PMs.

4.2.1. Modelo de Datos A

El Modelo A sigue un enfoque relacional basado en los modelos YANG de T-API, adoptado por Infinera para estructurar los datos de su red. La decisión de utilizar un modelo relacional está alineada con las prácticas comunes en la industria, como se destaca en el trabajo de Huyen [16], que sugiere comenzar con modelos relacionales y migrar a modelos NoSQL solo cuando los datos o consultas se vuelven demasiado complejos. Este modelo tiene como objetivo capturar la mayor cantidad posible de la información contenida en el archivo `common-context` para cada instancia de la red.

El archivo `common-context` se organiza en subcontextos modulares, cada uno enfocado en un aspecto específico de la red. Para modelar los datos extraídos de estos subcontextos, se clasifican las entidades en tres categorías: centrales, características y de vinculación.

- **Entidades centrales**: Son los elementos clave que representan la red, como Service-Interface-Point, Node,Node-Edge-Point, Link, Connection, Device, Access-Port y Physical Span. Estas entidades centrales pueden tener atributos comunes a varias entidades o atributos específicos de cada una.

- **Entidades características**: Proporcionan información adicional sobre las entidades centrales y se vinculan a ellas. Estas entidades siguen el concepto de herencia presente en los modelos YANG de T-API, lo que significa que las entidades características dependen de las centrales.

- **Entidades de vinculación**: Establecen relaciones entre las entidades centrales, lo que facilita la navegación dentro del modelo y mejora su eficiencia. Estas entidades permiten interconectar las entidades centrales, reduciendo la cantidad de operaciones necesarias en etapas posteriores.

La Figura 4.2 muestra cómo se estructuran y se relacionan las entidades centrales, características y de vinculación dentro del modelo, destacando su interdependencia.

Este modelo incluye 52 tablas. En el nivel bajo, se encuentra una única entidad central. En el nivel medio, el modelo incluye 13 entidades centrales, 6 entidades de vinculación y 32 entidades características (ver Figuras A.1 y A.2). La entidad central que tiene más relaciones con las entidades características es el SIP (ver indicación 1, Figura A.1), que representa el elemento de mayor abstracción en la red.

Existen otras 52 tablas, indicadas en rojo en las figuras correspondientes, que no están implementadas en el modelo actual. Estas tablas están relacionadas con las entidades ya implementadas, en caso de que se desee incorporarlas en el futuro. La razón por la cual no están implementadas se debe a que, aunque forman parte de los modelos T-API e Infinera, no se encuentran en los archivos JSON de monitoreo de la red de ANTEL. Esto puede deberse a que esas funcionalidades no están habilitadas en los equipos de la red o a que Infinera aún no las ha implementado en su hardware.

Un ejemplo claro de esto es la tabla `connectivity-services` (ver indicación 2, Figura A.4), que modela la estructura `connectivity-services` definida en el modelo `tapi-connectivity`. Esta estructura no está implementada en la red de ANTEL, pero sería de gran valor para la construcción de servicios, ya que Infinera la utiliza como la abstracción máxima de un servicio. Si esta estructura estuviera presente en la red, no sería necesario realizar el análisis de Subconnection, lo que simplificaría la tarea al eliminar la necesidad de vincular elementos entre diferentes contextos, como `topology-context` y `connectivity-context`. Así, el análisis solo se haría cuando fuera necesario conocer una característica específica de un camino en una visión concreta de la red.

4.2.2. Modelo de Datos B

El Modelo B tiene como objetivo principal dar soporte a la aplicación de visualización. Este modelo contiene los últimos seis meses de datos extraídos del Modelo A, junto con un conjunto de 19 entidades centrales derivadas de dicho modelo. Las entidades se agrupan en tres grandes categorías: tipo `all_<nombre>`, hardware y topología.

- Tipo `all_<nombre>`: Estas entidades tienen el propósito de agrupar todos los elementos de un tipo específico. Un ejemplo es la tabla `all_equipment`, que contiene todos los registros de Equipment correspondientes a la fecha más reciente de carga del Modelo A. Esta tarea es necesaria debido a que en el Modelo A la información a veces se encuentra dispersa entre diversas tablas.

- Hardware: Este grupo incluye las entidades utilizadas para representar el hardware de la red. Por ejemplo, para un NE GX o FLEX, se selecciona un ROADM y se pueden acceder a las placas que contiene, así como a los puertos de cada una de esas placas. Además, de cada puerto se pueden consultar características tales como el estado administrativo y operacional (ver indicación 1, Figura B.1).
- Topología: Esta categoría tiene como objetivo simplificar las estructuras de datos para poder construir los diferentes grafos de la red. El enfoque para los grafos es el siguiente: en primer lugar, se construye el grafo físico, luego el grafo de Lightpath y finalmente el grafo de servicios (ver indicación 2, Figura B.1).

Para la topología física, es necesario contar con las entidades `all_eots`, que corresponden a los OTS (en la nomenclatura de ANTEL, EOTS), para poder establecer la relación (NE - EOTS). Para el grafo de Lightpath o OCH (EOCH en la nomenclatura de ANTEL), es necesario disponer de las entidades `all_eoch`, `all_eoch_info` y `all_eoch_ceps_neps`. Esto permite implementar el grafo de Lightpath sobre el grafo físico y también obtener características adicionales sobre estas topologías.

Finalmente, para implementar el grafo de servicios, es necesario disponer de entidades como `all_service_T_to_T` y `all_service_L_to_L`. La primera entidad permite representar el cliente en una placa de entrada a la red, es decir, un puerto Tributary sobre una placa CHM6 de un equipo G42.

La segunda entidad implementa las relaciones de los puertos Line entre las placas CHM6, las cuales están vinculadas a un servicio. Estas entidades contienen la información de las Connection, que permiten construir el camino completo de un servicio, desde el momento en que el cliente ingresa a la red hasta que egresa de ella.todo el camino de un servicio, desde que el cliente ingresa a la red hasta que egresa.

4.2.3. Modelo de Datos PM

Para modelar los datos de PMs, se consideraron dos enfoques ampliamente utilizados: el modelo relacional y el modelo documental. El modelo relacional es generalmente la primera opción debido a su estructura bien definida y la capacidad de establecer relaciones directas entre los diferentes elementos de la red. Sin embargo, el modelo documental, aunque menos estructurado, ofrece mayor flexibilidad y capacidad expresiva, lo que lo hace adecuado cuando se trabaja con grandes volúmenes de datos y estructuras dinámicas.

Uno de los principales desafíos al modelar los PMs es la variabilidad en los parámetros de monitoreo (PM). La cantidad de parámetros de monitoreo (PIs) puede variar tanto entre diferentes grupos de PIGs como dentro de las mismas entidades de la red. Por ejemplo, se ha observado que dos Devices o Access-Ports pueden tener un número distinto de PIs para un mismo PIG. A pesar de consultar varios documentos oficiales de Infinera [19], no se pudo determinar con certeza

las causas de estas diferencias ni obtener un listado completo de las agrupaciones disponibles por tipo de entidad, lo que limita el alcance del análisis.

El modelo relacional presenta dificultades en este contexto debido a la necesidad de definir estructuras estáticas, lo cual no es viable ante la variabilidad observada. Por otro lado, el modelo documental permite manejar estructuras más flexibles con un número variable de parámetros, lo que lo hace más adecuado para este tipo de datos. Los modelos documentales, como MongoDB, almacenan los datos en un formato similar a JSON, lo que permite guardar directamente los PMs y realizar transformaciones simples sobre los datos. Además, el modelo documental facilita la adaptación a cambios en la estructura de los datos, lo que no requiere la modificación de un esquema estructurado como en el caso del modelo relacional.

Con base en estos criterios, se optó por implementar un modelo documental utilizando MongoDB, un motor de base de datos ampliamente adoptado en entornos de datos no estructurados. MongoDB es capaz de gestionar colecciones con millones de documentos de forma eficiente y ofrece capacidades de indexación que permiten el procesamiento de grandes volúmenes de datos y la gestión de registros duplicados. A partir de los documentos `PM Data` proporcionados por ANTEL, se definió una estructura para la representación de los registros de PMs.

A continuación, se presenta la tabla 4.1, que enumera los parámetros seleccionados para estructurar los registros en la base de datos. Cada parámetro va acompañado de una breve descripción, destacando su relevancia dentro del modelo. Al comparar esta selección con la estructura base de los PMs presentada en la Sección 3.2.2, se identificaron dos campos que fueron descartados: `pmpGranularity` y `retrievalTime`.

- `pmpGranularity`: Se omite, ya que la periodicidad fija de 15 minutos para la captura de todos los PIGs no aporta información adicional.
- `retrievalTime`: Indica el momento de extracción del archivo `PM Data`, pero no ofrece valor analítico, ya que el campo `periodEndTime` ya define el fin del intervalo de medición.

A partir de estas decisiones y considerando la descripción funcional de cada campo, se definió una estructura depurada que mantiene el formato original y realiza ajustes mínimos para lograr una visualización más clara, ordenada y útil de los datos.

4.3. Análisis del Modelo General

Una vez definido el modelo de datos general, se dispone de una visión consistente de los datos y de sus relaciones. A partir de ello, el objetivo de esta sección es analizar el alcance y la utilidad práctica de los modelos construidos, priorizando aquellos aspectos que condicionan el uso posterior en el pipeline y en la visualización.

El análisis se centra en el Modelo A y en el Modelo PM, ya que el Modelo B conserva las mismas entidades y relaciones que el Modelo A, pero restringidas a una

Tabla 4.1: Estructura definida para registros PIG del Modelo de PMs.

Parámetro	**Descripción**
`timestamp`	Fecha de captura de la medición. Se toma `periodEndTime` como instante de referencia (fin de la ventana de observación). Permite construir series temporales y usarlo como criterio de filtrado.
`device_uuid` / `equipment_uuid` / `access_port_uuid` / `connection_uuid`	Identificadores de entidades extraídos y normalizados desde `pmpMonitoredObject`. Vinculan cada PIG con sus entidades correspondientes en el modelo relacional.
`name` / `name_info`	Nombre del PIG. Se separa en `name` (solo el nombre del PIG) y `name_info` (datos de la entidad asociada) para clarificar la estructura y permitir filtrado sin duplicados por `name`.
`location`	Ubicación de la medición (valor de `pmpLocation`); aporta contexto para interpretar las mediciones.
`direction`	Dirección de la medición (valor de `pmpDirection`); complementa ubicación y nombre del grupo para identificar con precisión el punto de medición.
`monitoredTime`	Duración del período de recopilación; intervalo temporal sobre el cual se calculan/promedian las métricas. Es clave para su correcta interpretación.
`value` (lista de PI)	Estructura interna de cada PIG: `value` contiene la lista de PI y sus metadatos (`name`, `value`, `indicatorUnit`, `status`).

ventana temporal. En consecuencia, las conclusiones sobre estructura y recorridos obtenidas para el Modelo A son transferibles al Modelo B, por lo que no se lo analiza en profundidad de forma independiente.

En la Subsección 4.3.1 se presenta el análisis del Modelo A a partir de su navegabilidad vertical y horizontal, clave para relacionar elementos y acceder a la información con la menor cantidad de operaciones posibles. Además, se cuantifica la correspondencia entre lo disponible en el `common-context` y lo capturado por el parseo del sistema, y se estima el grado de utilización de las estructuras de T-API e Infinera en la red observada. Luego, en la Subsección 4.3.2 se analiza el Modelo de PMs, relevando qué métricas de desempeño están efectivamente presentes en los archivos `PM Data`, qué parámetros incluyen y su cobertura sobre los equipos, con el fin de delimitar la información disponible para la construcción de la herramienta de visualización.

4.3.1. Análisis de Modelo A

En esta sección se analiza el Modelo A desde tres perspectivas complementarias. En primer lugar, se evalúa la navegabilidad del esquema relacional propuesto, es decir, en qué medida la estructura del modelo permite recorrer la red de forma eficiente y coherente, tanto siguiendo jerarquías (`top-down`/`bottom-up`) como explorando relaciones entre entidades (por ejemplo, desde servicios hacia topología y conectividad).

Además de la navegabilidad, resulta de interés conocer cuánta información de las fuentes disponibles queda efectivamente representada en el Modelo A. Por ello, en segundo lugar se cuantifica la proporción de rutas del `common-context` que el sistema logra extraer y persistir, utilizando como referencia la instancia más reciente disponible. Este análisis estima el grado de correspondencia entre lo que entrega la red en el monitoreo y lo que el Modelo A alcanza a modelar.

En tercer lugar, y con un enfoque más general, se analiza qué fracción del universo definido por T-API y las extensiones de Infinera aparece materializada en el `common-context`. Esta comparación permite identificar qué partes del estándar están presentes en la red observada y cuáles no, y aporta contexto para interpretar tanto el alcance del Modelo A como el del Modelo B.

Navegabilidad del Modelo

La estructura relacional definida para el Modelo A busca facilitar la exploración de la red mediante relaciones explícitas entre entidades. Este enfoque permite acceder a la información desde distintos puntos de entrada (por ejemplo, un servicio, un nodo o una conexión) y recorrer el grafo de relaciones de forma directa, lo cual resulta clave para el análisis y la operación.

La navegabilidad se evalúa en dos dimensiones: vertical y horizontal. La navegabilidad vertical refiere a la posibilidad de recorrer jerarquías y dependencias entre entidades, tanto en sentido `top-down` como `bottom-up`, es decir, desde un nivel general hacia los detalles específicos y viceversa. En el modelo, esta propiedad se refleja principalmente en los vínculos que conectan las visiones de `topology-context` y `equipment-context`, permitiendo relacionar nodos, dispositivos, puertos y puntos de terminación.

Por su parte, la navegabilidad horizontal evalúa la capacidad de seguir relaciones laterales a lo largo de la infraestructura, es decir, avanzar entre elementos equivalentes en la red (por ejemplo, entre nodos conectados) sin perder el vínculo con servicios y conectividad. En la Figura 4.3 se ilustra este recorrido: un Service-Interface-Point se vincula con un NEP (`owned-node-edge-point`) mediante la entidad `mapped-sip` (indicación 1), y ese NEP pertenece a un Node dentro de `topology-context`. A partir de allí, la relación Link-NEP permite transitar entre Node y Link para recorrer la topología (indicación 3).

En `connectivity-context` (indicación 4, Figura 4.3), los Node se vinculan con las Connection a través de la relación `connection-end-point-connection`.

A su vez, se preserva la integración entre vistas mediante relaciones de soporte, como `supporting-access-port` y `supporting-devices`, que conectan Access-Port con NEP y Device con Node, respectivamente (indicación 2). En conjunto, estas asociaciones permiten pasar de la perspectiva de conectividad a la topológica y viceversa.

Además, el modelo fue diseñado para admitir extensiones futuras. En particular, si se requiere incorporar nuevos parámetros de la red, es posible agregar entidades características asociadas a entidades centrales existentes, o bien introducir nuevas entidades centrales y conectarlas a la estructura mediante entidades de vinculación (ver Figura 4.4). Este criterio resulta coherente con la evolución esperable del `common-context` a medida que la red habilite estructuras adicionales dentro del marco extendido por Infinera sobre T-API.

Dado que la topología física es un insumo clave para la operación (servicios y lightpaths dependen de ella), se analiza en detalle la navegabilidad horizontal en el vínculo entre servicios, conectividad y topología. En la Figura 4.5 se muestra que un SIP puede mapearse, mediante `mapped-sip`, hacia un NEP de `topology-context` asociado a un Node GX (indicación 1). A su vez, dicho NEP se relaciona con un CEP, que constituye un extremo de una Connection. La entidad `connection-end-point-connection` permite entonces recorrer la red a través de conexiones, habilitando la navegación horizontal entre nodos GX sin requerir conocer de antemano los elementos intermedios internos (nodos FLEX).

Complementariamente, la indicación 2 de la Figura 4.5 ilustra el vínculo entre el `service-interface-point` y la visión de `topology-context`, permitiendo identificar los Node que componen el trayecto de un servicio dentro de la red. De este modo, el Modelo A integra las visiones de `connectivity` y `topology` a partir de un mismo elemento de entrada (SIP), facilitando análisis orientados a operación y diagnóstico.

Metodología de Comparación por Rutas

El diseño del Modelo A se basa en los modelos YANG de T-API y en las extensiones provistas por Infinera. En esta sección se apartado se busca cuantificar dos aspectos complementarios: qué proporción de la información efectivamente entregada por el `common-context` es capturada por el Modelo A (a través del parseo implementado), y qué fracción del universo de estructuras definido por T-API e Infinera aparece materializada en los archivos `common-context`.

Para ello, se adopta un enfoque cuantitativo basado en el conteo de rutas (paths) dentro de estructuras jerárquicas. En este análisis, una ruta se entiende como la secuencia ordenada de nodos que permite acceder, dentro de un `context`, a un elemento del modelo (desde la raíz del `context` hasta un campo hoja (atributo) o hasta una estructura contenedora (objeto/lista) que agrupa información de la red). Bajo esta definición, el conjunto de rutas observadas funciona como una aproximación a la cobertura estructural de cada fuente.

A partir de estas rutas se comparan tres fuentes de referencia:

- **TAPI/Infinera (universo teórico):** conjunto de rutas definido por los modelos YANG de T-API y sus extensiones de Infinera.
- **Instancia `common-context` (red observada):** rutas efectivamente presentes en una captura concreta del monitoreo, tomada como referencia de la información disponible.
- **Modelo A (parseo Proyecto Final de Carrera (PFC)):** rutas que el sistema implementado es capaz de extraer y persistir para construir el Modelo A.

Para caracterizar el universo T-API/Infinera se consideran los módulos resumidos en la tabla 4.2.

Tabla 4.2: Módulos YANG del modelo T-API [28] y las especificaciones de Infinera.

Módulo YANG
`tapi-common@2020-04-23.yang`
`tapi-connectivity@2020-06-16.yang`
`tapi-equipment@2020-04-23.yang`
`tapi-path-computation@2020-04-23.yang`
`tapi-photonic-media@2020-06-16.yang`
`tapi-topology@2020-04-23.yang`
`infn-tapi-common@2024-02-22.yang`
`infn-tapi-equipment@2024-01-29.yang`
`infn-tapi-photonic-media@2023-05-04.yang`
`infn-tapi-topology@2022-09-01.yang`

A partir de estos módulos, utilizando `pyang` [9] y su plugin `tree`, se genera un árbol jerárquico que integra T-API y los augments de Infinera (Figura 4.6). El resultado se almacena en el archivo `Todo_modelo_TAPI_INFN.tree`, que reúne las rutas del modelo teórico, respetando tanto las agregaciones como las dependencias entre módulos.

Como referencia de la red observada, se selecciona la última instancia disponible del `common-context`, la cual corresponde al 6 de agosto de 2025, y se calculan sus rutas. Esta elección se realiza por criterio cronológico, dado que no se dispone de control sobre la consulta a la red ni sobre la política de generación de archivos. En consecuencia, es importante tener presente que el JSON más reciente no necesariamente puede coincidir con aquel que presenta la mayor cantidad de estructuras habilitadas. Sin embargo, a efectos de análisis a realizar se trabaja teniendo presente este concepto.

Por su parte, la cobertura del Modelo A se determina a partir del conjunto de rutas que el sistema implementado reconoce, extrae y persiste a partir del `common-context`. A este proceso se le denomina `Parseo sistema PFC`.

A partir de esto, se producen tres listados comparables de rutas: rutas del universo T-API/Infinera, rutas presentes en la instancia `common-context` seleccionada, y rutas cubiertas por el parseo que alimenta el Modelo A. La comparación se realiza sobre los `context` principales del `common-context` (`service-interface-point`, `topology-context`, `equipment-context` y `connectivity-context`) y se apoya en dos métricas: métrica A, la cual es una medida de coincidencia basada en la intersección de rutas, y la B, medida de cobertura basada en proporciones.

En el cálculo A se compara la cantidad de rutas de PFC o JSON con las rutas del JSON o T-API e Infinera, respectivamente. La fórmula del cálculo A es:

$$\text{Cálculo A} = \frac{\text{Coincidencias}}{\text{Rutas JSON}} \quad \text{o} \quad \text{Cálculo A} = \frac{\text{Coincidencias}}{\text{Rutas T-API/Infinera}} \tag{4.1}$$

Por otro lado, el cálculo B cuenta la cantidad de rutas de PFC o JSON respecto a las rutas del JSON o T-API e Infinera:

$$\text{Cálculo B} = \frac{\text{Rutas PFC}}{\text{Rutas JSON}} \quad \text{o} \quad \text{Cálculo B} = \frac{\text{Rutas JSON}}{\text{Rutas T-API/Infinera}} \tag{4.2}$$

La interpretación de estos cálculos requiere considerar que la extensión del JSON depende de qué estructuras estén habilitadas en la red y de cómo se haya obtenido la captura. En particular, pueden existir rutas presentes en el JSON que no queden cubiertas por el parseo (limitaciones del Modelo A), o rutas previstas por T-API/Infinera que no aparezcan en la instancia evaluada (funcionalidades no habilitadas u operativas en la red). Asimismo, al generar el árbol con `pyang` pueden omitirse módulos no cargados explícitamente, lo que puede introducir diferencias puntuales entre las rutas del JSON y las rutas del universo T-API/Infinera considerado.

Con este esquema, el primer aspecto se evalúa comparando las rutas del `common-context` con las rutas cubiertas por el `Parseo sistema PFC`, mientras que el segundo se cuantifica comparando las rutas del `common-context` con las rutas del universo T-API/Infinera. A continuación se presentan ambos resultados por separado: primero se analiza la cobertura del Modelo A respecto de la información disponible en el `common-context`, y luego se estima el grado en que la red materializa las estructuras definidas por T-API e Infinera.

Correspondencia del Modelo A con el `common-context`

Este apartado responde la primera pregunta del análisis: determinar en qué medida el `Parseo sistema PFC` captura la estructura efectivamente presente en el `common-context`, tomado como referencia de la red observada. Para ello se comparan, por `context` (`service-interface-point`, `topology-context`, `equipment-context` y `connectivity-context`), las rutas extraídas por el parseo con las rutas

existentes en el JSON. La Tabla 4.3 resume esta comparación mediante el número de coincidencias, las rutas exclusivas de cada fuente (Solo PFC y Solo JSON) y el total de rutas por `context`.

Tabla 4.3: Todas las rutas de cada sección, `Parseo sistema PFC` y JSON.

Context	**Coincidencias**	**Solo PFC**	**Solo JSON**	**Rutas PFC**	**Rutas JSON**
`service-interface-point`	48	3	8	51	56
`topology-context`	76	35	71	111	147
`equipment-context`	60	23	57	83	117
`connectivity-context`	46	14	27	60	73

Como se observa en la tabla 4.4, el `Parseo sistema PFC` obtiene, en el cálculo A, valores superiores al 50 % en todos los `context` evaluados, alcanzando un máximo de 85.7 % en `service-interface-point`. En términos de cobertura (cálculo B), el mayor valor también se da en `service-interface-point` (91.1 %), consistente con el criterio de diseño del Modelo A, que priorizó la representación de la red a partir de los servicios. Esto se refleja en el Modelo A (ver Apéndice A, Figura A.1), donde `service-interface-point` concentra la mayor cantidad de entidades características.

En contraste, los `context` con menor cobertura relativa son `topology-context` y `equipment-context`. Este comportamiento se explica por el orden de priorización adoptado durante el modelado y la implementación del parseo, donde se extrajeron primero las estructuras asociadas a `service-interface-point`, luego `connectivity-context`, y finalmente `topology-context` y `equipment-context`. En consecuencia, una fracción mayor de rutas presentes en estos dos últimos `context` queda fuera del alcance del `Parseo sistema PFC`.

Tabla 4.4: Porcentajes del `Parseo sistema PFC` respecto del `common-context` (instancia seleccionada).

Context	**Rutas PFC**	**Rutas JSON**	**Cálculo A**	**Cálculo B**
`service-interface-point`	51	56	85.7 %	91.1 %
`topology-context`	111	147	51.7 %	75.5 %
`equipment-context`	83	117	51.3 %	70.9 %
`connectivity-context`	60	73	63.0 %	82.2 %

Además del análisis puntual por `context`, se evalúa la evolución temporal de la cobertura del `Parseo sistema PFC` sobre el conjunto de 529 instancias disponibles del `common-context`. En este caso se aplica la misma lógica del cálculo B, pero calculada globalmente para cada archivo, de forma iterativa (instancia por instancia).

