



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención de Computación

**Diseño e Implementación del Network Data Analytics
Function (NWDAF) según el Estándar de 3GPP
para la Gestión de Datos Analíticos y Predictivos
en Redes 5G**

Autor:
D. Pelayo Torres Alonso



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención de Computación

**Diseño e Implementación del Network Data Analytics
Function (NWDAF) según el Estándar de 3GPP
para la Gestión de Datos Analíticos y Predictivos
en Redes 5G**

Autor:

D. Pelayo Torres Alonso

Tutor:

D. Jesús M. Vegas Hernández

Resumen

El Network Data Analytics Function (NWDAF) es una entidad crucial en el ecosistema de redes modernas, especialmente aquellas que operan bajo los estándares definidos por la 3rd Generation Partnership Project (3GPP). Esta función tiene la responsabilidad de analizar los datos generados en la red con el fin de proporcionar información valiosa que facilite tanto la optimización como la gestión eficiente de la red.

La función principal del NWDAF es recopilar y procesar datos provenientes de diferentes puntos dentro de la infraestructura de red. Estos datos pueden incluir información sobre el tráfico de la red, el estado de los dispositivos conectados, el rendimiento de los componentes de la red, entre otros. Una vez recopilados, estos datos son sometidos a un procesamiento avanzado que permite extraer información relevante.

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas y entidades que han hecho posible la realización de este Trabajo de Fin de Grado.

En primer lugar, quiero agradecer profundamente a mis padres. Su amor incondicional, sacrificios y constante apoyo han sido la piedra angular de mi camino académico. Gracias por ser mi fuente inagotable de inspiración y por creer en mí desde el principio. Su aliento y orientación han sido fundamentales para alcanzar este logro.

A mi tutor de TFG, Jesús M. Vegas Hernández por su orientación, paciencia y valiosos consejos durante el desarrollo de este proyecto. Su dedicación y experiencia han sido fundamentales para alcanzar los objetivos propuestos.

También quiero agradecer a Telefónica por la oportunidad de realizar mis prácticas de empresa. En especial, quiero expresar mi gratitud a Rodrigo Sanz Sanz y Jesús Macias Portela por su ayuda y guía en el proyecto. Gracias por permitirme aplicar mis conocimientos en un entorno profesional y por proporcionarme una experiencia enriquecedora y formativa que ha contribuido significativamente a mi crecimiento personal y profesional.

A toda mi familia, gracias por su amor, apoyo y por estar siempre a mi lado. Sus palabras de aliento y su confianza en mí han sido clave para superar los retos de esta etapa

A mis amigos, gracias por su amistad, apoyo y por estar siempre ahí cuando más lo necesitaba. Su compañía y comprensión han sido esenciales para mantener el equilibrio y la motivación durante este periodo.

Finalmente, gracias a todas aquellas personas que, de una u otra forma, han aportado su granito de arena en este camino. A todos, mi más profundo agradecimiento.

Tabla de contenido

1	Introducción	9
1.1	Motivación	9
1.2	Descripción	9
1.3	Objetivos	11
1.4	Estructura de la memoria	11
2	Especificaciones técnicas que implementar	13
2.1	Especificación técnica 3gpp_TS23288	13
2.1.1	Flujo de operaciones	13
2.1.2	Descripción de la funcionalidad	14
2.1.3	Procedimientos del NWDAF	15
2.2	Especificación técnica 3gpp_TS29520	20
2.2.1	Servicios del NWDAF	20
2.2.2	Nnwdafevenstsubscription	22
2.2.3	Nnwdafanalyticsinfo	26
2.2.4	Nnwdafdatamanagement	27
2.2.5	Nnwdafmmlmodelprovision	30
2.2.6	Opiniones sobre la especificación 29520	33
3	Planificación	35
3.1	Tareas del proyecto	35
3.2	Presupuesto	36
3.3	Riesgos	37
4	Diseño y desarrollo del prototipo	39
4.1	Alcance del diseño	39
4.2	Objetivos del proyecto	39
4.3	Tecnología usada	41
4.3.1	Visual Studio Code	41
4.3.2	GitHub	41
4.3.3	Python	41
4.3.4	Flask	41
4.3.5	OpenAPI Generator	42
4.3.6	MongoDB	42
4.3.7	Docker	42
4.3.8	Swagger	42
4.3.9	Postman	42
4.4	Requisitos del proyecto	42
4.4.1	OpenAPI Generator	43
4.4.2	Nginx	44
4.4.3	Docker	44
4.4.4	Estructura de los servicios y del proyecto	45
4.5	Arquitectura del NWDAF	46
4.6	Metodología	47