Dado que la red se encuentra en expansión desde el 23 de enero de 2024, cabría esperar una tendencia decreciente, asociada a la aparición de nuevas estructuras

que no estén contempladas en el parseo. Sin embargo, sobre el conjunto de 529 archivos se observa una variación acotada del porcentaje de parseo, del orden del 4 %, lo que se considera bajo. Este resultado sugiere que el crecimiento observado no se da principalmente por la incorporación de nuevas estructuras, sino por el aumento en la cantidad de elementos dentro de estructuras ya presentes.

Por ejemplo, una entidad característica como `service-interface-point-name` agrega campos (`value` y `value-name`) asociados a `service-interface-point`. Cuando aparece un nuevo `service-interface-point` en la red, este se incorpora como una nueva instancia dentro de la estructura ya definida, sin introducir nuevas rutas. En consecuencia, el volumen de datos crece por cardinalidad de listas/instancias, más que por expansión del árbol de rutas.

La Figura 4.7 permite visualizar este comportamiento. En los primeros 190 JSON (ordenados cronológicamente desde el 23 de enero de 2024) el porcentaje global se mantiene alto, en torno al 83 %. Alrededor del JSON número 200 se observa un pico (ver indicación 1, Figura 4.7); su impacto visual se ve acentuado por la interpolación lineal entre puntos, aunque la variación numérica respecto de instancias cercanas no supera aproximadamente el 1 %.

Al inspeccionar los valores que originan el gráfico, se observa que hasta la posición 197 existen alrededor de 410 rutas, mientras que en las posiciones 198 y 199 se registran 408 y 413 rutas, respectivamente. Estas variaciones se asocian a que los JSON no están equiespaciados temporalmente: no existe un archivo diario entre el 23 de enero de 2024 y el 6 de agosto de 2025, ni se dispone de una política conocida de habilitación de nuevas estructuras. En particular, el entorno señalado corresponde a agosto de 2024, donde se registran faltantes de archivos en el histórico:

- **Enero 2024:** fechas 24, 25, 27 y 28.
- **Julio 2024:** fechas 12, 13 y 14.
- **Agosto 2024:** fechas 16, 17 y 18.
- **Febrero 2025:** fecha 18.
- **Abril 2025:** fechas 29 y 30.
- **Mayo 2025:** fechas del 1 al 18, 20 y 21.

A partir del JSON número 250 (ver indicación 2, Figura 4.7), la cantidad de rutas se estabiliza, lo que sugiere que el conjunto de estructuras habilitadas se mantiene relativamente constante durante un tramo extendido. Hacia el final del histórico, alrededor del JSON número 500, se observa la incorporación de nuevas estructuras, reflejada en una disminución del porcentaje de parseo (ver indicación 3, Figura 4.7), consistente con la necesidad de actualizar el parseo ante cambios en el `common-context`.

En términos globales, el histórico permite evaluar la estabilidad del `Parseo sistema PFC` frente a una red en expansión. El porcentaje de parseo se mantiene en torno al 83 % a lo largo de aproximadamente un año y medio, con una pérdida

inferior al 4 %. En el mismo período, el tamaño de los archivos `common-context` aumenta de 17.5 MB (23 de enero de 2024) a 55.2 MB (6 de agosto de 2025), es decir, más del triple. Esta combinación sugiere que el crecimiento observado se manifiesta principalmente como aumento de cardinalidad dentro de estructuras ya conocidas, más que como expansión sostenida del conjunto de rutas.

A su vez, puede ocurrir que el `Parseo sistema PFC` incluya rutas que no aparecen en la instancia evaluada del JSON (ver columna "Solo PFC" en la tabla 4.3). Esto se debe a que el parseo fue implementado tomando como referencia los modelos T-API e Infinera, mientras que el `common-context` observado puede no tener habilitadas, o no reportar, determinadas funcionalidades en el momento de la captura.

Grado de Utilización de T-API e Infinera en `common-context`

Para estimar qué proporción del universo definido por T-API e Infinera se utiliza efectivamente en la práctica, se comparan las rutas presentes en la instancia seleccionada de `common-context` con las rutas definidas por los modelos YANG (obtenidas a partir del árbol generado con `pyang`). La tabla 4.5 resume, para cada `context`, la cantidad de rutas coincidentes y las rutas exclusivas de cada fuente.

Tabla 4.5: Comparación de rutas por `context`: `common-context` vs. (T-API e Infinera).

Context	Coincidencias	Solo JSON	Solo T-API-Infinera	Rutas JSON	Rutas T-API-Infinera
`service-interface-point`	56	0	94	56	150
`topology-context`	130	17	420	147	550
`equipment-context`	74	43	71	117	145
`connectivity-context`	53	20	309	73	362

A partir de estos conteos se calculan, a partir de las expresiones de calculo A y B presentadas, dos indicadores complementarios (tabla 4.6): la fracción de rutas del modelo T-API/Infinera que aparece materializada en el `common-context`, y el tamaño relativo del `common-context` frente al universo T-API/Infinera.

En `service-interface-point` ambos valores coinciden (37.3 %), ya que todas las rutas presentes en el JSON también están definidas en T-API/Infinera (columna Solo JSON igual a 0). Esto indica que, para este `context`, el `common-context` utiliza un subconjunto consistente del estándar, aunque dicho subconjunto representa poco más de un tercio del universo modelado.

En `equipment-context` se observa un comportamiento diferente: el indicador A es 51.0 %, mientras que B asciende a 80.6 %. Esta diferencia se explica por la presencia de rutas Solo JSON (43 rutas), es decir, estructuras que aparecen en el `common-context` pero no quedan reflejadas en el árbol T-API/Infinera utilizado para la comparación. En términos prácticos, esto sugiere que el monitoreo incorpora atributos adicionales (principalmente de hardware) o bien que la restricción de modelos considerada en `pyang` no captura la totalidad de esas extensiones.

Los menores niveles de utilización se obtienen en `connectivity-context` ($A = 14{,}6\,\%$) y `topology-context` ($A = 23{,}6\,\%$). En ambos casos, la cantidad de rutas Solo T-API/Infinera es muy superior a las coincidencias (tabla 4.5), lo que evidencia que el `common-context` hace uso de una porción reducida de las estructuras y campos que el estándar y las extensiones contemplan para describir conectividad y topología.

Tabla 4.6: Porcentajes de rutas del `common-context` respecto a (T-API e Infinera).

Context	**Rutas JSON**	**Rutas T-API-Infinera**	**Cálculo A**	**Cálculo B**
`service-interface-point`	56	150	37.3 %	37.3 %
`topology-context`	147	550	23.6 %	26.7 %
`equipment-context`	117	145	51.0 %	80.6 %
`connectivity-context`	73	362	14.6 %	20.1 %

4.3.2. Análisis del Modelo de PMs

A partir de la construcción del Modelo de PMs se busca determinar el alcance y la disponibilidad de PIG presentes en los archivos provistos por ANTEL. Para ello se analiza el conjunto de archivos `PM Data` entregados.

El objetivo es identificar qué PIG están efectivamente disponibles y, para cada uno, relevar los PI asociados. En paralelo, se caracteriza el vínculo entre cada PIG y los tipos de elementos de red a los que se asocia, cuantificando cuántos dispositivos aportan registros por PIG. De este modo se obtiene una visión de cobertura real de métricas en el conjunto analizado.

La metodología se implementa en Python. Se parsean los archivos `PM Data` y se estructuran sus registros conforme al modelo propuesto en la Sección 4.2.3. Con la biblioteca Pandas se generan las estructuras necesarias para el procesamiento y la visualización de resultados. A partir del total de registros se identifican los PIG presentes y, para cada uno, sus PI. Durante el proceso se observa que, para un mismo PIG, el conjunto de PI puede variar según la entidad de red sobre la que se mide. Dado que no se cuenta con evidencia suficiente para explicar esta variación, y a fin de mantener una caracterización consistente y comparable, en estos casos se toma como referencia, para cada PIG, la unión de PI observados.

Como insumo principal se elabora un documento de referencia que lista los PIG identificados y sus PI, acompañado de una breve descripción de cada parámetro. Estas descripciones se construyen combinando la consulta de fuentes oficiales de Infinera, cuando fue posible, con la inspección directa de los datos. Si bien algunas definiciones pueden requerir validación adicional, el documento constituye una base práctica para la continuidad del trabajo y para el uso operativo por parte de ANTEL.

El análisis cuantifica la cantidad de registros por archivo (ver tabla 4.7). En el conjunto analizado se identifican 42 PIG diferentes, que abarcan tanto métricas

de estado de los equipos (corriente y tensión de alimentación, ventilación, estado de almacenamiento, uso de memoria) como mediciones ópticas de potencia y parámetros vinculados a degradaciones de la red.

En este análisis se profundiza en un subconjunto de PIG de especial interés operativo para ANTEL, cuya descripción se presenta en la tabla 4.8. Posteriormente, se caracteriza su cobertura sobre la totalidad de equipos disponibles (ver tabla 4.9). Finalmente, se estima la distribución de registros por equipo mediante estadísticos simples (mínimo, mediana y máximo) para cada PIG (ver tabla 4.10), a fin de disponer de un panorama aproximado de los volúmenes de información involucrados.

Tabla 4.7: Resumen de los archivos `PM Data` analizados: número total de archivos considerados (61) y estadísticos de registros por archivo (mínimo, mediana y máximo).

Tipo de Archivo	**Archivos**	**Registros por Archivo**		
		Mínimo	**Mediana**	**Máximo**
`PM Data`	61	10143	10565	11763

Tabla 4.8: Grupos de indicadores de performance (PIG) relevantes para la operación en ANTEL, identificados en los 61 archivos `PM Data`. Para cada PIG se presenta una descripción que incluye el ámbito de aplicación y el tipo de equipo donde se registra (ROADM, OLA, GX).

PIG	**Descripción**
`OMS Optical Power Egress`	Mide la potencia óptica de salida en la sección OMS en equipos ROADM u OLA.
`OMS Optical Power Ingress`	Mide la potencia óptica de entrada en la sección OMS en equipos ROADM u OLA.
`Span Loss`	Mide la pérdida óptica del tramo entre dos nodos (sección OMS); el resultado se expresa como atenuación total del enlace. Se registra en equipos ROADM.
`Client_PMP_Ingress`	Mide la potencia óptica recibida en el puerto cliente de equipos GX.
`Client_PMP_Egress`	Mide la potencia óptica transmitida desde el puerto cliente de equipos GX.
`OCH_PMP_Egress`	Mide la potencia del canal óptico (OCH) transmitida por el puerto de línea del equipo GX.
`OCH_PMP_Ingress`	Mide parámetros del canal óptico (OCH) en el ingreso por el puerto de línea del equipo GX (p. ej., potencia óptica, OSNR, FEC, Q-factor).

Tabla 4.9: Registros por PIG y cobertura de equipos. Para cada PIG se informa el total de registros y, en la dimensión de equipos, la cantidad con datos, el total considerado y la cobertura (%). El universo analizado comprende 65 equipos tipo FLEX (ROADM, OLA) y 25 equipos tipo GX.

PIG	Total de Registros	Equipos		
		Con datos	Total	Cobertura (%)
OMS Optical Power Egress	10816	54	65	83.1 %
OMS Optical Power Ingress	10816	54	65	83.1 %
Span Loss	10816	54	65	83.1 %
Client_PMP_Egress	4121	20	25	80.0 %
Client_PMP_Ingress	4123	20	25	80.0 %
OCH_PMP_Ingress	12204	20	25	80.0 %
OCH_PMP_Egress	1356	20	25	80.0 %

Tabla 4.10: Registros y estadísticos por PIG. Para cada PIG se informa el total de registros, la cantidad de equipos que aportan datos y, sobre ese subconjunto, los estadísticos de registros por equipo (mínimo, mediana y máximo).

PIG	Total de Registros	Equipos con datos	Registros por equipo		
			Mínimo	Mediana	Máximo
OMS Optical Power Egress	10816	54	34	122	488
OMS Optical Power Ingress	10816	54	34	122	488
Span Loss	10816	54	34	122	488
Client_PMP_Egress	4121	20	52	122	771
Client_PMP_Ingress	4123	20	52	122	771
OCH_PMP_Ingress	12204	20	450	549	1098
OCH_PMP_Egress	1356	20	50	61	122

4.4. Resumen y Conclusiones

En este capítulo se describió el proceso de modelado de los datos de la red óptica de ANTEL a partir de tres fuentes con temporalidades distintas: las coordenadas geográficas de los Node de la red, el `common-context` y los archivos `PM Data`. A partir del estudio del problema y del relevamiento de los modelos YANG de T-API y de sus extensiones vendor-specific de Infinera, se identificaron las entidades y relaciones necesarias para representar la red en sus distintas vistas (`equipment-context`, `topology-context` y `connectivity-context`) y se construyó un modelo de datos general organizado en tres niveles (bajo, medio y alto).

Sobre esta base se definieron tres modelos concretos con usos complementarios. El Modelo A, relacional y de alcance amplio, se definió como base canónica del dominio. El Modelo B, derivado y optimizado a los últimos seis meses, se enriqueció con tablas agregadas y vistas simplificadas orientadas a la visualización y a la construcción de grafos de topología, lightpaths y servicios. El Modelo de PMs,

documental en MongoDB, se adaptó a la variabilidad de PIG y PI observada en los archivos `PM Data`. Finalmente, se evaluó el Modelo A mediante comparaciones sistemáticas entre las rutas YANG, el JSON `common-context` y el parseo del sistema PFC, y se analizó el Modelo de PMs sobre 61 archivos `PM Data`, identificando 42 PIG, su cobertura por tipo de equipo y un subconjunto de PIG relevantes para la operación, a partir del cual se elaboró un documento de referencia de PIG/PI para uso técnico y operativo.

Se mostró que es posible modelar la red óptica de ANTEL de forma consistente a partir de las definiciones de T-API y de las extensiones de Infinera. El estudio de los modelos YANG permitió explicitar la separación entre capas digitales y ópticas, así como la presencia de estructuras de vinculación que habilitan la navegabilidad horizontal y vertical. Estas mismas estructuras se reflejan en el diseño del Modelo A mediante la distinción entre entidades centrales, de características y de vinculación, lo que hace al modelo representativo del dominio y, al mismo tiempo, mnemotécnico y extensible.

En particular, la forma en que se categorizan las entidades y se explicitan las relaciones de vinculación no solo favorece la extensibilidad del esquema, sino que también contribuye a comprender el dominio de la red. Esta organización habilita recorridos horizontales entre entidades centrales y la incorporación incremental de características según el nivel de detalle requerido, mientras que la navegabilidad vertical permite conectar abstracciones desde la perspectiva de servicio hasta el hardware que lo soporta. Además, el orden de priorización aplicado en el parseo (primero `service-interface-point`, luego `connectivity-context` y finalmente `equipment-context` y `topology-context`) ayuda a interpretar las diferencias de cobertura observadas entre `context` y resulta consistente con el objetivo de preservar trazabilidad entre servicios, conexiones y topología para la visualización.

La validación cuantitativa indicó que el Modelo A cubre de forma sólida lo efectivamente presente en la red. Las comparaciones entre JSON y el `Parseo sistema PFC` mostraron coberturas superiores al 50 % en todos los `context`, con un máximo del 85.7 % en `service-interface-point`, coherente con el foco de diseño orientado a los servicios. La comparación entre JSON y los modelos T-API/Infinera evidenció que la red utiliza un subconjunto del universo funcional disponible, especialmente limitado en `connectivity-context`, lo que explica la presencia de tablas potenciales no implementadas por ausencia de datos en el monitoreo (por ejemplo, `connectivity-services`). El análisis histórico sobre 529 `common-context` confirmó una capacidad de parseo estable en torno al 83 %, con una caída inferior al 4 % en un año y medio, a pesar de que el tamaño de los archivos se triplicó en el período. Esto sugiere que la red creció principalmente en cantidad de elementos dentro de estructuras ya conocidas, más que en la aparición sostenida de nuevas estructuras.

En cuanto al Modelo B, este restringe el horizonte temporal, agrega información dispersa en el Modelo A y ofrece tablas específicas para hardware y topología, lo que habilita consultas eficientes alineadas con el visualizador y con los KPI de red. El Modelo de PMs, por su parte, resulta apropiado para la naturaleza de los PIG y PI y permite absorber variaciones en la estructura de los datos sin rediseñar el

esquema. El procesamiento de `PM Data` abarcó la totalidad de los registros disponibles, aunque en el capítulo se presenta un resumen centrado en los indicadores más relevantes. A partir de este relevamiento se verificó la disponibilidad de métricas de interés operativo, como potencias ópticas en línea y en puertos cliente, parámetros de calidad (p. ej., Q-factor) y pérdidas por tramo. No obstante, los volúmenes por PIG y por equipo son todavía acotados, por lo que los resultados deben interpretarse como una caracterización inicial del alcance de métricas disponibles. En este contexto, el documento de referencia de PIG/PI elaborado reduce la incertidumbre sobre la disponibilidad real de métricas y proporciona un insumo verificable para su integración en la herramienta de visualización.

Las principales limitaciones identificadas no derivan del diseño de los modelos, sino del nivel de funcionalidad actualmente expuesto por la red en los JSON de monitoreo, como la ausencia de `connectivity-services` y la disponibilidad parcial de ciertos PIG y PI. Sobre la base de los modelos de datos definidos y validados en este capítulo, el siguiente capítulo presenta el pipeline de procesamiento que implementa su uso práctico.

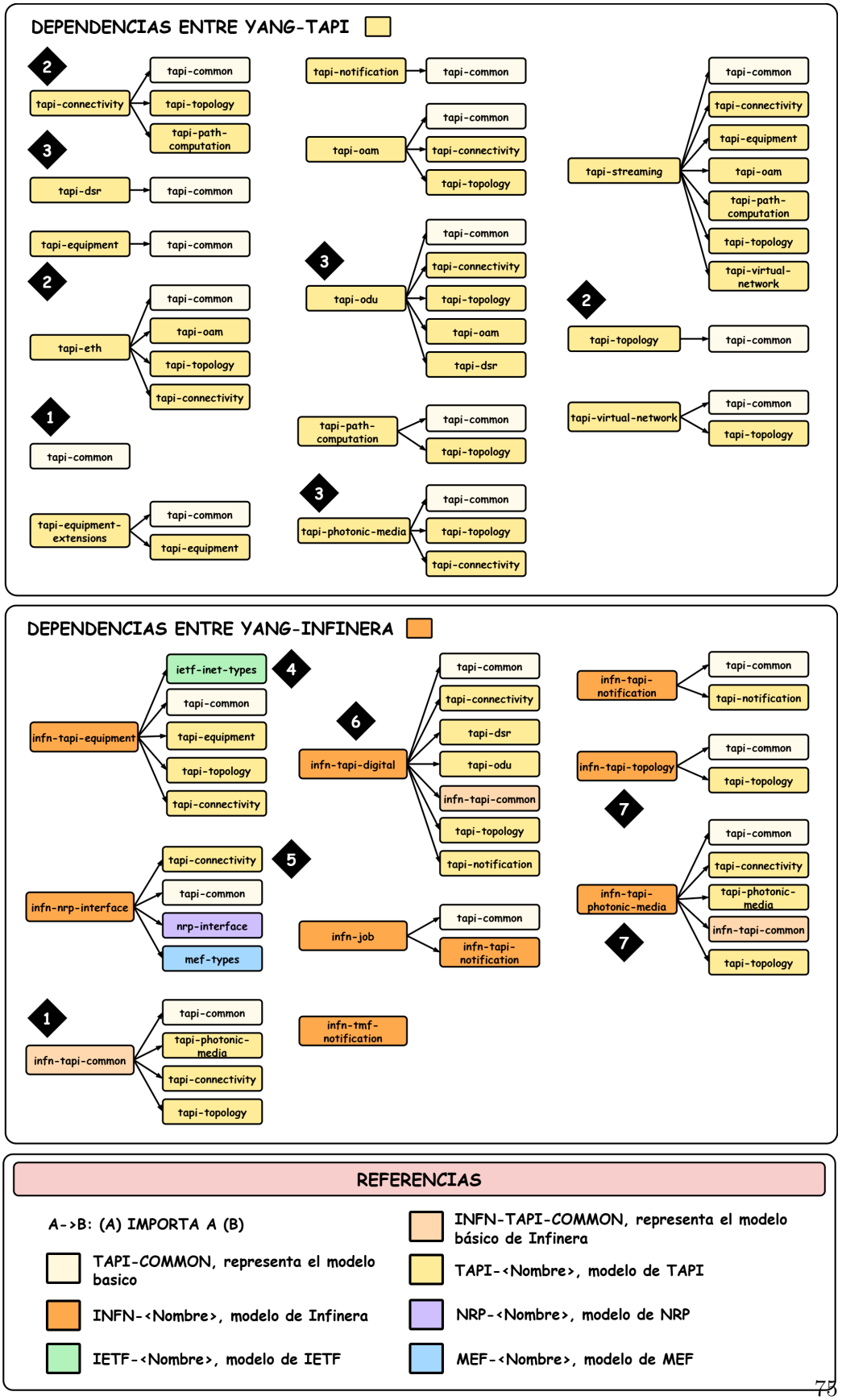

Figura 4.1: Dependencias entre los modelos YANG de T-API e Infinera, mostrando por separado las relaciones entre los módulos de cada dominio, así como las dependencias con modelos definidos por IETF y MEF.

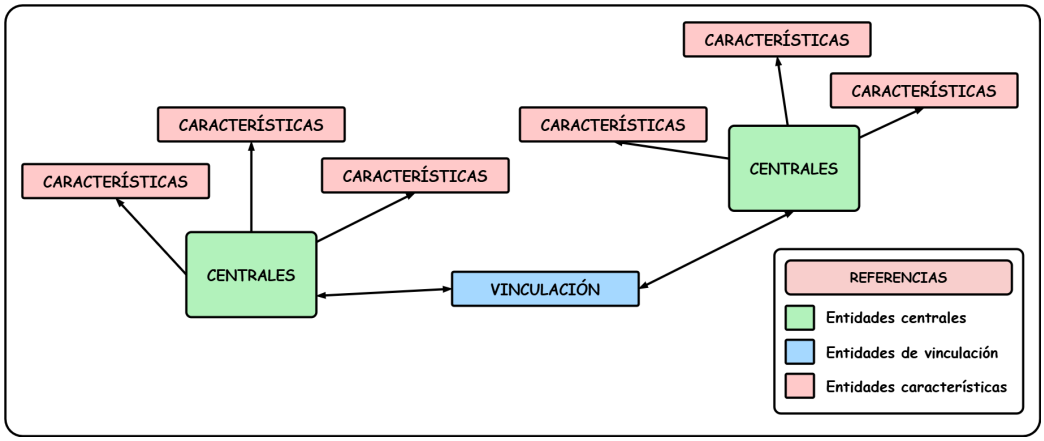

Figura 4.2: Relación entre las entidades centrales, características y de vinculación en el Modelo A, mostrando su interdependencia y el rol que cada una cumple dentro de la estructura global.