4.7	Generación de los servicios	48
4.8	Configuración del proxy inverso con NGINX.....	49
4.9	Configuración de Docker Compose	49
4.10	Gestión de suscripciones	51
4.10.1	Creación de la suscripción	51
4.10.2	Eliminación de la suscripción	51
4.10.3	Actualización de la suscripción.....	52
4.10.4	Problemas y limitaciones.....	53
4.11	Envío de notificaciones	54
4.12	Servicio Analytics Info	56
5	<i>Validación de servicios.....</i>	59
5.1	Plan de pruebas.....	59
5.2	Resultados.....	62
6	<i>Conclusiones.....</i>	65
7	<i>Abreviaturas</i>	66
8	<i>Referencias</i>	67

Figuras y tablas de la memoria

Figura 1. Arquitectura 5G con NWDAFs ^[16]	10
Figura 2. Recolección de datos de un NF.....	14
Figura 3. Arquitectura de peticiones del NWDAF.....	14
Figura 4. Suscripción/desuscripción del NWDAF.....	16
Figura 5. Petición de análisis al NWDAF.....	17
Figura 6. Recolección de datos del NWDAF con el servicio DataManagement.....	18
Figura 7. Suscripción a modelos de ML.....	19
Figura 8. Suscripción a eventos.....	22
Figura 9. Actualización de suscripción a eventos.....	24
Figura 10. Eliminación de suscripción a eventos.....	25
Figura 11. Envío de notificaciones del NWDAF.....	25
Figura 12. petición de información analítica al NWDAF.....	27
Figura 13. Petición de suscripción de datos al NWDAF.....	28
Figura 14. Actualización de suscripción a notificaciones de datos.....	29
Figura 15. Eliminación de suscripción a notificaciones de datos.....	29
Figura 16. Envío de notificaciones del NWDAF.....	30
Figura 17. Petición de suscripción de modelos ML al NWDAF.....	31
Figura 18. Actualización de suscripciones a notificaciones de modelos ML.....	32
Figura 19. Eliminación de suscripciones a notificaciones de modelos de ML.....	32
Figura 20. Envío de notificaciones del NWDAF.....	33
Figura 21. Arquitectura del NWDAF.....	46
Figura 22. NGINX como proxy inverso.....	49
Figura 23. Base de datos "eventDB".....	54
Tabla 1. Tareas del proyecto.....	36
Tabla 2. Coste total del proyecto.....	37
Tabla 3. Posibles riesgos del proyecto.....	37
Tabla 4. Soluciones planteadas para mitigar o corregir los riesgos.....	38
Tabla 5. Objetivos del proyecto.....	39
Tabla 6. Plan de pruebas.....	59

1 Introducción

1.1 Motivación

En los últimos años, hemos presenciado un crecimiento exponencial en la demanda de conectividad, impulsado por la adopción masiva de dispositivos inteligentes, el Internet de las Cosas (IoT) y la expansión de servicios digitales avanzados. Esta proliferación ha diversificado significativamente los servicios disponibles, abarcando desde el “streaming” de alta definición y las videoconferencias, hasta aplicaciones de realidad aumentada y virtual. Esta diversificación ha impuesto una presión sin precedentes sobre las infraestructuras de red existentes, las cuales deben lidiar con volúmenes de tráfico de datos cada vez mayores y más variados.