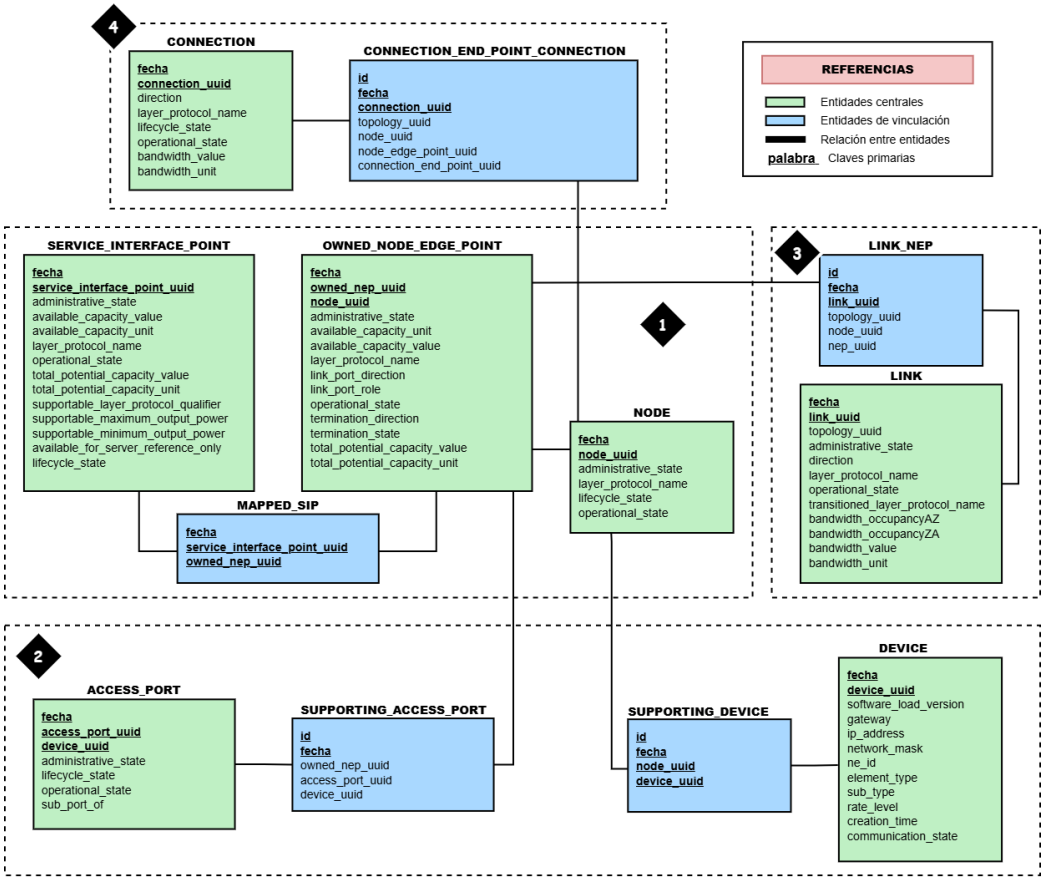

Figura 4.3: Relación entre entidades centrales y de vinculación del Modelo A. Se incluyen cuatro indicaciones que ilustran distintos tipos de relaciones: (1) el mapeo de SIP sobre Node mediante `mapped-sip`; (2) las relaciones de soporte que conectan Equipment con `topology`; (3) la interconexión entre Node a través de Link; y (4) la vinculación de Node con Connection en `connectivity-context`.

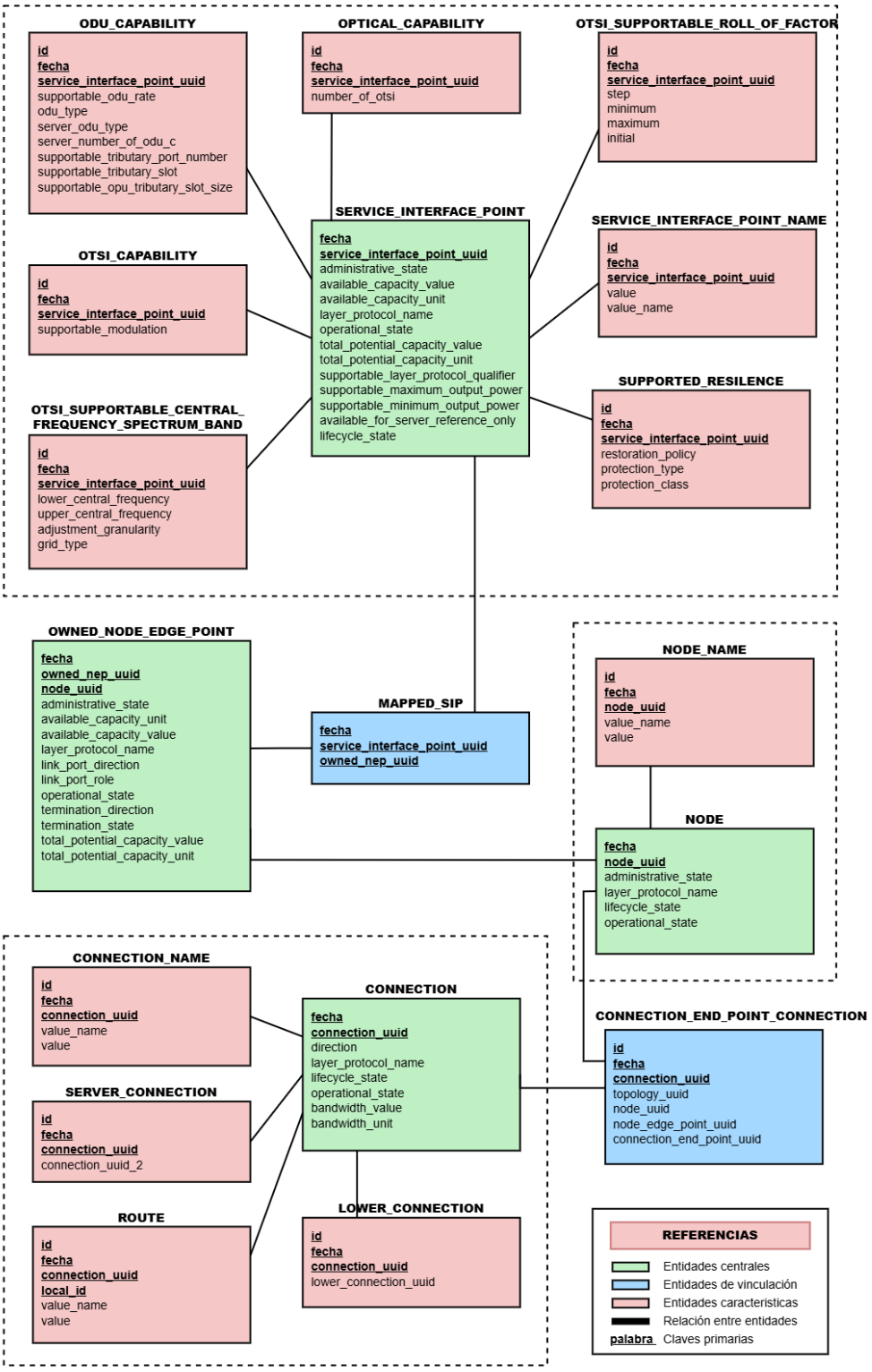

Figura 4.4: Relación entre entidades centrales y entidades características del Modelo A: (1) entidades características asociadas a SIP; (2) entidades características asociadas a Connection; y (3) entidades características asociadas a Node.

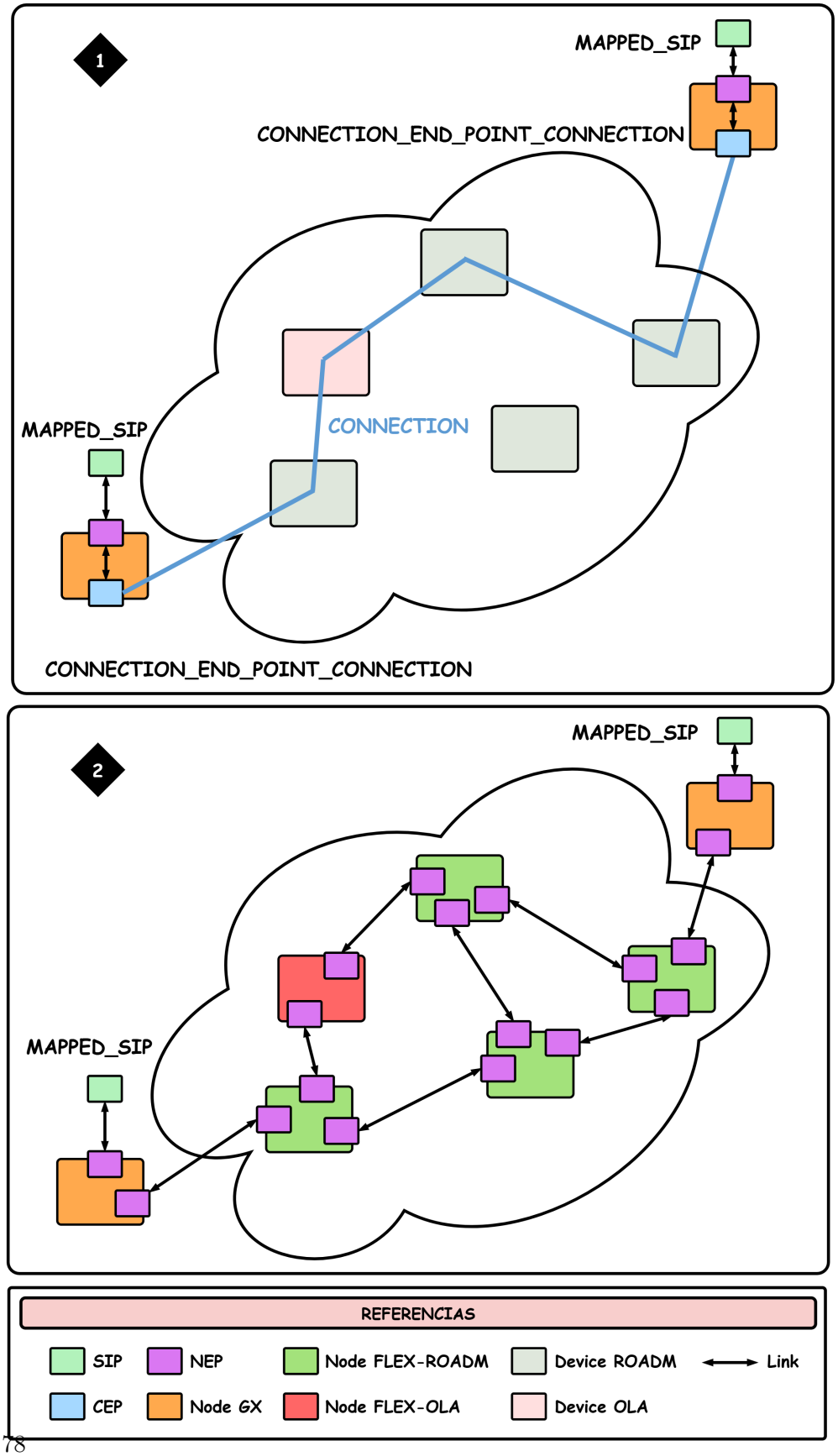

Figura 4.5: Vinculación de un SIP con `topology-context` y `connectivity-context`. En (1) se muestra el recorrido desde SIP a través de las estructuras que permiten representar la red desde la visión de `connectivity-context`; en (2) se ilustra la correspondencia con la visión `topology-context`, evidenciando el trayecto del servicio a través de la red.

```
user-pfc@pfc ~$ pyang -p . -f tree *.yang > Todo_modelo_TAPI_INFN.tree
```

Figura 4.6: Comando para generar el árbol del modelo T-API/Infinera (INFN) con `pyang`.

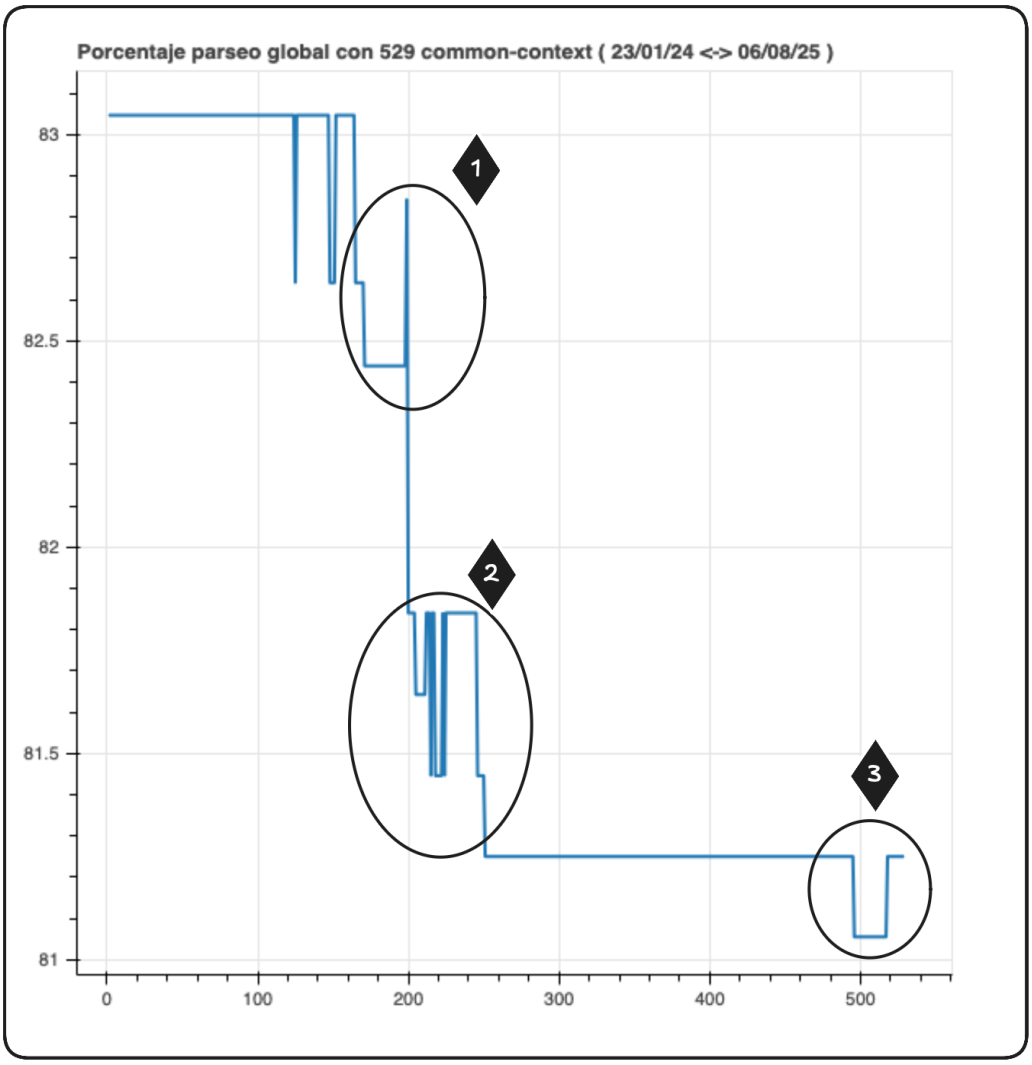

Figura 4.7: Porcentaje de parseo del sistema PFC (ordenada) frente al histórico de JSON `common-context` (abscisa, 529 entradas). El eje vertical muestra el porcentaje global de rutas del `common-context` capturadas por PFC; el eje horizontal representa el orden cronológico de los 529 archivos analizados. Se indican tres zonas de interés: (1) un pico de alta tasa de parseo, (2) un tramo de variación asociado a cambios en la cantidad de rutas entre instancias y (3) una disminución vinculada a la incorporación de nuevas estructuras no contempladas por el parseo.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Pipeline

Este capítulo presenta el pipeline de datos desarrollado para implementar los modelos definidos en el Capítulo 4. El pipeline define procesos de extracción, transformación y carga (ETL) sobre las fuentes disponibles, con el objetivo de cargar las bases de datos asociadas y asegurar consistencia, trazabilidad y unicidad de procesamiento. La solución se organiza en tres etapas encadenadas: la Etapa A ingiere `coordenadas` y `common-context` para construir el Modelo A, la Etapa B deriva el Modelo B a partir de la Base A y la Etapa C procesa `PM Data` para poblar el Modelo de PMs.

El capítulo se estructura de la siguiente manera. En la Sección 5.1 se presentan los requerimientos y limitaciones que condicionan el diseño del pipeline. Luego, en la Sección 5.2 se introduce la arquitectura adoptada, describiendo primero la estructura general del sistema y, posteriormente, la arquitectura particular de cada etapa. A continuación, en la Sección 5.3 se describe la implementación, incluyendo el stack tecnológico y la organización del código. Por último, la Sección 5.4 detalla las instancias de validación y los resultados obtenidos tras procesar los datos disponibles, y la Sección 5.5 resume los principales aportes y conclusiones del capítulo.

5.1. Parámetros de Diseño

El diseño del pipeline parte de un objetivo central: implementar, para cada modelo de datos, procesos de extracción, transformación y carga desde las fuentes disponibles hacia sus bases de datos asociadas. Además de los requisitos específicos de cada modelo (derivados de la naturaleza de las fuentes, las estructuras a representar y los esquemas de destino), existen requisitos propios del pipeline que condicionan su operación, entre ellos escalabilidad, latencia y mantenibilidad. En paralelo, se identifican limitaciones externas que condicionan decisiones de diseño, como el grado de control sobre las fuentes y la disponibilidad efectiva de los datos. En la Subsección 5.1.1 se presentan los requisitos y en la Subsección 5.1.2 las limitaciones que enmarcan la solución.

5.1.1. Requisitos

Los requisitos del pipeline se derivan directamente de los modelos de datos definidos en el Capítulo 4 y de las características de las fuentes disponibles. El objetivo principal es implementar de forma robusta la carga de los Modelos A, B y de PMs, garantizando consistencia, integridad y capacidad de crecimiento.

El pipeline debe trabajar con tres fuentes en formato JSON, con periodicidades y tamaños muy distintos: un archivo de coordenadas, cuya estructura fue definida internamente, y los archivos `common-context` y `PM Data`, ambos basados en el modelo T-API [5]. Mientras que los archivos de coordenadas manejan del orden de cientos de entradas, los `common-context` y `PM Data` contienen cientos de miles de registros por archivo. La solución debe ser capaz de extraer los datos de cada fuente, validar la estructura y el formato de los archivos para garantizar que pueden ser procesados y respetar la periodicidad de extracción por tipo de fuente, asegurando además la unicidad de procesamiento (sin reprocesar archivos ya cargados ni generar registros duplicados).

Una vez extraídos, los datos deben someterse a controles de calidad acordes a las especificaciones de los modelos de destino, transformarse a las estructuras requeridas y cargarse en las bases correspondientes. La integridad de carga es un requisito central, para cada archivo o lote, las estructuras derivadas deben insertarse de manera atómica. Ante cualquier incidencia durante la carga, el proceso no debe escribir resultados parciales y la inserción debe revertirse, sin alterar los datos previamente almacenados.

El diseño del pipeline también debe respetar el orden lógico entre modelos. El Modelo A constituye la base canónica y debe cargarse en primer lugar. El Modelo B depende directamente de los datos del Modelo A, por lo que su procesamiento sólo puede ejecutarse una vez completada la carga del Modelo A. En cuanto al Modelo de PMs, su información adquiere sentido únicamente en relación con las entidades centrales definidas en el Modelo A. Por ello, la carga de PMs debe apoyarse en la existencia previa de estas entidades y en sus identificadores.

Desde el punto de vista operativo, el pipeline debe ser escalable. Aunque en el marco del proyecto se trabaja con un volumen acotado (un archivo de `coordenadas`, 529 archivos `common-context` y 61 archivos `PM Data`), se espera un incremento sostenido. Descontando las coordenadas, cuya actualización es esporádica, se proyecta la recepción de del orden de 730 archivos por año (un `common-context` y un `PM Data` por día), cada uno con cientos de miles de entradas. El pipeline debe ser capaz de procesar este volumen sin degradar de forma significativa su operación. No se fija un tiempo máximo estricto de procesamiento por archivo, dado que se asume que las estructuras a tratar son acotadas y los archivos están bien formados, pero el diseño debe evitar cuellos de botella innecesarios.

Finalmente, la solución debe ofrecer trazabilidad completa de extremo a extremo. Debe ser posible conocer en todo momento el estado de procesamiento de cada archivo, registrar qué lotes fueron cargados, cuáles fallaron y por qué motivo, y disponer de evidencia ante incidencias. El pipeline debe contar con mecanismos de manejo de errores que permitan registrar fallos, identificar el punto de ruptura y garantizar la integridad de los datos, evitando tanto pérdidas silenciosas de

información como duplicados en las cargas.

5.1.2. Limitaciones

Es importante mencionar dos factores que condicionan el diseño de la solución: la disponibilidad de hardware y limitación en la extracción de información de la red.

En primer lugar, respecto a la disponibilidad de hardware, se trabajó enteramente en un entorno local, utilizando computadoras portátiles. La ejecución y el desempeño de la solución están, por tanto, sujetos a las especificaciones de estas máquinas, que operan simultáneamente como servidor para los motores de base de datos y como ejecutor del procesamiento del pipeline.

En cuanto a la limitación en la extracción de información de la red, no se dispone de acceso a la interfaz TRNBI [19] para realizar consultas específicas de extracción, tampoco se tiene un control sobre la cantidad de archivos extraídos. La información se recibe por lotes sin una política de adquisición preestablecida y se almacena en los directorios base. En consecuencia, la latencia y el volumen de ingreso están determinados por el proveedor de las fuentes y pueden presentar arribos irregulares y variaciones de volumen.

5.2. Arquitectura General

Esta sección describe la arquitectura lógica y las principales decisiones de diseño del pipeline de datos desarrollado. A partir de los requerimientos y las limitaciones definidos en la Sección 5.1, se adopta un esquema con tres etapas, una por modelo. Cada etapa ejecuta un proceso de extracción, transformación y carga para su modelo correspondiente. Las etapas se estructuran secuencialmente: primero se ejecuta la etapa del Modelo A, luego la del Modelo B y, por último, la del Modelo de PMs. Si bien la dependencia del Modelo de PMs es directa con la del Modelo A, se integra al proceso luego del procesamiento del Modelo B para simplificar la operación.

Para materializar este diseño, el flujo se estructura con un orquestador global que administra el ciclo de vida del proceso, registrando el inicio y la finalización de cada etapa y aplicando políticas de manejo de errores. Cada etapa, a su vez, cuenta con un orquestador propio que gestiona la conexión a la base de datos, el acceso a los directorios fuente, expone métricas de operación (latencia por etapa y volumen procesado) y notifica el progreso y alertas ante errores.

Esta segmentación organiza el flujo y facilita su interpretación. Permite aislar errores por etapa, reducir el alcance de las fallas y simplificar el mantenimiento. La orquestación ofrece un control centralizado que coordina la ejecución, preserva la trazabilidad y aplica acciones ante errores, reforzando la robustez de la solución. En lo que respecta a las fuentes de datos, para cada tipo se define un directorio que conserva el histórico de sus archivos. El consumo por etapa se detalla a continuación:

- **Etapa A (Modelo A):** procesamiento de archivos de `coordenadas` y `common-context`.
- **Etapa B (Modelo B):** procesamiento de la base de datos asociada al Modelo A (Base A).
- **Etapa C (Modelo de PMs):** procesamiento de archivos `PM Data`.

Para todas las fuentes, antes de su procesamiento, se valida su formato y estructura. En caso que la estructura y el formato sean los adecuados, los datos se incorporan al flujo. En caso contrario, se omiten y se genera una notificación. En la Figura 5.1 se presenta un diagrama simplificado de la arquitectura completa del sistema.

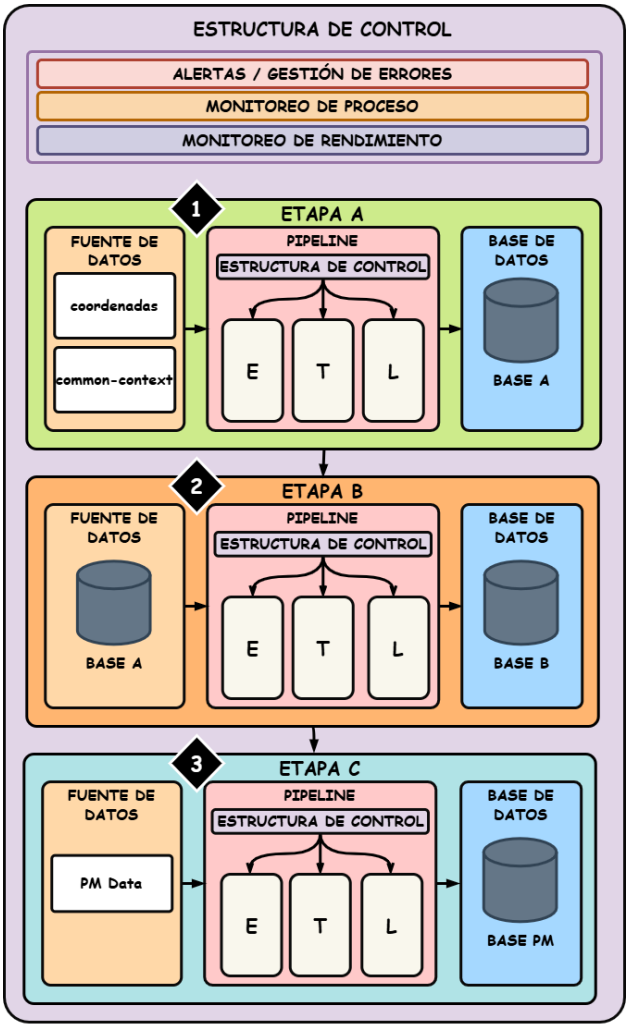

Figura 5.1: Arquitectura general del pipeline de datos. Cada etapa se representa mediante un bloque con sus fuentes, el pipeline y la base de datos asociada. La numeración indica el orden de ejecución orquestado por la estructura de control: (1) Etapa A, (2) Etapa B y (3) Etapa C.