Esta situación ha subrayado la necesidad imperiosa de desarrollar soluciones innovadoras que puedan gestionar eficientemente el crecimiento del tráfico de datos. Las redes modernas no solo deben ser capaces de manejar el aumento del volumen de datos, sino que también deben anticiparse y adaptarse dinámicamente a las cambiantes demandas de los usuarios. La capacidad de respuesta en tiempo real es crucial para asegurar una experiencia de usuario fluida y satisfactoria.

En este contexto, la capacidad de la IA para analizar y comprender los patrones de uso de la red se convierte en un elemento fundamental. La analítica avanzada de datos de red mediante IA permite a los operadores identificar tendencias y patrones en el comportamiento de los usuarios, lo que es esencial para tomar decisiones estratégicas informadas. Esta información puede utilizarse para optimizar el rendimiento de la red, prever y mitigar posibles problemas antes de que afecten a los usuarios, y asegurar que los recursos de la red se utilicen de manera eficiente.

Además, la capacidad de análisis permite personalizar y mejorar la experiencia del usuario, adaptando los servicios a sus necesidades específicas. Por ejemplo, los operadores pueden priorizar el tráfico de aplicaciones críticas durante los picos de uso, mejorar la calidad del servicio en áreas con alta demanda y desarrollar nuevas ofertas basadas en los hábitos de uso de los clientes ^[10].

En resumen, el análisis y la comprensión de los patrones de uso de la red no solo son esenciales para garantizar la eficacia operativa de las infraestructuras de red, sino que también son clave para mejorar la experiencia del usuario y mantener la competitividad en un mercado digital en constante evolución. Las redes del futuro deben ser inteligentes, adaptativas y capaces de evolucionar junto con las demandas de sus usuarios.

1.2 Descripción

Este proyecto se desarrolla en el contexto de unas prácticas empresariales con Telefónica, bajo el marco del proyecto NAAS6G. NAAS6G representa una iniciativa esencial para la integración y gestión eficiente de capacidades avanzadas de red, siendo un pilar fundamental en la evolución hacia redes de próxima generación. Debido a la naturaleza innovadora y en desarrollo de este proyecto, los permisos para acceder a los repositorios y ejecutar ciertos procesos están estrictamente regulados y condicionados por las directrices del proyecto

Con la creciente necesidad de gestionar de manera eficiente el aumento exponencial del tráfico de datos y la diversificación de los servicios digitales, surge la importancia de las organizaciones que establecen los estándares y las especificaciones técnicas necesarias para asegurar un funcionamiento de las tecnologías de comunicación a nivel global. En este contexto, encontramos a la 3rd Generation Partnership Project (3GPP), una asociación internacional que agrupa a organizaciones de telecomunicaciones y desarrolla

especificaciones técnicas para los sistemas de comunicaciones móviles. 3GPP se dedica a coordinar y estandarizar las tecnologías móviles para asegurar la interoperabilidad y la compatibilidad entre diferentes redes y dispositivos alrededor del mundo [1].

Para abordar las crecientes demandas y desafíos de las redes modernas, la 3GPP ha creado el estándar para un componente esencial en la arquitectura de las redes 5G: el Network Data Analytics Function (NWDAF). Este componente es crucial para satisfacer la necesidad de análisis avanzado de datos de la red, proporcionando información valiosa que puede ser utilizada para la optimización y gestión eficiente de la red.

El estándar NWDAF es una parte integral de la arquitectura de las redes 5G. Su función principal es recopilar, procesar y analizar datos provenientes de diversas fuentes dentro de la red. Estos datos pueden incluir información sobre el tráfico de usuarios, el rendimiento de la red, la calidad del servicio, entre otros aspectos. Al recibir datos de múltiples fuentes como la Session Management Function (SMF) o el User Plane Function (UPF), el NWDAF consolida esta información para proporcionar un panorama completo y detallado del estado de la red [15].