A continuación, la Subsección 5.2.1 describe la arquitectura de la Etapa A, detallando las fuentes que consume, la estructura de control y las fases de extracción, transformación y carga sobre la Base A. Luego, la Subsección 5.2.2 presenta la Etapa B, enfocada en el procesamiento de la Base A para construir las ventanas temporales y tablas derivadas del Modelo B en la Base B. Finalmente, la Subsección 5.2.3 desarrolla la Etapa C, que ingiere y procesa los archivos `PM Data` para poblar la base documental del Modelo de PMs.

5.2.1. Arquitectura de Etapa A

La Etapa A realiza la extracción de las fuentes `coordenadas` y `common-context`, las procesa de acuerdo a lo definido en el Modelo A y carga las estructuras resultantes en la base de datos asociada al Modelo A (Base A). En la Figura 5.2 se presenta un diagrama detallando este proceso.

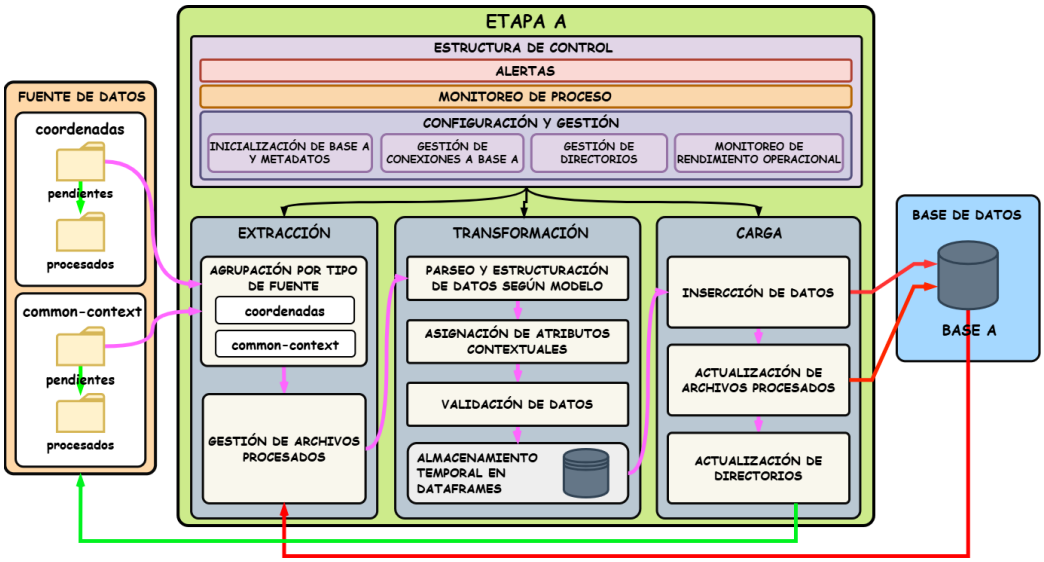

Figura 5.2: Diagrama de la arquitectura de la Etapa A del pipeline de procesamiento de datos. El esquema muestra las fuentes utilizadas, las fases de ejecución (extracción, transformación y carga), la estructura de orquestación con sus funciones principales y la base de datos de destino. Las líneas rojas indican la comunicación entre las etapas y la base de datos (Base A); las líneas verdes, la interacción de la última etapa con las fuentes; y la línea violeta representa el flujo de procesamiento de los datos

Fuentes de Datos

Se consideran dos fuentes de datos: `coordenadas` y `common-context`. Estas se organizan en directorios separados, uno por tipo de archivo, donde cada directorio contiene dos subdirectorios: `pendientes` y `procesados`. El subdirectorio `pendientes` almacena los archivos aún no procesados y `procesados` conserva los que ya fueron previamente procesados. Esta organización permite evitar

re-procesamientos y proporciona conocimiento sobre la cantidad de archivos procesados y pendientes de procesar. Dado el volumen actual, esta estrategia es válida. Sin embargo, para mayores volúmenes de datos se debería evolucionar hacia un esquema de organización y almacenamiento más escalable. Operativamente, el movimiento de archivos desde `pendientes` a `procesados` se realiza tras una carga de archivo exitosa.

Estructura de Control

La estructura de control actúa como orquestador de la Etapa A. Inicia y finaliza cada fase del proceso, valida sus resultados antes de habilitar la siguiente etapa y notifica ante errores, aplicando las políticas definidas (interrupción del proceso, omisión de archivos, rollback para evitar inserciones parciales). Además, monitorea la ejecución de punta a punta, actualizando el estado de los datos y del flujo. En cuanto a la operativa, inicializa la base de datos y sus metadatos en caso de que corresponda, gestiona el ciclo de vida de las conexiones a la base de datos, la organización de directorios (lectura, escritura y movimiento de archivos entre `pendientes` y `procesados`) y la implementación de métricas para validar rendimiento. Entre las métricas mínimas que expone se incluyen latencia por etapa y volumen procesado.

Extracción

Esta etapa realiza validaciones sobre los archivos ubicados en los directorios `pendientes` de ambas fuentes, con el fin de garantizar su integridad y unicidad antes del procesamiento. Dado que las dos fuentes presentan estructuras y periodicidades distintas, se definen dos flujos de procesamiento independientes para ordenar la manipulación. A efectos de la descripción se presentan como un único proceso. Para ambas fuentes se valida que el archivo sea JSON bien formado y que su nombre respete la convención `[tipo_fuente]_[fecha]_[hora].json` (por ejemplo `coordenadas_YYYYMMDD_HHMM.json`). A partir de esa etiqueta se deriva un timestamp que se asocia a los datos del archivo como parámetro temporal.

Tras estas validaciones, se contrasta cada archivo contra un registro persistente de archivos procesados (tabla en Base A) para evitar re-procesamientos. La verificación se realiza por medio del parámetro timestamp del archivo. Si existe una coincidencia previa, el archivo se descarta, en caso contrario continúa en el flujo. De esta forma se garantiza que en la base solo se tenga almacenada en la base un archivo `common-context` por día.

Transformación

La etapa de transformación tiene como objetivo estructurar los datos conforme a las definiciones del Modelo A. El procesamiento se realiza un archivo a la vez y el parseo es incremental, lo que permite recorrer archivos grandes sin cargarlos completamente en memoria. Esto se implementa de esta forma ya que los archivos

de tipo `common-context` contienen cientos de miles, por lo que un enfoque en memoria completa resultaría ineficiente.

A partir del parseo, los datos se mapean a las estructuras del modelo, se les agrega el parámetro timestamp y se valida la presencia de campos obligatorios. Si se detectan inconsistencias, se omite la carga del archivo completo para preservar la integridad. No se aplica eliminación selectiva de registros porque las estructuras presentan dependencias fuertes entre parámetros y la omisión parcial podría degradar la información de manera difícil de controlar.

Cuando las validaciones resultan satisfactorias, las estructuras derivadas se almacenan temporalmente en DataFrames, y luego se ejecuta la carga controlada hacia las bases de datos. Este almacenamiento temporal ofrece un mayor control en la fase de carga permitiendo insertar los datos en las correspondientes tablas de la Base A en una única transacción, permitiendo revertir el proceso en caso de presentarse algún error, sin dejar resultados parciales.

Carga

La etapa de carga inserta las estructuras temporales, generadas durante la etapa de transformación, en las tablas correspondientes de la Base A. Para preservar la integridad, cada archivo se carga en una única transacción con la base. Aunque el procesamiento se lleve a cabo de forma iterativa, estructura por estructura, todas las inserciones del archivo se ejecutan dentro de una misma transacción y sólo se confirman al finalizar el proceso sin errores. Si alguna inserción falla, se realiza rollback y el estado de la base queda igual al estado previo a la carga de ese archivo. Una vez confirmada la transacción, se actualiza la tabla de control de archivos procesados y se realiza la actualización de directorios, moviendo el archivo al subdirectorio `procesados`.

5.2.2. Arquitectura de Etapa B

La Etapa B realiza la extracción de las estructuras de la Base A, las procesa de acuerdo a lo definido en el Modelo B y carga las estructuras resultantes en la base de datos del Modelo B (Base B). En la Figura 5.3 se presenta un diagrama que expone de forma simplificada este proceso.

Fuente de Datos

La fuente de datos de esta etapa es la Base A. A partir de sus tablas se extrae una ventana temporal móvil con los últimos seis meses de datos y se aplican las operaciones necesarias para construir tablas derivadas en Base B.

Estructura de Control y Gestión

La estructura de control actúa como orquestador de la Etapa B. Inicia y finaliza cada fase del proceso, valida sus resultados antes de habilitar la siguiente y emite alertas ante fallas, aplicando las políticas definidas (interrupción del proceso,

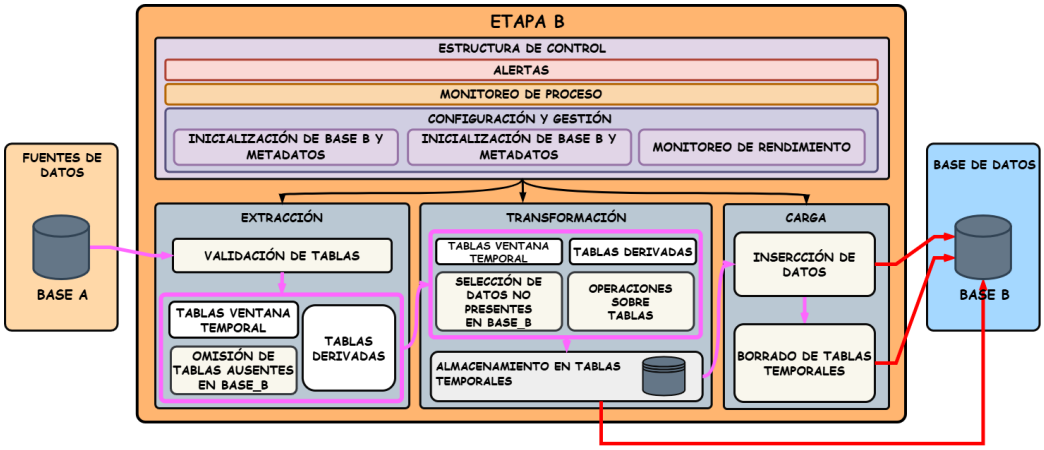

Figura 5.3: Diagrama de la arquitectura de la Etapa B del pipeline de procesamiento de datos. El esquema muestra las fuentes utilizadas, en este caso la Base A, las fases de ejecución (extracción, transformación y carga), la estructura de orquestación con sus funciones principales y la base de datos de destino (Base B). Las líneas rojas indican la comunicación entre las etapas y la base de datos; y la línea violeta representa el flujo de procesamiento de los datos.

omisión de archivos, rollback para evitar inserciones parciales). Además, monitorea la ejecución de punta a punta, actualizando el estado de los datos y del flujo en los registros de control.

En cuanto a la operativa, inicializa la base de datos y sus metadatos en caso de que corresponda, gestiona el ciclo de vida de las conexiones a la base de datos y la implementación de métricas para validar rendimiento (tiempo de procesamiento por archivo, cantidad de archivos procesados).

Extracción

La etapa de extracción comienza con una validación sobre la Base A. Para cada conjunto a procesar se comprueba que las tablas origen requeridas existan y que contengan datos. Este control es previo y obligatorio tanto para construir las ventanas temporales como para generar las tablas derivadas en Base B, dado que en ambos casos las fuentes provienen de Base A.

Para ventanas temporales, se parte de un inventario de tablas objetivo definido en Base B y se filtran en Base A aquellas que no estén contempladas en dicho inventario. Únicamente las habilitadas continúan a la transformación. Para tablas derivadas, no se aplica filtrado adicional, ya que el proceso opera sobre un conjunto explícito de tablas fuente previamente validadas.

Transformación

La etapa de transformación se ejecuta por medio de consultas a través de las base de datos. Para las tablas de ventanas temporales, la extracción se realiza en función del estado de su tabla homóloga en Base B. Si Base B está vacía, se efectúa la carga inicial de toda la ventana extraída de Base A. Si ya contiene datos,

se incorpora únicamente lo faltante mediante comparación contra el contenido de Base B, de forma que la re-ejecución sea consistente y no aparezcan duplicados.

En el caso de las tablas derivadas, la herramienta expresa las transformaciones como consultas SQL sobre las tablas de Base A. Para este tipo de tablas, las operaciones se realizan considerando únicamente la fecha más reciente de datos en la Base A. Esta elección simplifica tanto las operaciones entre tablas de Base A como la visualización en la herramienta, que no requiere conservar datos históricos.

En ambos casos, los resultados se almacenan primero en tablas temporales. Este almacenamiento transitorio tiene como propósito mantener la integridad de los datos. La carga de los datos en Base B se ejecuta dentro de una única transacción que abarca todas las estructuras de la ventana y sólo se confirma si las validaciones previas resultan satisfactorias. ante cualquier incidencia, se realiza rollback para evitar la carga parcial.

Carga

La etapa de carga toma las tablas temporales generadas en el paso previo e inserta su contenido en las tablas destino de Base B. Se distinguen dos agrupamientos de tablas: las de ventanas temporales y las derivadas. Para cada agrupamiento se abre una única transacción que abarca a todas las tablas del grupo y, dentro de esa transacción, cada tabla se carga iterativamente en lotes para no sobrecargar la base de datos.

La transacción sólo se confirma si todas las tablas del agrupamiento completan su carga sin errores. Ante cualquier incidencia, se ejecuta un rollback, de modo que ninguna tabla del grupo queda parcialmente actualizada. Tras un proceso de inserción exitoso, se eliminan las estructuras temporales. En caso de fallo, se interrumpe la transacción, se notifica y se eliminan las temporales sin modificar el estado previo.

5.2.3. Arquitectura de Etapa C

La Etapa C realiza la extracción de las fuente `PM Data`, las procesa de acuerdo a lo definido en el Modelo PMs y carga las estructuras resultantes en la base de datos del Modelo de PMs (Base PMs). En la Figura 5.4 se presenta un diagrama que expone de forma simplificada este proceso.

Fuente de Datos

Esta etapa consume datos directamente de la fuente `PM Data`. Al igual que en la Etapa A, la ingesta se organiza en dos directorios por fuente: `pendientes` y `procesados`. Los archivos arriban a `pendientes` y sólo se mueven a `procesados` tras un procesamiento exitoso.

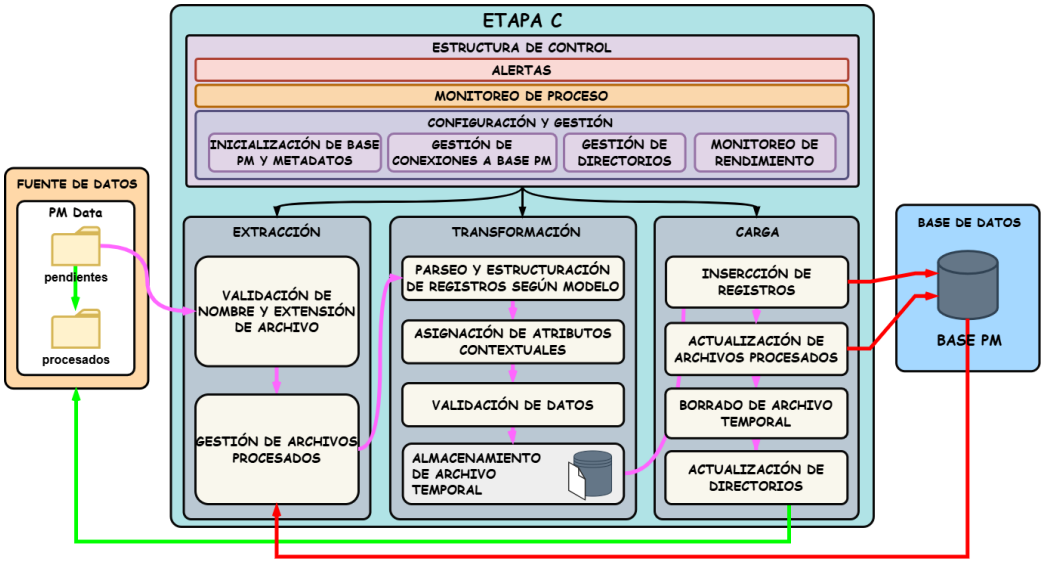

Figura 5.4: Diagrama de la arquitectura de la Etapa C del pipeline de procesamiento de datos. El esquema muestra la fuente de datos utilizada, en este caso `PM Data`, las fases de ejecución (extracción, transformación y carga), la estructura de orquestación con sus funciones principales y la base de datos de destino (Base PMs). Las líneas rojas indican la comunicación entre las etapas y la base de datos; las líneas verdes, la interacción de la última etapa con las fuentes para gestionar los cambios directorios de los archivos; y la línea violeta representa el flujo de procesamiento de los datos.

Estructura de Control

La estructura de control actúa como orquestador de la etapa C: gestiona la ejecución de cada fase, indicando comienzo y fin de cada una y habilitando la siguiente cuando corresponde. Cuando es necesario, inicializa la base de datos de destino y sus metadatos, gestiona la apertura y cierre de conexiones, administra la lectura y escritura en los directorios de trabajo y monitorea el rendimiento operativo para disponer de una noción clara de la capacidad de la herramienta. Además, publica el estado del proceso, emite alertas y aplica las políticas de manejo de fallas.

Extracción

Al igual que en la Etapa A, la extracción se realiza sobre los archivos del directorio `pendientes`, procesando un archivo por vez. Antes de incorporarlos al flujo se valida que cada archivo sea JSON bien formado y que su nombre respete la convención `pm_YYYYMMDD_HHMM.json`.

Posteriormente, se verifica en un registro persistente de archivos procesados (colección de control en `Base PMs`) si el archivo ya fue procesado previamente. La verificación se realiza por nombre de archivo, campo que se indexa y sobre el que se establece una restricción de unicidad para acelerar la búsqueda y evitar re-procesamientos. Si existe una coincidencia previa, el archivo se descarta, en caso

contrario, continúa su proceso.

Transformación

La etapa de transformación se encarga de estructurar los archivos que llegan desde la extracción. El procesamiento comienza con un parseo incremental del documento, leyendo un registro por vez para evitar cargar el archivo completo en memoria. Durante el parseo se valida cada registro (integridad y completitud mínimas). Los que no cumplen los criterios se descartan y se dejan registrados para trazabilidad, mientras que los válidos se mapean a la estructura esperada por el modelo. Los atributos se formatean según las definiciones del modelo de datos. A medida que los registros válidos se transforman, se insertan en un archivo temporal que representa el mismo contenido del archivo de entrada pero adaptado al esquema objetivo. Este archivo se utiliza como fuente durante la etapa de carga.

Carga

Durante esta etapa se insertan en la base de datos los registros contenidos en los archivos temporales generados en la transformación. La inserción asegura que cada registro se escriba en la colección anual correspondiente, la cual se deriva del parámetro timestamp. Si la colección destino no existe, la herramienta la crea junto con sus índices al iniciar la carga del archivo. Una vez completado el proceso, se elimina el archivo temporal; la estructura de control registra el archivo procesado en la colección de control y se mueve el archivo del directorio `pendientes` a `procesados`.

5.3. Implementación

Esta sección materializa la arquitectura del pipeline en decisiones concretas de desarrollo. En la Subsección 5.3.1 se detallan las tecnologías y bibliotecas utilizadas para el procesamiento de datos y el acceso a las bases de datos. Luego, la Subsección 5.3.2 expone cómo se organiza el código y las fuentes en el proyecto, describiendo la estructura de directorios y el rol principal de cada archivo.

5.3.1. Tecnologías Utilizadas

El desarrollo se lleva adelante a través de Python por la amplia disponibilidad de bibliotecas para procesamiento de datos y acceso a bases de datos. A continuación se presentan las bibliotecas más relevantes utilizadas durante la implementación:

- **Pandas:** [34] librería de código abierto para el análisis y tratamiento de datos. Se utiliza para instancias de almacenamiento temporal de estructuras de datos.
- **iJSON:** [39] parser iterativo de JSON que permite procesar grandes archivos de forma incremental, reduciendo el uso de memoria durante la extracción, lo que hace que el procesamiento sea más eficiente.

- **SQL-Alchemy:** [6] toolkit para trabajar con bases de datos relacionales. Gestiona conexiones a bases de datos, transacciones y ejecución directa de SQL desde Python.

- **PyMongo:** [27] driver de MongoDB para Python. Se utiliza para gestionar conexiones, colecciones, índices y operaciones de inserción.

5.3.2. Estructuración de la Herramienta

La implementación del pipeline se organiza en una estructura de directorios pensada para reflejar la arquitectura por etapas definida en la Sección 5.2 y, al mismo tiempo, facilitar el mantenimiento. El directorio `ANTEL_data` concentra las fuentes originales (archivos de monitoreo, coordenadas y PMs), mientras que el directorio `Pipeline` reúne toda la lógica de extracción, transformación y carga.

Dentro de `Pipeline` se adopta una organización por etapas (`etapa_a`, `etapa_b` y `etapa_c`), donde cada etapa cuenta con sus propios módulos de orquestación, definición de tablas y funciones de proceso. Esta separación permite aislar problemas por modelo, acotar el impacto de posibles errores y evolucionar cada etapa de forma independiente. De forma complementaria, archivos generales como `db.py` y `aux_functions.py` centralizan la gestión de conexiones y las funciones auxiliares reutilizables, evitando duplicación de lógica. En la Figura 6.8 se presenta la estructura de directorios implementada, y en la tabla 6.3 se resume el rol principal de cada componente.

```
optical-networks/
    |-- ANTEL_data/
    |   |-- PM_full/
    |   |-- datos_common_context/
    |   ‘-- datos_coord/
    ‘-- Pipeline
        |-- aux_functions.py
        |-- db.py
        |-- etapa_a/
        |   |-- main_a.py
        |   |-- tablas_a.py
        |   ‘-- proceso_a.py
        |-- etapa_b/
        |   |-- main_b.py
        |   |-- tablas_b.py
        |   ‘-- proceso_b.py
        ‘-- etapa_c/
            |-- main_pm.py
            ‘-- proceso_pm.py
```

Figura 5.5: Estructura de directorios del pipeline de datos. El directorio `ANTEL_data` contiene las fuentes originales (archivos de monitoreo, coordenadas y PMs), mientras que `Pipeline` agrupa la lógica de procesamiento organizada en tres etapas (A, B y C) y los módulos auxiliares.

Tabla 5.1: Estructura de archivos del pipeline de datos.

Elemento	Descripción
`aux_functions.py`	Almacena funciones auxiliares reutilizables (validaciones, utilidades de formato, métricas, etc.) que no forman parte directa de la lógica de procesamiento de una etapa específica.
`db.py`	Gestiona las definiciones de conexión a las bases de datos relacional y documental (parámetros de conexión, creación de motores, helpers de sesión).
`etapa_a/`	Directorio que implementa la Etapa A. El archivo `main_a.py` actúa como orquestador: crea conexiones a la Base A, administra el acceso a los directorios de fuentes y coordina la ejecución de extracción, transformación y carga. El archivo `tablas_a.py` define la metadata de las tablas del Modelo A, y `proceso_a.py` implementa las operaciones concretas de cada fase del ETL.
`etapa_b/`	Directorio que implementa la Etapa B. El archivo `main_b.py` orquesta el procesamiento sobre Base A y Base B, gestionando conexiones y el ciclo de ejecución. El archivo `tablas_b.py` define la metadata de las tablas del Modelo B, y `proceso_b.py` implementa las transformaciones y cargas que materializan las ventanas temporales y tablas derivadas en Base B.
`etapa_c/`	Directorio que implementa la Etapa C. El archivo `main_pm.py` actúa como orquestador sobre la `Base PMs`, gestionando las conexiones a la base documental, el acceso a los directorios de `PM Data` y la ejecución de las fases del proceso. El archivo `proceso_pm.py` define las operaciones de extracción, transformación y carga de los registros de PMs.

5.4. Validación y Resultados

En esta sección se describen las estrategias y criterios de validación de la solución implementada, detallando las herramientas y los procedimientos aplicados. Posteriormente, se presenta un análisis de resultados obtenidos tras la ejecución completa de la versión presentada del pipeline. Este se realiza sobre el total de archivos disponibles, en un entorno local, cuya configuración se especifica, con el objetivo de caracterizar el desempeño.

Cabe aclarar que la metodología de validación se aplica exclusivamente a la solución implementada en este trabajo, conforme a lo definido en las secciones previas. La evaluación del montaje de la solución en el ecosistema Apache, desarrollada en paralelo por el equipo de investigación OMNI, queda fuera del alcance de esta sección.

5.4.1. Metodología de Validación

La metodología de validación integró tres enfoques complementarios: revisión manual, retroalimentación del cliente y mediciones instrumentadas de desempeño. En la validación manual, se efectuaron varias instancias de análisis pos-carga tras la implementación de cada etapa y una verificación del proceso completo al finalizar el desarrollo. En cada caso se comprobó el estado de las cargas en las bases de datos destino y la correcta ejecución del proceso en equipos con distintas especificaciones para tener una perspectiva general del procesamiento.

En paralelo, se realizaron tres instancias de entrega con ANTEL en las que se presentaron avances. Este intercambio complementó la validación interna, donde la retroalimentación del equipo operativo aportó una visión externa del problema. Finalmente, se incorporaron métricas instrumentadas para obtener una visión cuantitativa del desempeño del proceso: tiempos de ejecución por fase y del proceso completo, latencia media por archivo y volumen procesado (archivos y filas). Estas mediciones, junto con los registros y alerta emitidas por la estructura de control brindan evidencia objetiva sobre el comportamiento de la herramienta en el entorno de evaluación.

5.4.2. Resultados y Desempeño

Se lleva adelante una ejecución completa sobre la versión del pipeline presentada. El proceso se ejecuta sobre en un entorno local, en una computadora MacBook Pro [4] cuyas especificaciones se listan en la tabla 5.2. Las bases de datos utilizadas y sus versiones se listan en la tabla 5.3. En la tabla 5.4 se listan la cantidad de archivos por tipo junto con su tamaño medio por archivo, tamaño total y cantidad total de archivos. El proceso se realiza sin interrupciones, la totalidad de los archivos procesados son cargados al sistema.

Tabla 5.2: Especificaciones de la PC a utilizar para realizar el procesamiento completo de los datos (modelo de referencia: MacBook Pro 13", 2017).

Parámetro	**Descripción**
`OS`	macOS 10.15 Catalina
`CPU`	2.3 GHz dual-core Intel Core i5
`RAM`	8 GB LPDDR3 2133 MHz integrada
`Almacenamiento`	SSD 128 GB

El proceso se lleva adelante corriendo ambas bases de datos de manera local. Se almacenan los archivos en disco. El proceso se realiza desde cero, es decir, sin definir previamente ninguna base de datos, colecciones o esquemas. Los resultados temporales obtenidos tras el procesamiento se presentan en la tabla 5.5. Se presentan a detalle los tiempos de procesamiento mínimos, medio y máximo para la

Tabla 5.3: Bases de datos agrupadas por modelo; se indican el gestor y el motor, con sus respectivas versiones, utilizados durante el procesamiento.

Tipo	Bases de Datos	Gestor	Motor
`Base de Datos SQL`	`base_a`/`base_b`	MariaDB 10.4.28	InnoDB 10.4.28
`Base de Datos NoSQL`	`base_pm`	MongoDB 4.2.25	WiredTiger (integrado)

Tabla 5.4: Cantidad de archivos por tipo, junto con los tamaños mínimo, medio, máximo y total para cada categoría.

Tipo de Archivo	Cantidad	Tamaño			
		Mínimo	Medio	Máximo	Total
`coordenadas`	1	21.27 KB	21.27 KB	21.27 KB	21.27 KB
`common-context`	529	17.55 MB	33.46 MB	52.67 MB	18.84 GB
`PM Data`	61	11.59 MB	12.25 MB	13.24 MB	746.98 MB

Etapa A, correspondiente al procesamiento de los archivos `common_context` en la tabla 5.6, los resultados de la Etapa b en la tabla 5.7, donde se presentan los tiempos de procesamiento de las tablas viejas y de las tablas nuevas de acuerdo a como se indicó en secciones anteriores. Por último, la tabla 5.8 especifica los tiempos de procesamiento de los archivos `PM Data`.

Tabla 5.5: Tiempos de procesamiento por etapa.

Etapa	Tiempo Total (HH:MM:SS)
Etapa A	00:56:53
Etapa B	00:09:59
Etapa C	00:01:26

5.5. Resumen y Conclusiones

En este capítulo se describió el diseño e implementación del pipeline de datos encargado de poblar los modelos definidos en el Capítulo 4. A partir de los requisitos y limitaciones planteados se adoptó una arquitectura secuencial en tres etapas, alineadas con los Modelos A, B y de PMs, cada una con su propio proceso ETL y una estructura de control asociada. Se detalló la arquitectura general y la particular de cada etapa, así como las tecnologías utilizadas en la implementación, y se validó

Tabla 5.6: Tiempos de procesamiento por etapa (resumen).

Etapa	Tipo de Archivo	Cantidad de Archivos	Tiempo Total (HH:MM:SS)	Tiempo por Archivo (s)		
				Mínimo	Medio	Máximo
Etapa A	`common_context`	529	00:56:53	2.0	5.9	26.0

Tabla 5.7: Tiempos por etapa: total, ventanas temporales y creación de nuevas tablas.

Etapa	Tiempo Total (HH:MM:SS)	Tiempo Ventanas Temporales (HH:MM:SS)	Tiempo Nuevas Tablas (HH:MM:SS)
Etapa B	00:09:59	00:09:51	00:00:08

el comportamiento de la solución mediante pruebas completas sobre el conjunto de archivos disponibles, analizando tiempos de procesamiento y capacidad de carga.

Se concluye que el pipeline diseñado es capaz de procesar la totalidad de los archivos disponibles para cada una de las fuentes: un archivo de `coordenadas`, 529 archivos `common-context` (18.84 GB) y 61 archivos `PM Data` (746.98 MB). La ejecución punta a punta, en un entorno local con recursos moderados (MacBook Pro con CPU Intel i5 de dos núcleos, 8 GB de RAM y SSD de 128 GB), completa el procesamiento en aproximadamente una hora y diez minutos, respetando las dependencias entre modelos y garantizando la unicidad de carga mediante el uso combinado de directorios `pendientes`/`procesados` y tablas/colecciones de control. Las políticas de transacción por archivo o grupo de tablas y los mecanismos de rollback aseguran integridad, ante fallos no se escriben resultados parciales y el sistema conserva un estado consistente.

Desde el punto de vista del diseño, el pipeline cumple con los objetivos de robustez y mantenibilidad. La separación en tres etapas, cada una con su orquestador, permite aislar errores y extender el sistema por etapas sin afectar al resto. El uso de parseo incremental sobre los archivos JSON (`common-context` y `PM Data`) permite manejar volúmenes de cientos de miles de registros por archivo sin requerir cargas completas en memoria. La Etapa B, basada exclusivamente en consultas sobre Base A, habilita la construcción eficiente de ventanas temporales y tablas derivadas para el Modelo B, alineadas con las necesidades del visualizador y reduciendo el costo de consulta sobre el Modelo A. En paralelo, el Modelo de PMs se implementa sobre MongoDB, lo que facilita la incorporación de nuevos PIG y PI sin cambios de esquema.

Por otra parte, se identifican limitaciones ligadas principalmente al entorno de ejecución y al esquema de almacenamiento de los archivos crudos. Si bien para el volumen actual de datos el pipeline puede ejecutarse en un único equipo local con las especificaciones mencionadas, el crecimiento futuro, con una tasa esperada del orden de un archivo `common-context` y uno `PM Data` por día, exigirá, en algún punto, escalar verticalmente (mayor capacidad de cómputo y almacenamiento) o bien adoptar una estrategia más robusta de almacenamiento y gestión de archivos. Asimismo, la ausencia de control directo sobre la interfaz TRNBI y la recepción

Tabla 5.8: Tiempos de procesamiento por etapa (resumen).

Etapa	Tipo de Archivo	Cantidad de Archivos	Tiempo Total (HH:MM:SS)	Tiempo por Archivo (s)		
				Mínimo	Medio	Máximo
Etapa C	`PM Data`	61	00:01:26	1.0	1.4	1.8

de datos por lotes imponen restricciones sobre la latencia y regularidad de ingreso, que el pipeline mitiga mediante su esquema de directorios y control de archivos, pero no puede eliminar.

En conjunto, el diseño e implementación del pipeline proporcionan una base operativa sólida para explotar los modelos de datos construidos en el Capítulo 4. La solución actual demuestra que es posible integrar las tres fuentes de información en un flujo coherente que alimenta las bases de datos de los Modelos A, B y de PMs. Sobre esta infraestructura de procesamiento se apoyará, en el siguiente capítulo, la construcción de la herramienta de visualización y análisis, que hará uso directo de las estructuras generadas por el pipeline.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 6

Herramienta de Visualización de Datos

En este capítulo se describe el proceso de diseño e implementación de la herramienta de visualización de datos, construida sobre el modelo de información y el pipeline desarrollados en los capítulos anteriores. El objetivo del visualizador es transformar datos estructurados de la red en vistas operativas que faciliten su exploración e interpretación, manteniendo coherencia con los requerimientos de ANTEL y con las restricciones de ejecución del entorno objetivo.

La Sección 6.1 presenta los fundamentos de diseño adoptados para la interfaz y el modo en que los requerimientos funcionales se tradujeron en una arquitectura de información basada en categorías de navegación. A partir de este marco, la Sección 6.2 detalla el contenido implementado en cada categoría del visualizador, explicitando su propósito operativo, la información expuesta y el esquema de navegación utilizado.

Luego, la Sección 6.3 describe los aspectos de implementación que sostienen la solución, incluyendo el stack tecnológico seleccionado, los mecanismos que permiten su ejecución íntegramente en entorno local y la arquitectura y organización del código. La Sección 6.4 resume el enfoque de pruebas manuales y el proceso de validación funcional y operativa realizado sobre la herramienta en conjunto con ANTEL. Finalmente, la Sección 6.5 cierra el capítulo con un resumen de los principales resultados y conclusiones.

6.1. Diseño de Interfaz

La herramienta de visualización constituye el punto de contacto entre los repositorios de datos construidos en etapas previas y su uso operativo por parte de ANTEL. Por este motivo, el diseño de la interfaz no se limita a criterios estéticos, si no que además define cómo se navega la información, qué se prioriza y de qué forma se organiza el contenido para facilitar su interpretación y uso en operación.

La Subsección 6.1.1 presenta los criterios de diseño de la interfaz, definidos a partir de restricciones operativas y de iteraciones de retroalimentación con ANTEL. La Subsección 6.1.2 resume los requerimientos funcionales iniciales provistos por el cliente, que constituyen el punto de partida para estructurar la información.

Por último, la Subsección 6.1.3 describe el proceso de refinamiento de categorías y la estructura final de navegación adoptada. Esta estructura sirve de marco para la Sección 6.2, donde se detalla el contenido implementado en cada categoría del visualizador.

6.1.1. Criterios de Diseño de la Interfaz

La herramienta de visualización constituye la etapa final del flujo de desarrollo, integrando los modelos de datos y el pipeline implementados en los capítulos anteriores. Su objetivo es presentar al usuario la información de la red de forma directa, por lo que la interfaz debe ser clara, intuitiva y consistente. En particular, la organización visual de los elementos impacta de forma directa en la interpretación de los datos y en su utilidad operativa.

El diseño final se definió mediante un proceso iterativo que incluyó múltiples reuniones con ANTEL. En estas instancias se presentaron propuestas de interfaz y se ajustaron criterios de navegación, estructura de páginas y prioridad de indicadores. Un lineamiento central fue la necesidad de una visualización con alto contenido gráfico (gráficas, tablas y mapas), acompañada por una disposición clara que permita ubicar rápidamente la información relevante.

Sobre esta base se adoptó una estructura paginada, donde cada página agrupa información asociada a un conjunto de requerimientos específicos. Esta división se alinea con la lógica utilizada por ANTEL en su solicitud de KPIs, permitiendo una presentación segmentada y coherente con los objetivos de monitoreo. En la Figura 6.1 se muestra la distribución general de la información en la interfaz. A nivel de componentes, la interfaz se organiza en:

- **Barra lateral de navegación:** ubicada en el margen izquierdo (ver indicación 1 en Figura 6.1), resume las categorías temáticas de la herramienta y ofrece una visión global del contenido disponible. Se implementa como una barra vertical, encabezada con el logo de ANTEL y un esquema de colores alineado con la identidad visual de la empresa.

- **Tarjetas de resumen:** en la pantalla principal se presentan bloques (ver indicación 2 en Figura 6.1) que exponen conteos agregados de equipos por categoría (GX, ROADM y OLA). Para favorecer la asociación visual, estas tarjetas mantienen el mismo patrón cromático utilizado en las visualizaciones topológicas.

- **Paneles por pestañas:** para el inventario de hardware se utiliza una organización por pestañas (ver indicación 3 en Figura 6.1), que permite segmentar la exploración por tipo de equipo y acceder a su información general y estructural (por ejemplo, chasis y placas).

- **Indicador temporal global:** en el margen superior derecho se incluye la fecha correspondiente a la última carga de datos (ver indicación 4 en Figura 6.1), que contextualiza el estado de la información mostrada (en el entorno evaluado: 6 de agosto de 2025).

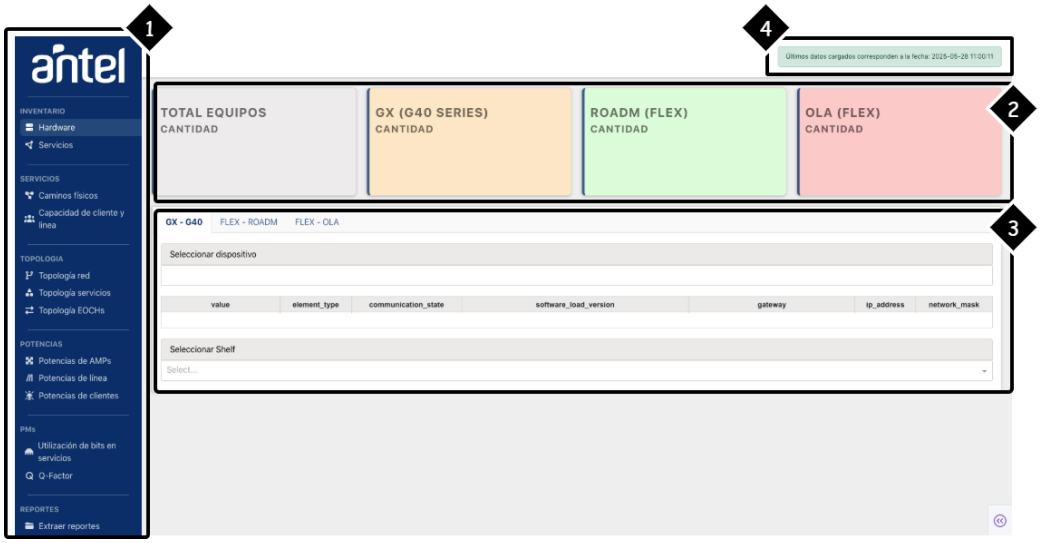

Figura 6.1: Pantalla principal del visualizador de datos de la red óptica. Se señalan cuatro elementos: (1) barra lateral con las categorías de navegación; (2) conteo total de equipos, diferenciados en GX, ROADM y OLA; (3) panel con pestañas para explorar información específica por tipo de equipo; y (4) fecha de la última carga de datos.

Definidos estos criterios de interacción y presentación, el diseño de la herramienta se completó a partir de los requerimientos funcionales provistos por ANTEL, que se resumen a continuación.

6.1.2. Requerimientos Funcionales Iniciales

La información a incorporar en la herramienta surge de los requerimientos iniciales proporcionados por ANTEL. Estos consisten en un conjunto de KPIs agrupados inicialmente por el cliente en distintas categorías: inventario, servicios, topología, métricas de desempeño (PMs) y reportes.

La tabla 6.1 presenta el listado original con los requerimientos mencionados, tal como fue entregado por ANTEL. Este listado representa el punto de partida para el diseño de la estructura de la información y manifiesta el conjunto de intereses y necesidades que ANTEL considera relevantes para la supervisión de la red.

Estos requerimientos constituyen la base para la organización de la herramienta en categorías temáticas; en la siguiente subsección se describe el proceso de refinamiento y la estructura final adoptada.

6.1.3. Refinamiento de Categorías y Estructura de Navegación

A partir del listado inicial de requerimientos, se llevó adelante un proceso iterativo de revisión y validación con ANTEL. Estas instancias de retroalimentación permitieron confirmar la utilidad operativa del contenido, ajustar criterios de presentación y redefinir indicadores cuando fue necesario.

Tabla 6.1: Listado inicial de requerimientos de ANTEL para la herramienta de visualización.

Requerimiento	**Descripción**
Inventario	Hardware y Servicios desplegados en la red.
Servicios	Caminos físicos y ópticos. Capacidad de línea y de cliente. Interfaces. Latencia.
Topología	Topología de red, de servicios y de OCHs.
PMs	Potencias ópticas de amplificadores, líneas y clientes.
Reportes	Pérdidas ópticas por tramo (Span Loss). Utilización de bits en servicios Ethernet. Utilización espectral por tramo y en toda la red. Latencia máxima de la red. Matriz de tráfico. Ocupación de Shelf por `Node`.

Como insumo complementario en las etapas iniciales se contó con una herramienta interna desarrollada por operarios de ANTEL, utilizada como referencia sobre las visualizaciones empleadas hasta ese momento. Si bien no presentaba una estructuración formal, aportó ejemplos concretos de información relevante para el monitoreo diario. En una fase posterior se tuvo acceso a otro visualizador utilizado por equipos de la red, con una interfaz local que, a partir de la selección de un equipo, muestra parámetros como estados operativos, cantidad de placas instaladas y puertos utilizados. Estas referencias contribuyeron a alinear el diseño con las prácticas y expectativas de los operarios.

Como resultado del refinamiento, se adoptó una organización por categorías temáticas (Figura 6.2), orientada a facilitar el acceso a la información y a brindar una visión global del contenido disponible. Se procuró conservar, en la medida de lo posible, los nombres de las categorías originalmente propuestas por ANTEL. No obstante, se introdujeron ajustes para reflejar con mayor precisión el contenido implementado. En particular, la categoría originalmente denominada PMs en el listado inicial (asociada a mediciones de potencia óptica) se renombró a Potencias (ver indicación 4 en Figura 6.2). En paralelo, se incorporó una nueva categoría denominada PMs (ver indicación 5 en Figura 6.2) para agrupar métricas de desempeño de mayor nivel, incorporadas durante las iteraciones con el cliente, como el Q-Factor.

Ademas, algunos indicadores inicialmente planteados como reportes se integraron a secciones con visualización interactiva. En particular, la utilización de bits en servicios Ethernet se incorporó dentro de PMs, dado que su análisis se beneficia de componentes dinámicos (gráficos y filtros temporales) en lugar de una presentación estática. De forma análoga, métricas originalmente contempladas en la categoría de Reportes (ver indicación 6 en Figura 6.2), como la utilización espectral por tramo, la latencia máxima de la red y la matriz de tráfico, fueron redistribuidas en otras secciones de la herramienta

La estructura final adoptada se resume en la tabla 6.2 y en la Figura 6.2. En la

Sección 6.2 se detalla el contenido implementado en cada categoría, indicando qué entidades y métricas se visualizan y cómo se navega la información.

Tabla 6.2: Categorías de la estructura final, implementada en la herramienta de visualización.

Requerimiento	**Descripción**
Inventario	Hardware. Servicios.
Servicios	Caminos físicos. Capacidad de cliente y línea.
Topología	Topología de red. Topología de servicios. Topología de OCHs.
Potencias	Potencias ópticas de amplificadores, líneas y clientes.
PMs	Utilización de bits en servicios; Q-Factor.
Reportes	Extracción de reportes.

6.2. Estructuración de Información

A partir de los criterios de diseño de interfaz y de la categorización final definidos en la Sección 6.1, esta sección describe la estructura de contenido implementada en la herramienta. Para cada categoría se presenta su propósito operativo, el tipo de información que expone y el esquema de navegación adoptado. La descripción sigue el orden de la barra lateral (Figura 6.2): Inventario (Subsección 6.2.1), Servicios (Subsección 6.2.2), Topología (Subsección 6.2.3), Potencias (Subsección 6.2.4), PMs (Subsección 6.2.5) y Reportes (Subsección 6.2.6).

6.2.1. Inventario

Esta categoría reúne la información estructural de la red. Permite conocer qué equipos están desplegados, cómo se componen y qué servicios se encuentran activos sobre ellos. El objetivo es ofrecer una vista ordenada del hardware y de los servicios, que apoye tareas de operación cotidiana, planificación de capacidad y mantenimiento. Para ello, la herramienta organiza el inventario en dos vistas principales: hardware y servicios.

Hardware

Esta sección presenta información detallada de los dispositivos que componen la red. Además del conteo general por tipo de dispositivo (G42, ROADM y OLA), se incluye un desglose de su estructura interna, presentando Shelfs, Slots por Shelf y puertos por Slot.

Para cada componente se reportan atributos relevantes, tales como estado operativo, versiones de software y hardware, número de serie, y datos del fabricante.

Figura 6.2: Barra lateral del visualizador con todas las secciones temáticas, donde se señalan seis indicaciones: (1) inventario de hardware y servicios; (2) servicios, con información de caminos físicos y capacidades de cliente y de línea; (3) topología de red, física, de servicios y de EOCHs; (4) potencias de amplificadores, de línea y de cliente; (5) PMs, con utilización de bits en servicios y Q-Factor; y (6) reportes, para la extracción de informes.

Esta información asiste a los operadores en tareas de control de inventario, planificación de actualizaciones y diagnóstico.

Como complemento, se incorporan representaciones gráficas del chasis asociadas al tipo de dispositivo del Node seleccionado (ver Figura 6.3). Estas visualizaciones

facilitan la identificación y ubicación de los componentes internos.

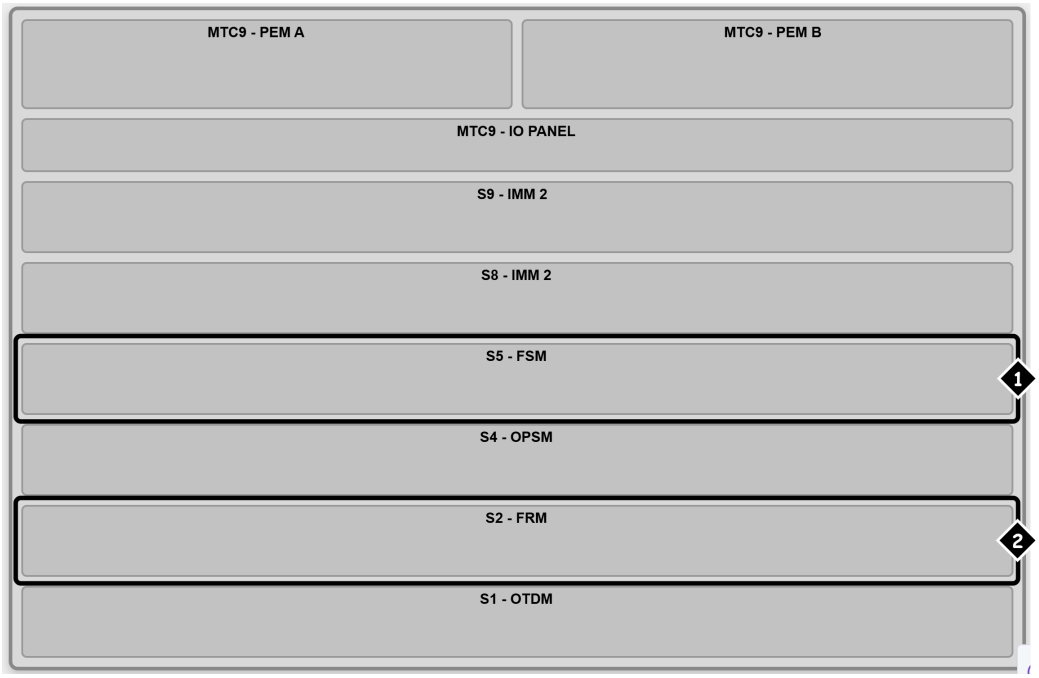

Figura 6.3: Chasis MTC-9 de un ROADM de un equipo de la red, mostrando placas que implementan bloques funcionales del ROADM, como CDC y WSS. Se indican: (1) placa FSM, que implementa el bloque CDC; y (2) placa FRM-20X, que implementa el bloque WSS.

Servicios

Esta sección lista los servicios activos por Node y, para cada uno, muestra su estructura de conexiones. Además, en la parte superior de la página se presenta un resumen mediante tarjetas con el total de servicios y la capacidad asociada, diferenciando transmisiones de 100 Gb/s y 400 Gb/s.

Al igual que en la vista de hardware, la información se complementa con una representación gráfica del chasis del Node seleccionado (ver indicación 1 en Figura 6.4). Cuando un puerto de una placa está en uso, se resalta en amarillo para indicar su ocupación por un módulo QSFP (ver indicación 2 en Figura 6.4). Esto aporta contexto visual, facilita la identificación de los componentes involucrados en cada servicio y apoya la evaluación de disponibilidad de hardware para nuevas demandas.

6.2.2. Servicios

Esta categoría se centra en cómo los servicios utilizan la infraestructura de la red: por dónde circulan físicamente y qué capacidad consumen en los puertos de cliente y de línea. El objetivo es ofrecer una vista que conecte la lógica de los servicios con los recursos concretos que los soportan, asistiendo tanto en el análisis

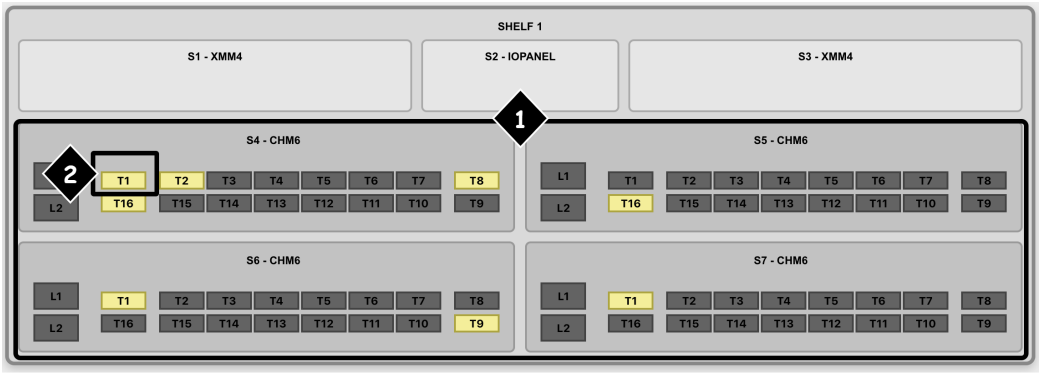

Figura 6.4: Shelf 1 de un equipo G42, con 4 placas CHM6 y los puertos utilizados. Se señalan dos elementos: (1) cantidad de placas CHM6 instaladas de las 4 posibles, mostrando que todas están colocadas; y (2) puertos ocupados, resaltados en amarillo.

de impacto ante fallos como en la planificación de nuevas altas. En esta sección se distinguen, por un lado, los requerimientos que finalmente se descartaron y, por otro, los que fueron incorporados en la herramienta.

Requerimientos descartados

- **Caminos ópticos:** refieren a la ruta lógica recorrida por las portadoras ópticas (EOCHs). Dado que esta información se aborda en profundidad en la categoría Topología, se considera redundante incluirla también en esta sección, por lo que se descarta.
- **Interfaces:** en el contexto de servicios, las interfaces corresponden a los puertos cliente. Esta información se incluye en Inventario (servicios), por lo que se descarta aquí para evitar redundancias.
- **Latencia:** representa el tiempo de propagación extremo a extremo de un servicio, determinado por la distancia del recorrido, las características de la fibra y los retardos introducidos por los equipos intermedios. Sin embargo, la latencia no se encontraba disponible en los datos provistos por la red, por lo que se descarta.

Requerimientos implementados

Bajo la premisa de respetar la estructura original propuesta por ANTEL, se definió la inclusión de las siguientes subsecciones dentro de la categoría de servicios:

- **Caminos físicos:** representan visualmente la infraestructura física (tramos de fibra y equipos) sobre la cual se soportan los servicios. Si bien esta información se utiliza en otras categorías, su incorporación en Servicios habilita funcionalidades específicas de valor operativo, como identificar qué servicios se verían afectados ante el corte de un tramo (EOTS) y detectar los tramos más críticos en términos de afectación.

- **Capacidad de línea y cliente:** representa la capacidad de transmisión de los puertos cliente (puertos Tributary) y de los puertos de línea asociados a los EOCHs. Dado que se trata de información específica, se la complementa con herramientas visuales que permiten explorar los dispositivos, sus Shelfs y Slots, junto con el estado de los puertos asociados. Esto facilita la toma de decisiones respecto a la disponibilidad de recursos para nuevos servicios.

6.2.3. Topología

La visión topológica constituye un aspecto central tanto para la visualización como para la gestión operativa de redes ópticas. Cada una de las tres secciones contempladas (topología de red, topología de EOCHs y topología de servicios) aporta una perspectiva complementaria que permite al operador comprender la infraestructura desplegada de la red.

Si bien la topología de servicios se construye sobre la topología de EOCHs, dado que los servicios se viajan a través de estos, ambas se presentan como secciones disjuntas. La sección de EOCHs se enfoca en la lógica de conexión a nivel de canal óptico, la sección de servicios permite observar explícitamente la relación extremo a extremo entre los dispositivos involucrados. Por otro lado, la topología de red proporciona la base física sobre la cual se construyen los dos niveles lógicos anteriores. Esta sección permite visualizar el despliegue geográfico de los Node y Link, sirviendo como punto de anclaje para la interpretación de las capas superiores.

Topología de Red

Se incluye un contador en la parte superior, con la cantidad total de tramos de fibra (EOTS), así como la cantidad de dispositivos por tipo G42, ROADM y OLA (ver indicación 1, Figura 6.5). Además, esta sección incluye un mapa interactivo de Uruguay en el que se trazan los Link y se ubican geográficamente los Node (ver indicación 2, Figura 6.5) donde se expone la topología de red representando la infraestructura física.

Topología de Servicios

La topología de servicios representa la visión lógica del recorrido extremo a extremo de cada servicio desplegado sobre la red. Esta visualización se construye a partir de la identificación de los EOCHs que transportan los servicios, permitiendo comprender no solo los extremos involucrados, sino también la secuencia de elementos intermedios.

Para su exploración, se adopta un esquema de navegación jerárquico. En primer lugar, se filtra por Node GX, lo que permite listar los servicios que lo involucran como Node extremo. A partir de la selección de un servicio, se despliega una tabla que describe su trazado a lo largo de las distintas capas de abstracción del modelo, desde su nombre en la capa de servicio hasta la ruta física en la red. Esta representación multicapas brinda al operador una comprensión clara de cómo el servicio se construye y despliega sobre la infraestructura subyacente.

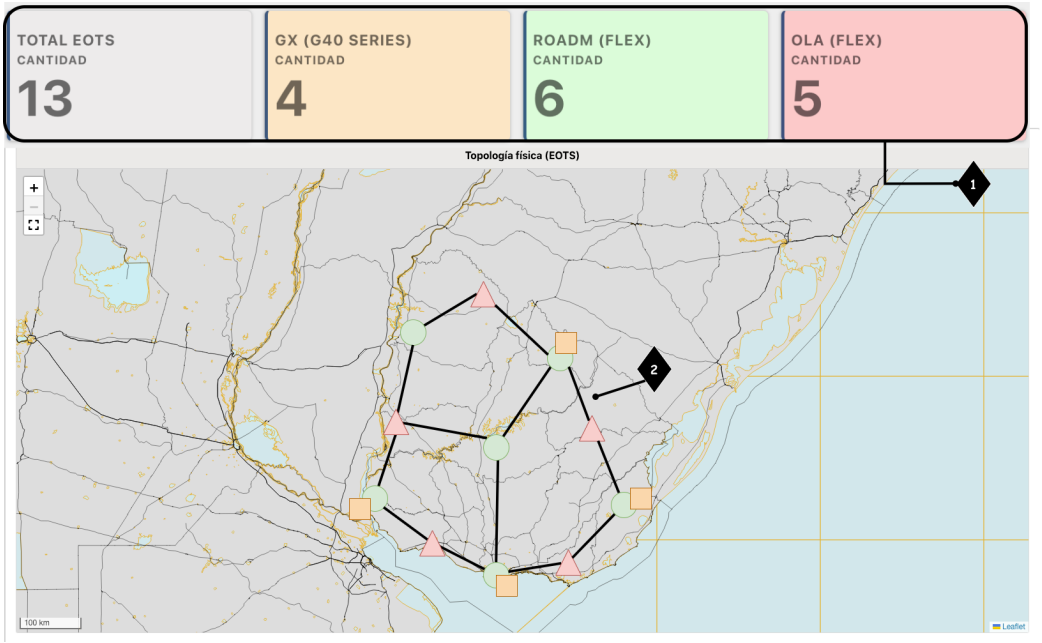

Figura 6.5: Se presenta a modo de ejemplo la topología física de la red mediante dos indicaciones: (1) cantidad de elementos, incluyendo EOTS que componen la red, y equipos de las series G42, ROADM y OLA; y (2) representación gráfica del grafo físico de la red, a modo de ejemplo. Por razones de confidencialidad, se ilustra con una estructura ficticia.

Además, se incorpora un mapa, sobre el cual la herramienta permite visualizar el recorrido asociado al servicio seleccionado, contemplando tanto el camino nominal como el camino protegido. Esto resulta fundamental para la identificación de los elementos implicados ante un eventual fallo. Como complemento, se incorpora información sobre los elementos que conforman los extremos del servicio:

- **Puertos de línea:** estado operativo y administrativo, capacidad disponible, características de EOCH (modulación, banda espectral, tipo de grilla, roll-off factor) e información de protección
- **Puertos cliente:** estado, capacidad de transmisión y rangos soportados de potencia óptica.

Topología de EOCHs

La topología de EOCHs, al igual que la topología de servicios representa una visión lógica de la red. El filtrado se realiza directamente por EOCH, permitiendo seleccionar hasta dos EOCHs de forma simultánea. Para cada uno de ellos se presenta información sobre la portadora, como el ancho de banda ocupado y sus características específicas. Esta información se complementa con una visualización interactiva sobre el mapa de Uruguay, donde se representa el recorrido físico extremo a extremo de los EOCHs seleccionados, incluyendo todos los Node y saltos intermedios.

Se hace la diferencia entre un EOCH nominal y un protegido. De esta forma se puede observar si se cumple que un EOCH su camino nominal y protegido son disjuntos respecto a la infraestructura que utilizan (ver Figura 6.6). Además, para cada EOCH elegido se lista el conjunto de tramos de fibra (EOTS) que atraviesa. Para cada EOTS se incorpora una representación gráfica del uso espectral, donde se muestra la posición de los EOCHs sobre el espectro disponible. Esta visualización muestra el espectro óptico y permite al operador identificar la disponibilidad de espacio para asignar nuevas portadoras, lo que resulta especialmente útil para la planificación y gestión operativa.

Como complemento, se incluye una visualización global del espectro que reúne todas las portadoras activas en la red. Esta representación ofrece una capa adicional de análisis que facilita la evaluación del estado general del uso del espectro de la red.

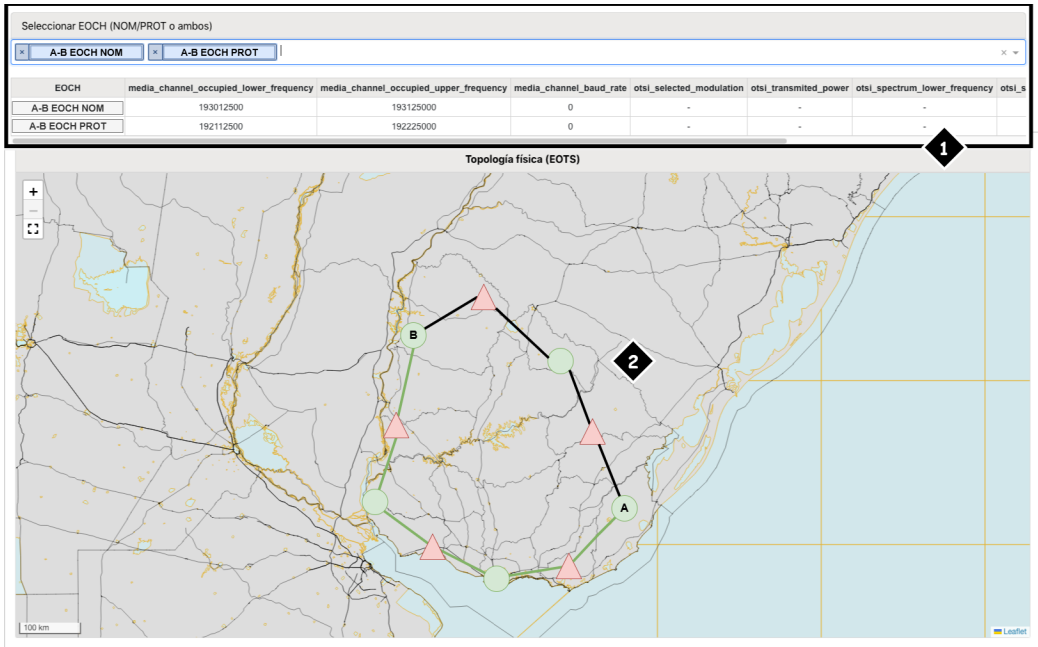

Figura 6.6: Vista de la sección de topología de EOCHs. En el ejemplo se seleccionan dos EOCH que conforman el par nominal (NOM) y protegido (PROT) de dos Node (A y B). En (1) se muestran sus principales parámetros, incluida la frecuencia de operación, y en (2) el mapa se representan sus recorridos sobre la red, con el camino nominal en negro y el protegido en verde.

6.2.4. Potencias

Esta categoría incluye los valores de potencia de amplificadores, en puertos de línea y en puertos cliente. A continuación se detallan las características de cada sección.

Potencias de Amplificadores

Se muestran los valores de potencia de entrada y salida para los amplificadores ópticos. El enfoque adoptado es el filtrado por servicio: dado un determinado servicio, se identifican todos los Node intermedios que contienen elementos amplificadores. Los dispositivos que se consideran con capacidad de amplificación son los ROADM y los OLA. Si bien la función específica de los OLA es la amplificación, mediante los Slots IAM-B-ECH2, se consideran también como componentes amplificadores los ROADM, por medio de la amplificación booster de los Slots FRM-20X. Para cada uno de estos Node, se permite visualizar:

- Gráficos de evolución temporal de la potencia de entrada y salida.
- Información sobre los puertos de entrada/salida implicados.
- Posibilidad de descargar los datos crudos en formato Comma-Separated Values (CSV).

Como complemento, se incluye un mapa interactivo que permite visualizar el recorrido del servicio seleccionado, para la ruta nominal o para la protegida, sobre el global de la topología física. También se incluye un contador de dispositivos intermedios (ROADMs y OLAs) lo cual ayuda a contextualizar el recorrido del servicio dentro de la topología de red.

Potencias de Línea

Esta sección muestra los valores de potencia de entrada y salida correspondientes a los puertos de línea, responsables del transporte de información a través de los EOCHs. Para acceder a estos datos, se toma un criterio de filtrado jerárquico basado en la estructura física de los dispositivos: Node → Shelf → Slot → Puerto. Para cada puerto de línea seleccionado, se presenta:

- Información de estado operativo y administrativo.
- Asociación con el EOCH correspondiente y los servicios que lo atraviesan.
- Gráfico de evolución temporal de la potencia de entrada y salida.
- Opción de descarga de los datos crudos en formato CSV.

Además, se integra una representación gráfica del chasis del dispositivo, lo que permite ubicar visualmente los puertos de línea y los Slots activos dentro del Shelf seleccionado.

Potencia Cliente

Esta sección presenta los valores de potencia óptica de entrada y salida correspondientes a los puertos cliente de los dispositivos de tipo GX. Al igual que en el caso anterior, se adopta un esquema de filtrado jerárquico basado en la estructura física del Node (Node → Shelf → Slot → Puerto). Para cada puerto cliente asociado a un servicio seleccionado, la herramienta ofrece:

- Visualización de los gráficos de evolución temporal de la potencia de entrada y salida para el puerto cliente.
- Información del estado operativo y administrativo del puerto.
- Capacidad de transmisión en Gb/s del servicio asociado.
- Indicación sobre la disponibilidad de datos de PMs.
- Opción de descarga de los datos crudos en formato CSV.
- Representación gráfica del chasis del dispositivo, para facilitar la ubicación de los puertos seleccionados.

6.2.5. Parámetros de Performance

Esta sección se presenta dos PMs que fueron requeridos con gran interés, dado que representan aspectos más específicos sobre del rendimiento de la red. Separar cada métrica en una sección específica permite profundizar en su visualización, incorporando herramientas que faciliten la interpretación por parte del operador, tales como mapas, gráficos temporales e interfaces gráficas de los dispositivos involucrados.

Utilización de Bits por Servicio

Esta sección presenta información sobre la cantidad de bits transmitidos a través de los puertos cliente donde residen los servicios. Se considera para ambos extremos del servicio, permitiendo visualizar su comportamiento en cada dirección de transmisión. Para acceder a esta información, se implementa un sistema de filtrado por servicio. La visualización incluye:

- Información general del servicio y sus puertos extremos.
- Gráfico de evolución temporal de la utilización para cada extremo.
- Posibilidad de selección de rango temporal.
- Descarga de los datos en crudo.

Q-Factor

La herramienta permite visualizar su evolución temporal para cada puerto de línea, enfocándose en aquellos ubicados en los Node extremos de los EOCHs. La implementación sigue el criterio jerárquico adoptado en otras secciones, permitiendo la navegación por Node $\rightarrow$ Shelf $\rightarrow$ Slot $\rightarrow$ Puerto. Para cada puerto seleccionado, la herramienta ofrece:

- Identificación del EOCH y del dispositivo asociado.
- Información operativa del puerto.

- Selección de ventana temporal.
- Gráfico de evolución del Q-Factor en el tiempo.
- Descarga de los datos en crudo.

6.2.6. Reportes

Esta categoría surge como una solución para presentar información operativa relevante que no se ajustaba a otras secciones de la herramienta, ya sea por su nivel de detalle o su carácter complementario.

El enfoque final de esta sección se basa en ofrecer una recopilación acotada de reportes operativos, principalmente orientados a inventario, servicios y disponibilidad de información de Performance Management (PMs). Se optó por representaciones simples y exportables, permitiendo al operario manipular los datos de forma directa mediante archivos en formato CSV.

Los reportes de inventario muestran información física de la red: cantidad de Shelfs por Node y el detalle de Shelfs y Slots (con datos de hardware, fabricante y número de serie) útiles para mantenimiento y planificación. Los reportes de servicios indican cuántos servicios activos tiene cada Node (por capacidad de 100 Gb o 400 Gb) y listan los servicios con sus extremos, puertos, estado y capacidad. Los reportes sobre PMs (Performance Monitoring) ofrecen una visión general y básica de los PIGs disponibles, permitiendo listar los PMs por Node, verificar disponibilidad de datos por fecha y exportar métricas en formato CSV para análisis externo.

6.3. Implementación

6.3.1. Stack Tecnológico

A continuación se presenta el stack tecnológico utilizado para desarrollar la herramienta de visualización y justifica su selección en función de los requerimientos de interacción, extensibilidad y ejecución local. En particular, se describen las tecnologías base empleadas para implementar la interfaz y el acceso a datos, así como el rol que cumple cada componente dentro de la arquitectura de la solución.

Lenguaje y Framework de Desarrollo

Para el desarrollo de la herramienta se optó por Python, principalmente por la disponibilidad de bibliotecas orientadas a análisis y visualización de datos y por su integración directa con los repositorios construidos en etapas previas. Dentro del ecosistema Python existen múltiples frameworks para aplicaciones web interactivas. En la etapa inicial se evaluaron distintas opciones, buscando un equilibrio entre facilidad de implementación, flexibilidad visual y control detallado sobre el comportamiento de la interfaz.

La primera alternativa explorada fue Streamlit [36], un framework de rápida implementación, pero con limitaciones relevantes en términos de personalización

del diseño y control fino de la interfaz. La segunda alternativa fue Vizro [40], que ofrece mayor flexibilidad visual y una estructura más modular, aunque su nivel de abstracción limita el control necesario para implementar comportamientos específicos requeridos por el proyecto. Finalmente, se seleccionó Dash [18], un framework full-stack para Python desarrollado por Plotly, que combina una implementación directa en Python con un alto grado de control sobre la interacción y la estructura visual.

Dash permite construir aplicaciones web interactivas sin definir explícitamente una separación entre backend y frontend. La lógica de la aplicación se implementa en Python, mientras que internamente se apoya en Flask para el backend y en React para el renderizado del frontend. Una funcionalidad central del framework es el sistema de callbacks, que define el comportamiento reactivo de la interfaz: tablas, gráficos, mapas y otros componentes se actualizan dinámicamente en respuesta a acciones del usuario o cambios en el estado de la aplicación. Además, Dash dispone de un ecosistema de bibliotecas de componentes que facilita la construcción de interfaces completas y personalizables.

Bibliotecas Complementarias

Además de Dash y sus bibliotecas de componentes, el desarrollo se apoyó en bibliotecas del ecosistema Python para manipulación de datos y acceso a las bases de datos. Las principales son:

- **Pandas:** biblioteca utilizada para estructurar los datos extraídos, aplicar transformaciones (filtrado, limpieza y agregación) y preparar la información para su visualización en tablas y gráficos.
- **SQLAlchemy:** toolkit para el acceso a bases de datos relacionales mediante consultas SQL. Se utiliza para conectarse a la base relacional del Modelo B y ejecutar consultas sobre las tablas empleadas por las distintas secciones del visualizador.
- **PyMongo:** biblioteca para interactuar con bases de datos MongoDB. Se emplea para acceder al Modelo de PMs, realizar consultas específicas y obtener series temporales y métricas utilizadas en visualizaciones y reportes.

6.3.2. Ejecución en Entorno Local

Por requerimiento de ANTEL, la herramienta de visualización debía ejecutarse íntegramente en un entorno local, sin dependencia de conexión a internet. En consecuencia, todos los componentes necesarios para su funcionamiento debían estar disponibles de forma local, evitando llamadas externas a servidores o recursos remotos.

Un punto crítico para cumplir este requerimiento fue la visualización del mapa interactivo. En su configuración por defecto, la librería Dash Leaflet obtiene los fragmentos de mapa mediante peticiones a una API pública provista por servidores

externos. Dado que este comportamiento no era compatible con las restricciones de ejecución definidas, fue necesario adaptar la solución para operar en modo offline.

Para ello, se obtuvo un mapa vectorial de Uruguay a partir de una plataforma de código abierto MapTiler [MapTiler]. Posteriormente, este insumo se procesó utilizando Quantum Geographic Information System (QGIS) [35], una herramienta de código abierto para tratamiento de información geoespacial, empleada aquí para generar el conjunto de fragmentos de mapa requeridos por Dash Leaflet. Finalmente, los fragmentos se organizaron en la estructura de directorios esperada por la aplicación y se almacenaron localmente dentro del proyecto, de modo que la interfaz pueda renderizar el mapa sin depender de servicios externos.

6.3.3. Arquitectura y Organización del Código

Se presentan dos perspectivas complementarias de la implementación de la herramienta: la arquitectura de la aplicación y la organización del código que la materializa. Primero se describe la arquitectura modular utilizada para implementar las distintas vistas del visualizador y la centralización del acceso a datos. Luego se detalla la estructura de directorios adoptada y el rol de los principales archivos del proyecto.

Arquitectura de la Aplicación

La herramienta adopta una arquitectura modular por sección: cada vista o categoría del visualizador se implementa como un módulo independiente. Este enfoque facilita incorporar nuevas secciones o ajustar las existentes sin afectar el resto de la aplicación. En términos de implementación, cada sección se organiza en dos archivos bien diferenciados:

- **Archivo de interfaz y callbacks:** define el layout de la vista y los callbacks asociados.
- **Archivo de funciones:** implementa las funciones invocadas por los callbacks (transformaciones, acceso a datos, validaciones y formateo).

Los archivos de layout de las distintas secciones se integran mediante un archivo central (`app`), que cumple el rol de orquestador de la herramienta. Su estructura es análoga a la de las secciones, ya que define el layout principal, los callbacks transversales y se apoya en un archivo `app_functions`, donde se agrupan las funciones auxiliares utilizadas por dichos callbacks. En particular, `app`:

- Define la estructura base de la interfaz y determina qué sección visualizar según la interacción del usuario.
- Implementa callbacks transversales (cambio de sección, parámetros globales y estado compartido entre vistas) y coordina la actualización de contenidos.
- Inicializa la aplicación, registra las secciones disponibles y expone el punto de ejecución.

El acceso a las bases de datos se realiza a través de los archivos de funciones de cada sección, pero se centraliza en un módulo específico (`db`). Este módulo encapsula las definiciones y la configuración de conexión tanto para la base relacional como para la base documental. Adicionalmente, contiene las definiciones de las tablas del Modelo B utilizadas por la herramienta, lo que permite un acceso consistente a los datos desde las distintas vistas.

En la Figura 6.7 se presenta un diagrama que resume la estructura descrita y la comunicación entre los principales componentes.

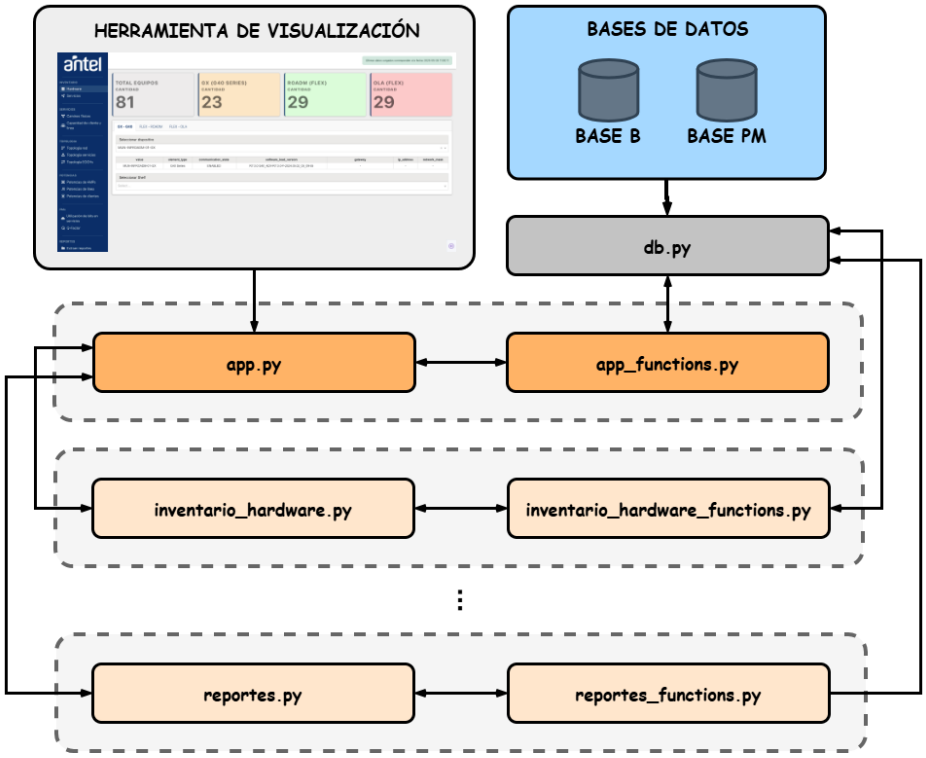

Figura 6.7: Diagrama de comunicación de archivos de la herramienta. Representa la arquitectura modular de la solución, mostrando los archivos centrales y aquellos donde se definen sus funciones, así como la forma en que se vinculan entre sí. También se ilustra su relación con las bases de datos a través del módulo db y la integración general realizada por app.

6.3.4. Estructuración de Directorios

Para sostener la modularidad por secciones y facilitar el mantenimiento, el código y los recursos de la herramienta se organizan en directorios con nombres descriptivos, separando responsabilidades de forma explícita. La estructura se definió según los siguientes criterios:

- **Separación por responsabilidad:** archivos de interfaz y callbacks por un lado, y funciones por otro. Además, se dispone de un directorio dedicado a

la gestión de conexiones a bases de datos y de un archivo central para la orquestación de la aplicación.

- **Convenciones de nombre:** directorios y archivos autoexplicativos, de modo que ubicar componentes específicos sea lo más directo posible.

En la Figura 6.8 se presenta la estructura de directorios implementada para la herramienta de visualización. Una descripción resumida del rol de cada componente se muestra en la tabla 6.3.

```
optical-networks/
    |
    `-- Dashboard
        |-- app.py
        |-- assets/
        |-- db/
        |-- files/
        `-- pages/
```

Figura 6.8: Estructura de directorios de la herramienta de visualización. Se muestra el directorio `Dashboard`, que contiene los archivos relacionados con la interfaz y su ejecución.

Tabla 6.3: Estructura de archivos de la aplicación.

Elemento	Descripción
`assets`	Directorio que contiene imágenes, recursos necesarios para desplegar el mapa en modo local y hojas de estilo para el funcionamiento offline de la interfaz.
`app.py`	Archivo principal de la aplicación; su ejecución inicializa el sistema y pone en marcha el visualizador.
`db`	Directorio que contiene el módulo encargado de centralizar la configuración y las definiciones de conexión a las bases de datos relacional y documental.
`files`	Directorio que almacena funciones asociadas a cada sección de la herramienta (consultas, transformaciones, validaciones y formateo).
`pages`	Directorio que contiene los archivos de interfaz y callbacks correspondientes a cada sección del visualizador.

6.4. Pruebas y Validación

Para verificar el correcto funcionamiento de la herramienta se adoptó un enfoque en dos niveles. En primer lugar, se realizaron pruebas manuales orientadas a validar el comportamiento de la interfaz y de los flujos de navegación bajo patrones de uso típicos. En segundo lugar, se llevó adelante una validación funcional y operativa con ANTEL, basada en instancias de entrega y retroalimentación, con el

objetivo de confirmar la utilidad y pertinencia de las visualizaciones e indicadores implementados.

La Subsección 6.4.1 describe las pruebas realizadas, mientras que la Subsección 6.4.2 presenta las estrategias de validación adoptadas.

6.4.1. Pruebas Manuales de Interfaz

Las pruebas se realizaron de forma manual. Este enfoque se adoptó por el carácter interactivo de la herramienta y por la necesidad de evaluar la coherencia entre múltiples componentes visuales (tablas, gráficos y mapas) ante acciones del usuario. Las pruebas incluyeron, entre otros aspectos, la navegación entre secciones, la aplicación combinada de filtros, la actualización simultánea de visualizaciones y la descarga de resultados en formato CSV.

6.4.2. Validación Funcional y Operativa

La validación funcional y operativa se llevó adelante mediante tres instancias de entrega coordinadas con ANTEL, alineadas con los hitos de inventario, topología y métricas de desempeño. En cada instancia se presentaron avances al equipo de Transporte y se obtuvo retroalimentación sobre la claridad de la interfaz, la utilidad de las visualizaciones, la pertinencia de los indicadores y el comportamiento general de la aplicación. Las observaciones recopiladas se utilizaron para priorizar ajustes e incorporar mejoras iterativas entre entregas.

En cada entrega se incluyó documentación de avance con el estado funcional del sistema. En la última instancia, dicha documentación se consolidó en una guía de uso que describe la estructura de la herramienta, su navegación y los principales casos de uso previstos para los operadores.

6.5. Resumen y Conclusiones

Este capítulo presentó el diseño e implementación de la herramienta de visualización de datos, desde la definición de la interfaz y la estructuración de la información hasta la selección del stack tecnológico, la arquitectura de software y la organización del código. Se describió cómo los requerimientos operativos de ANTEL se tradujeron en categorías temáticas (inventario, servicios, topología, potencias, PMs y reportes), cómo se materializaron en vistas interactivas apoyadas en gráficos, tablas y mapas, y cómo la herramienta se integra con las bases de datos construidas por el pipeline de procesamiento. Por último, se presentaron las formas de validación de la herramienta, para la cual se implementó testing manual a partir de la interacción con la aplicación, así como un proceso de validación iterativa basado en las devoluciones del equipo de ANTEL en cada entrega.

El desarrollo realizado brinda a los operarios una forma directa y práctica de visualizar la topología física de la red, los servicios desplegados y los parámetros de desempeño más relevantes, integrando en una misma solución información que antes se encontraba dispersa en múltiples fuentes. El visualizador ofrece, además,

una visión global del estado y distribución de los equipos. El diseño de la interfaz se mantuvo deliberadamente sencillo e intuitivo, priorizando una organización clara de las secciones y de las visualizaciones para facilitar el entendimiento de la red y reducir la curva de aprendizaje para los operadores. A nivel de desarrollo, la construcción de la herramienta supuso también una curva de aprendizaje significativa sobre Dash y su ecosistema. Fue necesario un período de estudio y experimentación para familiarizarse con su funcionamiento antes de converger al diseño actual.

La arquitectura del sistema, basada en una organización modular por secciones y en la separación entre interfaz y lógica de acceso a datos, facilita la incorporación de nuevas funcionalidades. Este diseño permite ampliar el conjunto de PMs integrados, enriquecer las visualizaciones existentes o añadir secciones específicas de análisis y algoritmos sobre la topología y los servicios sin reestructurar la herramienta. En conjunto, la solución consolida un primer entorno de visualización sobre el que es posible construir capacidades más avanzadas de monitoreo y análisis, y que sirve como base para futuras líneas de trabajo orientadas a la optimización y automatización de la operación de la red. En el próximo capítulo se presentan las conclusiones generales del proyecto y se proponen posibles líneas de trabajo futuro.

Capítulo 7

Conclusiones y Trabajos Futuros

En este capítulo se presentan las conclusiones alcanzadas a partir del desarrollo del proyecto, organizadas en torno a sus tres componentes principales: modelo de datos, pipeline y herramienta de visualización. Para cada uno se ofrece una síntesis y las conclusiones específicas asociadas. Finalmente, se describen posibles trabajos futuros, tanto en forma de mejoras incrementales sobre lo implementado como de líneas de evolución derivadas de lo realizado.

7.1. Modelo de Datos

Se construyó un modelo de datos compuesto por tres componentes: un Modelo A, obtenido a partir de las fuentes `common-context` extraídas de la red; un Modelo B, derivado del Modelo A, cuyo propósito es materializar estructuras complejas a partir de las definiciones de A; y un Modelo de PMs, construido a partir de los archivos `\gls{pm}~Data`, que agrupan métricas de desempeño obtenidas vía Performance Management API.

El Modelo A y el Modelo de PMs representan el universo de la red (topologías, inventario y desempeño), mientras que el Modelo B se orienta específicamente al soporte de la herramienta de visualización, con el objetivo de simplificar costos de procesamiento y mejorar la experiencia de usuario.

7.1.1. Modelo A

Se obtiene un Modelo A que refleja la topología física de la red y la topología de servicios, eje central de la operativa en redes ópticas. Contar con una representación fiel de la distribución y de los componentes físicos permite definir rutas para nuevos servicios, entender los recorridos de los servicios actuales y tomar decisiones operativas con mayor fundamento.

Por ello, la representación del modelo debe ser lo más cercana posible a la red real. Un análisis de parseo de rutas, entre lo definido por T-API y las estructuras más significativas en términos de operación modeladas en el proyecto, permitió

estimar el alcance del modelo y evaluar el nivel de validez de la aproximación adoptada.

La elección de una estructura relacional con crecimiento por filas resultó adecuada para integrar las entidades T-API y acompañar el despliegue futuro de la red, ya sea por ampliación en nuevos equipos o por incorporación de nuevos servicios.

Además, el alto grado de relacionalidad del diseño habilita extensiones controladas del modelo, ya sea por expansiones del fabricante o por la inclusión de tablas inicialmente excluidas, sin comprometer la coherencia global del esquema.

7.1.2. Modelo B

Se obtiene un Modelo B derivado del Modelo A, diseñado para materializar estructuras complejas (relaciones y agregaciones) a partir de las tablas de A. Su propósito es simplificar el acceso y la navegación en la herramienta de visualización frente a los requerimientos de ANTEL, trasladando la complejidad al procesamiento para que la capa de interfaz opere sobre datos ya preparados y de consulta directa.

Esta estrategia demostró ser efectiva: reduce la complejidad de las consultas, simplifica la implementación de la interfaz y permite la visualización de estructuras complejas sin tiempos de carga que comprometan la experiencia del usuario. El intercambio adoptado —más lógica y uso de memoria durante el ETL a cambio de una exploración más fluida en la interfaz— resulta conveniente para el caso de uso, ya que concentra el esfuerzo en el pipeline y libera a la visualización de trabajo pesado.

7.1.3. Modelo de PMs

Se obtiene un Modelo de PMs a partir del procesamiento de los archivos `\gls{pm}~Data`. Se adopta un esquema documental que almacena, por registro, un PIG y lo relaciona con el Modelo A mediante el identificador de la entidad (`Device`, `Equipment`, `PhysicalSpan`, `AccessPort` o `Connection`) sobre la cual se realiza la medición. Los datos se guardan con granularidad de 15 minutos, lo que habilita la construcción de series temporales para su visualización a través de la herramienta y para análisis posteriores. Este registro periódico permite seguir la evolución de las métricas de manera consistente y deja una base útil para trabajos futuros (por ejemplo, análisis para anticipar fallas o generación de datos de prueba).

Durante la carga se aplicaron controles y validaciones para mantener la calidad de los datos y evitar errores. La principal limitación observada fue la disponibilidad de archivos, lo que impidió medir de forma concluyente el desempeño del esquema documental en términos de volumen y eficiencia. Aun así, por la naturaleza del dominio y el ritmo de generación de datos, se prevé un crecimiento sostenido para el cual el enfoque documental resulta adecuado.

Otro factor que condicionó el desarrollo fue la falta de documentación oficial específica de PMs/PIGs. Para suplirlo, se realizó un análisis poscarga de los datos: se relevaron y catalogaron los PIGs disponibles por tipo de entidad, se construyó

una estructura de referencia para conocer su disponibilidad y, sobre esa base, se elaboró un documento que reúne las descripciones de cada PIG y los PI asociados.

7.1.4. Resumen

El Modelo A y el Modelo de PMs abarcan el universo operativo de interés (representación de red y servicios, inventario y métricas de desempeño), mientras que el Modelo B aporta una capa derivada que simplifica el consumo en la herramienta de visualización.

El esquema resultante permite representar la infraestructura de la red tanto a nivel físico como lógico y brinda la capacidad de extraer información sobre su desempeño. Esto constituye una base sólida para la operación actual y para evoluciones futuras sobre la misma línea de trabajo.

7.2. Pipeline

Se construyó un pipeline capaz de extraer datos desde tres fuentes con estructuras diferentes, procesarlos conforme a los tres modelos definidos y cargarlos en sus bases de datos asociadas. Para su implementación se adoptó un proceso secuencial en tres etapas, orquestado por una estructura central que asegura la ejecución en orden y el aislamiento entre etapas. Cada etapa constituye un ETL en sí misma y se corresponde con un modelo de datos. El pipeline se ejecuta manualmente, con inicio y fin explícitos, tal como se definió para la operación.

El diseño incorporó desde el inicio trazabilidad por etapa y gestión de errores, con el objetivo de ofrecer un comportamiento robusto, evitando inconsistencias en los datos y facilitando el seguimiento del estado del proceso.

Su funcionamiento se validó mediante iteraciones de refinamiento y pruebas realizadas sobre el conjunto completo de datos disponibles. El desarrollo se llevó a cabo utilizando equipos propios, mientras que en paralelo se avanzó en la adaptación a la infraestructura de ANTEL. Esta actividad queda fuera del alcance del proyecto, siendo parte de una tarea ejecutada por el grupo de investigación OMNI.

El diseño estuvo condicionado por limitaciones del proceso de adquisición: la extracción desde el plano de control es gestionada exclusivamente por ANTEL. En el proyecto se recibieron los datos ya generados y se fijó su forma de ingreso al sistema, lo que restringió el diseño de la etapa de extracción. De haber contado con capacidad de consulta directa, podría haberse considerado un esquema casi en tiempo real o políticas de extracción selectiva, lo que justificaría un estudio comparativo de alternativas de pipeline y desempeños como trabajo futuro.

Aunque el desarrollo se centró en la red Infinera, que define especificidades propias, la solución admite adaptación a otras redes que soporten T-API.

7.3. Herramienta de Visualización

A partir de los requerimientos iniciales propuestos por ANTEL y de un análisis exhaustivo del modelo de datos para verificar su viabilidad, se desarrolló una herramienta de visualización alineada con los KPIs solicitados. El diseño prioriza una estructura clara y una interacción intuitiva, de modo que los operadores puedan localizar e interpretar la información con facilidad.

Todas las visualizaciones se sustentan en el modelado y procesamiento de los datos de la red; la solución se actualiza con cada nueva ingesta, manteniendo la información al día sin requerir cambios estructurales.

Sobre esta base, la herramienta representa en un mapa la topología física, de Lightpath y de servicios de la red Infinera de ANTEL, y permite consultar el inventario de hardware y servicios activos. Además, expone características específicas de estos elementos y métricas de desempeño relevantes sobre los dispositivos de la red.

La arquitectura modular adoptada facilita el crecimiento: es sencillo incorporar nuevas secciones o ampliar contenido sin afectar lo existente. Esto favorece la expansión de la herramienta, aspecto especialmente relevante en lo que refiere a PMs.

Tanto el diseño como la estructura se encuentran documentados en detalle, de modo que futuros desarrolladores puedan evolucionar la herramienta con criterios claros.

7.3.1. Resumen

Se diseñó e implementó la arquitectura de datos y la herramienta de visualización, se entregaron manuales de usuario y se documentaron en detalle los modelos de datos. Una parte sustantiva del esfuerzo se destinó al estudio del dominio y del caso de uso de ANTEL, que incluyó los conceptos de redes ópticas, los modelos T-API y las particularidades operativas, mediante la revisión de manuales de fabricantes, reuniones con el operador y el análisis de las fuentes de datos. Este proceso permitió alcanzar una comprensión profunda del caso de estudio y alinear la solución con las necesidades reales. Además, se trabajó con un cliente real, cumpliendo los plazos externos establecidos en el convenio FIng–ANTEL.

7.4. Trabajos Futuros

A partir de lo desarrollado, el proyecto abre varias líneas de continuidad. Por un lado, es posible profundizar sobre lo ya diseñado, refinando etapas del proceso o evaluando escenarios alternativos y, por otro, utilizarlo como base para estudios derivados.

7.4.1. Modelo de Datos

En lo que respecta al modelo de datos, una dirección concreta es aprovechar la topología para correr algoritmos de optimización de rutas de servicios. Este tema es objeto de estudio en el grupo de investigación OMNI. Poder correr este tipo de algoritmos sobre una estructura basada en T-API abre puertas en la continuación de esta línea de proyectos de investigación.

También puede avanzarse con la implementación de tablas contempladas inicialmente pero no procesadas por no ser de especial interés para los objetivos principales o por haber resultado vacías tras su incorporación. Dado que la red sigue en despliegue, es razonable esperar que, en el futuro, esa información gane utilidad o que estas estructuras inicialmente vacías comiencen a devolver datos.

En la misma línea, resulta pertinente evaluar esquemas de modelado alternativos (enfoque en grafos, documental o relacional con crecimiento por columnas), comparando desempeños y el grado en que simplifican la representación de la topología física, que es un objeto de interés central.

7.4.2. Pipeline

En cuanto al pipeline, cabe explorar alternativas que habiliten ingesta en streaming o la configuración de triggers en la entrada para detectar la llegada de nuevos archivos y orquestar automáticamente el proceso, añadiendo un mayor nivel de automatización.

Además, puede evaluarse y comparar el desempeño de diferentes herramientas y estrategias (incluida la ejecución en paralelo del procesamiento de PMs por un lado y de los Modelos A y B por otro). Esta línea podría constituir un proyecto en sí misma, con foco en aspectos técnicos de extracción, transformación y carga. Es importante recordar que el presente trabajo se enmarcó en un problema industrial con plazos definidos, lo que limitó la profundidad alcanzable.

7.4.3. Herramienta de Visualización

Respecto a la herramienta de visualización, puede profundizarse su refinamiento agregando nuevas secciones con información de interés, principalmente de PMs, o ajustando las existentes según necesidades del cliente. Por motivos de plazo no se avanzó más en estos puntos. Con base en el análisis poscarga y la documentación generada, podría realizarse un estudio para determinar qué otros parámetros convendría incorporar.

Otra funcionalidad que podría explorarse es aprovechar las capacidades que ofrecen las estructuras definidas por T-API en relación con el Switch Control. En un nivel de complejidad superior al de la topología de servicios, el Switch Control permite visualizar en la topología la implementación de la protección 1+1 sobre los caminos de Lightpath o EOCH, en caso de ser necesario.

7.4.4. Parámetros de Performance (PMs)

En PMs existe un potencial de exploración relevante. A partir del banco de datos disponible y de un procesamiento adecuado (tomando como referencia el análisis poscarga realizado), pueden implementarse y evaluarse modelos para detección temprana de fallas mediante series temporales de parámetros específicos. Esta línea resulta especialmente interesante por su potencial para aportar a la toma de decisiones en la operativa de ANTEL y por los intereses del equipo que lleva adelante este proyecto.

7.4.5. Migración a Infraestructura de ANTEL

La migración a la infraestructura de ANTEL representa un paso importante para la adopción efectiva de la solución. Esta actividad queda por fuera de lo contemplado dentro de los límites formales del proyecto y es llevada a cabo en paralelo por Bruno Tió, miembro del equipo de investigación del grupo OMNI. La consolidación de esta etapa permitirá evaluar el comportamiento del sistema en un entorno productivo y habilitar su uso directo por parte de los operadores de ANTEL.

Apéndice A

Diagrama de Modelo de Datos A

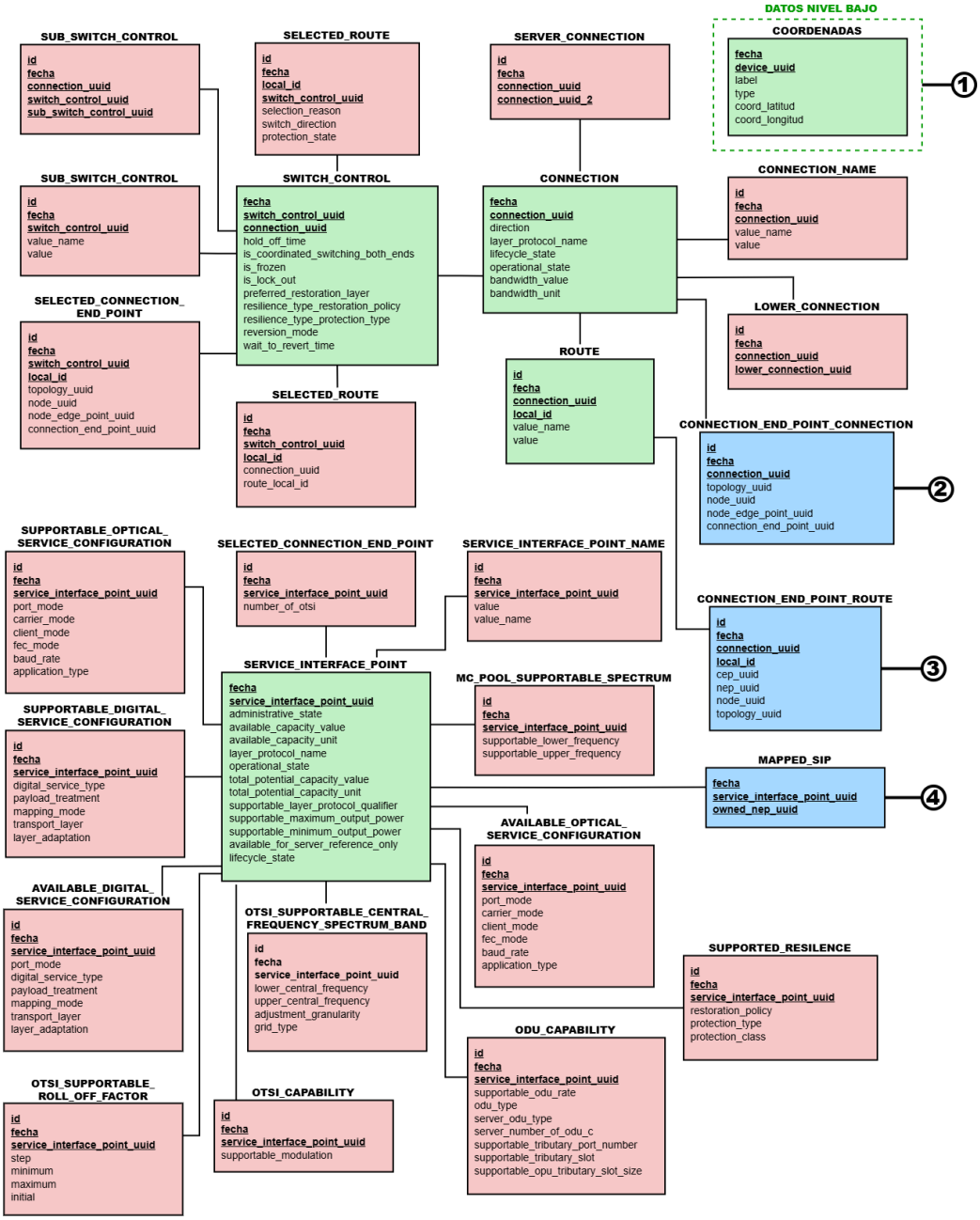

Figura A.1: Diagrama del Modelo A (parte izquierda), donde se muestran todas las entidades implementadas: en verde las entidades centrales, en azul las de vinculación y en rosado las de características. Cada entidad central que posee características tiene un número interno que las vincula, mientras que los números dentro de un círculo se utilizan para relacionar esta parte izquierda con la parte derecha del modelo. Se representan los datos de nivel bajo.

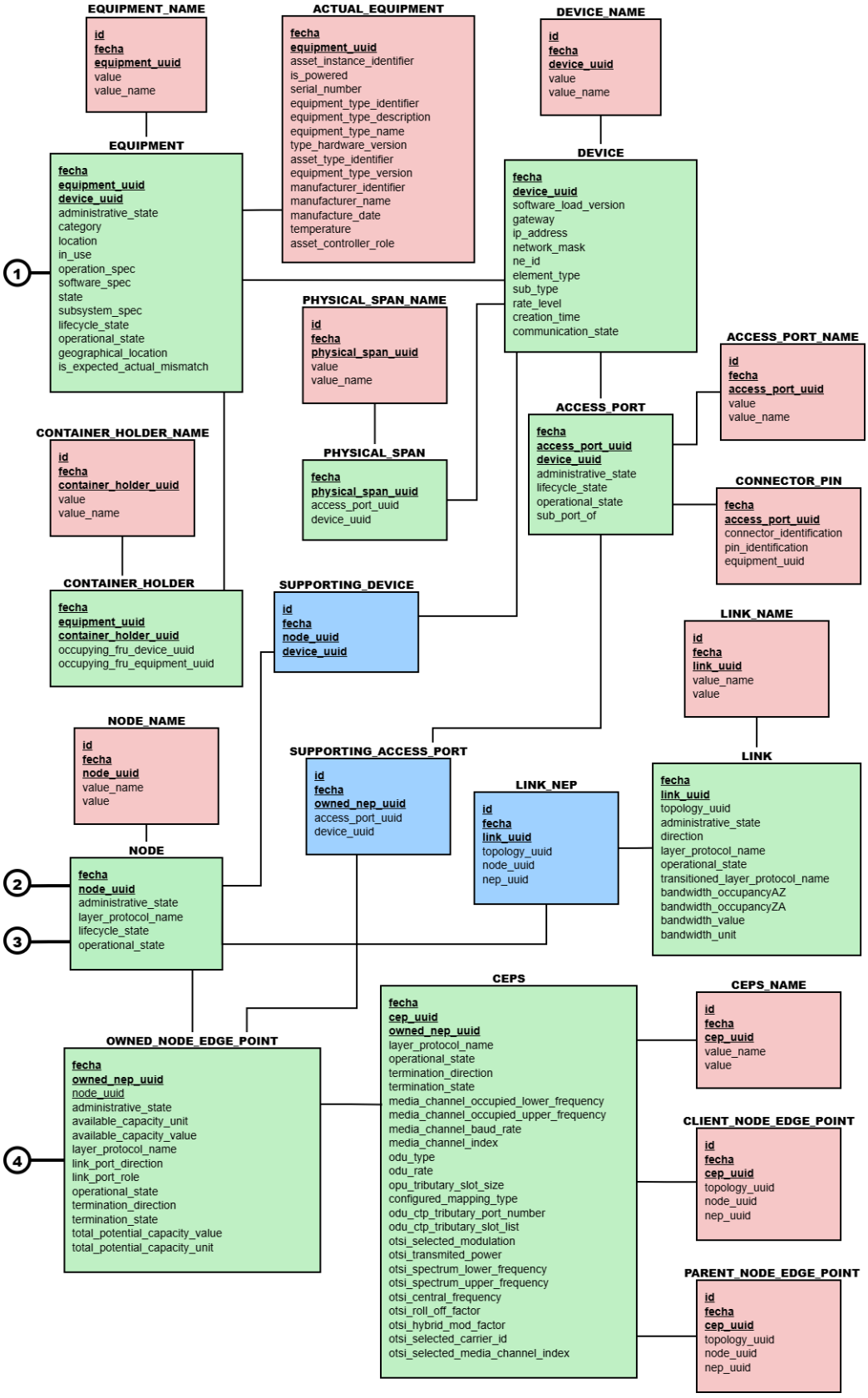

Figura A.2: Diagrama del Modelo A (parte derecha), donde se muestran todas las entidades implementadas: en verde las entidades centrales, en azul las de vinculación y en rosado las de características. Cada entidad central que posee características tiene un número interno que las vincula, mientras que los números dentro de un círculo se utilizan para relacionar esta parte derecha con la parte izquierda del modelo.

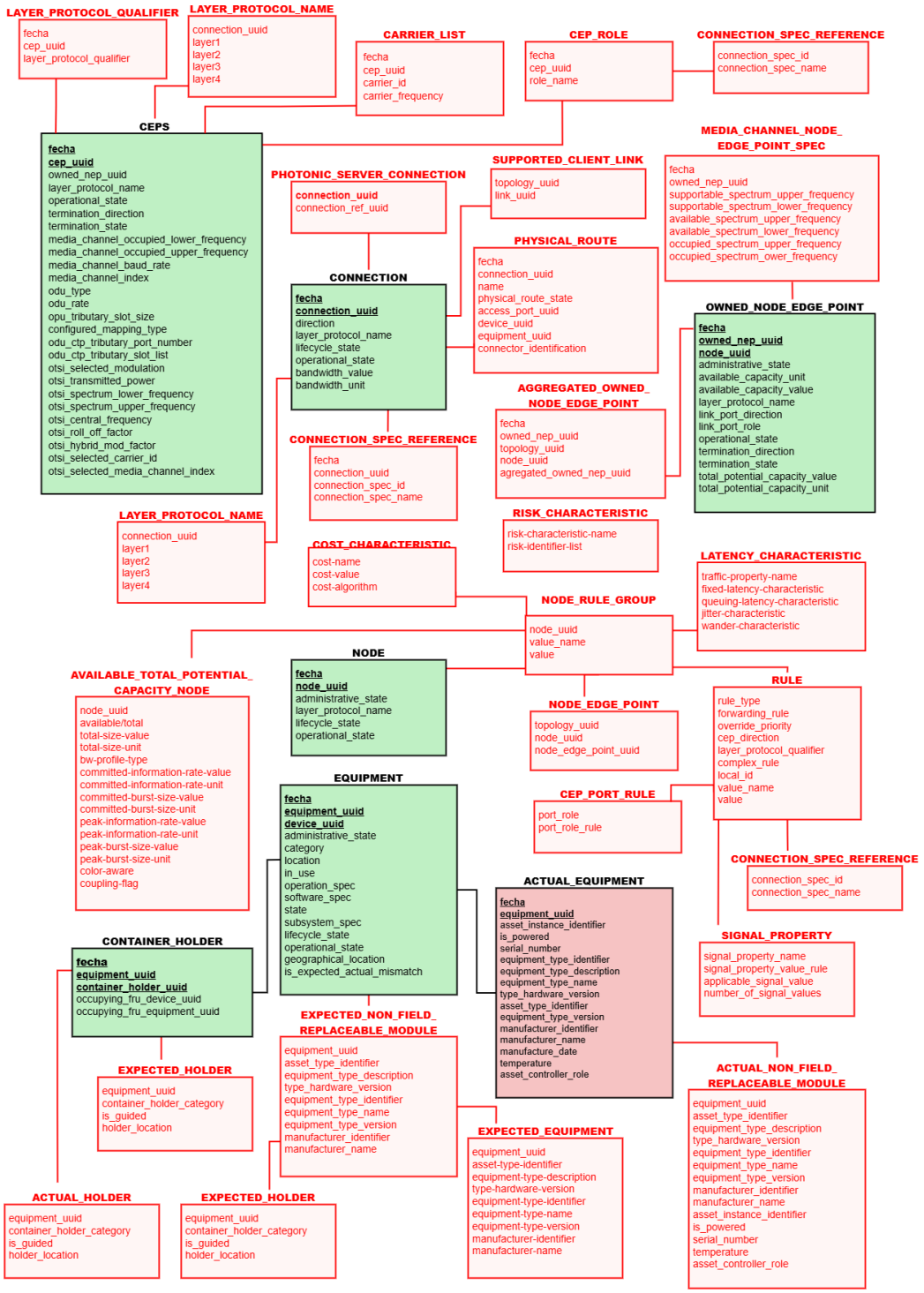

Figura A.3: Diagrama del Modelo A, donde se muestran las entidades no implementadas y su relación con las entidades ya implementadas (parte 1).

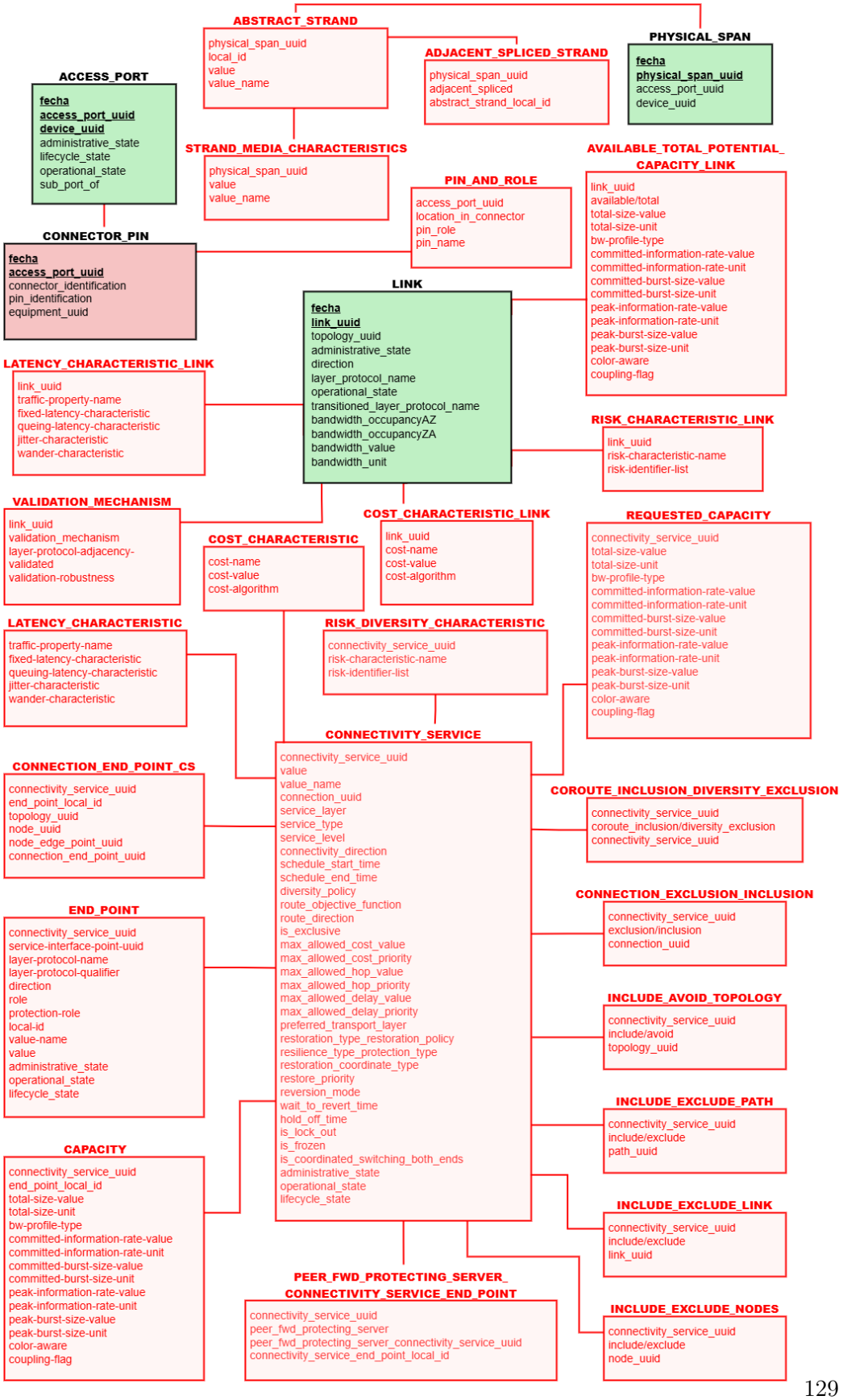

Figura A.4: Diagrama del Modelo A (parte 2), donde se muestran las entidades no implementadas y su relación con las entidades ya implementadas.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice B

Diagrama de Modelo de Datos B

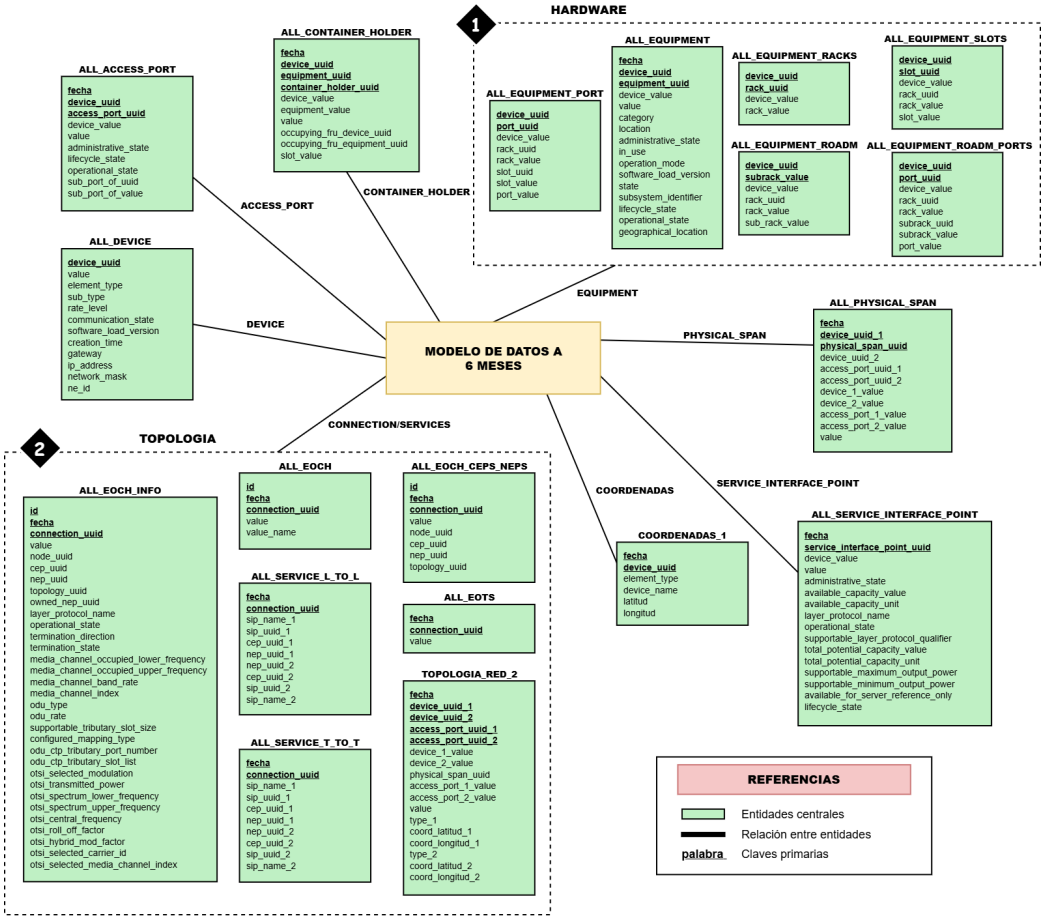

Figura B.1: Diagrama del Modelo B, donde se muestran las tablas de tipo `all_` junto con las relacionadas con la topología y el hardware. Se observa la relación del Modelo B como resultado de la integración de todas las tablas del Modelo A correspondientes a los últimos seis meses, además de las tablas adicionales propias del Modelo B. Se marcan dos indicaciones de interés: (1) se agrupan las entidades utilizadas para construir la jerarquía del hardware y en (2) las entidades empleadas para definir la topología.

Referencias

[1] (2020). *Infinera Transcend Software Suite: Enabling Evolution to Disaggregated, Virtualized, and Software-driven Hyperscale Networks*. Infinera Corporation. Document ID: 0162-SN-RevB-0420.

[2] (2022). *Infinera GX G40 Hardware Description Guide*. Infinera Corporation, release 6.0 edition. Version 6.0, November 2022.

[3] Agrawal, G. P. (2002). *Fiber-Optic Communication Systems*. Wiley-Interscience, New York, 3 edition. eBook ISBN: 0-471-22114-7.

[4] Apple Inc. (2017). Macbook pro (13-inch, 2017, two thunderbolt 3 ports) — technical specifications. `https://support.apple.com/en-us/111951`.

[5] Arturo Mayoral López de Lerma, Nigel Davis Ciena, A. M. (2020). TAPI v2.1.3 Reference Implementation Agreement. Technical report, Open Networking Foundation. `https://opennetworking.org/wp-content/uploads/2020/08/TR-547-TAPI-v2.1.3-Reference-Implementation-Agreement-1.pdf`.

[6] Bayer, M. (2012). Sqlalchemy. In Brown, A. and Wilson, G., editors, *The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks*. aosabook.org.

[7] Bierman, A., Björklund, M., and Watsen, K. (2017). Restconf protocol. RFC 8040, Internet Engineering Task Force (IETF). Accedido: 2025-10-25.

[8] Björklund, M. (2016). The yang 1.1 data modeling language. Technical Report RFC 7950, Internet Engineering Task Force (IETF). Accedido: 2025-10-25.

[9] Björklund, M. and Shafer, P. (2024). pyang: A yang validator, transformator and code generator. `https://github.com/mbj4668/pyang`. Versión 2.6.1.

[10] Elmasri, R. and Navathe, S. B. (2016). *Fundamentals of Database Systems*. Pearson, 7 edition. Global Edition.

[11] Enns, R., Björklund, M., Schoenwaelder, J., and Bierman, A. (2016). Rfc 7951: Json encoding of data modeled with yang. Accedido: 2025-10-25.

[12] Enns, R., Björklund, M., Bierman, A., and Schönwälder, J. (2011). Network Configuration Protocol (NETCONF). RFC 6241.

[13] FiberOptics4Sale Blog (2024). What is wavelength selective switch (wss)? https://www.fiberoptics4sale.com/blogs/archive-posts/95046534-what-is-wavelength-selective-switch-wss. Accedido: 2025-10-25.

[14] Fraunhofer HHI (2025). Fraunhofer hhi – the institute. https://www.hhi.fraunhofer.de/en/fraunhofer-hhi.html.

[15] Guillama, S. O. (2024). Aprendizaje automático aplicado al dominio de las redes ópticas. Tesis de grado, Universidad de la República (Uruguay). Facultad de Ingeniería, Montevideo, Uruguay. https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/48509.

[16] Huyen, C. (2022). *Designing Machine Learning Systems*. O'Reilly Media.

[17] Ignacio Bianchi, F. D. (2022). Ciencia de datos en el dominio de las redes ópticas. Tesis de grado, Universidad de la República (Uruguay). Facultad de Ingeniería, Montevideo, Uruguay. https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/30947.

[18] Inc., P. T. (2025). Dash: A python framework for building analytical web applications. https://dash.plotly.com. Accessed: 2025-11-01.

[19] Infinera Corporation (2025). Transcend trnbi interface specification. Technical report, Infinera Corporation. Accedido: 2025-10-25.

[20] International Telecommunication Union (ITU-T) (2020a). Recommendation g.694.1: Spectral grids for wdm applications: Dwdm frequency grid. Technical report, ITU-T. Accedido: 2025-10-25.

[21] International Telecommunication Union (ITU-T) (2020b). Recommendation g.709/y.1331: Interfaces for the optical transport network (otn). Technical Report G.709/Y.1331, ITU-T. Accedido: 2025-10-25.

[22] International Telecommunication Union (ITU-T) (2024). Recommendation g.872: Architecture of the optical transport network. Technical report. Accedido: 2025-10-25.

[23] Keiser, G. (2003). *Optical Communications Essentials*. McGraw-Hill Networking Professional. McGraw-Hill Professional, New York, NY.

[24] Keiser, G. (2011). *Optical Fiber Communications*. McGraw-Hill, New York, NY, USA, 4th edition.

[25] López, V. and Velasco, L. (2016). *Elastic Optical Networks, Architectures, Technologies, and Control*. Springer.

[MapTiler] MapTiler. Maptiler documentation. https://docs.maptiler.com/. Accessed: 2026-02-17.

[27] MongoDB, Inc. (2025). Pymongo: The official mongodb python driver. Version 4.14.0; accessed 2025-08-06.

[28] Open Network Models and Interfaces (ONMI) (2025). Tapi. `https://github.com/Open-Network-Models-and-Interfaces-ONMI/TAPI`. Accedido: 2025-10-25.

[29] Open Networking Foundation (ONF) (2012). Software-defined networking: The new norm for networks. White paper, Open Networking Foundation. Accedido: 2025-10-25.

[30] Open Networking Foundation (ONF) (2014). Sdn architecture, issue 1. Technical Report TR-502, Open Networking Foundation. Accedido: 2025-10-25.

[31] Open Networking Foundation (ONF) (2016). Functional requirements for transport api. Technical Recommendation TR-527, Open Networking Foundation. Accedido: 2025-10-25.

[32] Open Networking Foundation (ONF) (2021). Tapi reference implementation agreement v1.1. Technical Report TR-547, Open Networking Foundation. Accedido: 2025-10-25.

[33] Palmer, M. (2025). *Understanding ETL*. O'Reilly Media, updated edition edition. O'Reilly Online Learning.

[34] pandas development team, T. (2024). pandas-dev/pandas: Pandas.

[35] QGIS Development Team (2026). Qgis geographic information system. `https://qgis.org/`. Software. Accessed: 2026-02-17.

[36] Streamlit Inc. (2025). Streamlit: The fastest way to build data apps. `https://streamlit.io`. Accessed: 2025-11-01.

[37] The Fiber Optic Association (FOA) (2018). Fiber optics: The basics. Accedido: 2025-10-25.

[38] TM Forum (2024). Tmf628 performance management api rest specification r14.5.1. `https://www.tmforum.org/resources/interface/tmf628-performance-management-api-rest-specification-r14-5-0/`.

[39] Tobar, R. and ijson contributors (2024). ijson: Iterative json parser with pythonic interfaces. Version 3.3.0; accessed 2024-06-06.

[40] Vizro Team, JPMorgan Chase & Co. (2025). Vizro: Build powerful, composable data apps in python. `https://vizro.readthedocs.io`. Accessed: 2025-11-01.

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

Índice de figuras

Esta es la última página.
Compilado el viernes 20 febrero, 2026.
http://iie.fing.edu.uy/