5G Architecture with Distributed NWDAF

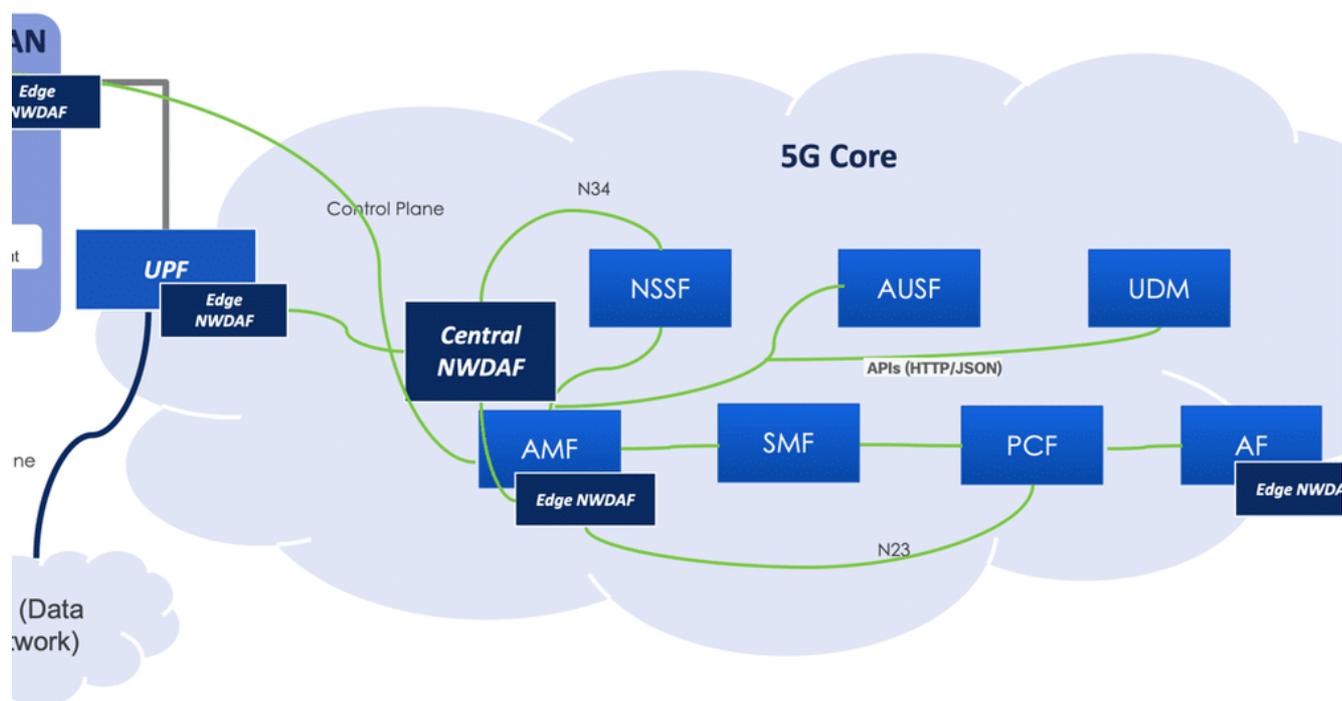


Figura 1. Arquitectura 5G con NWDAFs [16].

Para procesar estos datos y extraer información relevante, el NWDAF utiliza técnicas avanzadas de aprendizaje automático e inteligencia artificial. Estas técnicas permiten al NWDAF no solo analizar los datos en tiempo real, sino también hacer predicciones sobre el estado futuro de los diversos componentes de la red. Esto es crucial para anticipar posibles problemas y tomar medidas preventivas que aseguren el funcionamiento óptimo de la red.

La información proporcionada por el NWDAF puede ser utilizada por otros componentes de la arquitectura 5G, como la Policy Control Function (PCF) o la Access and Mobility Management Function (AMF). Estos componentes utilizan los datos analizados para tomar decisiones informadas sobre la asignación de recursos, la gestión del tráfico y la optimización general de la red. Además, los operadores de

red también pueden utilizar esta información para mejorar la planificación y el diseño de la red, así como para detectar y resolver incidentes de manera más eficiente ^[15].

En definitiva, el estándar del NWDAF define una función clave en la arquitectura de las redes 5G. Desempeña un papel fundamental en la recopilación y análisis de datos, lo que resulta esencial para la optimización y gestión eficiente de la red. Con su capacidad para procesar grandes volúmenes de datos y proporcionar información en tiempo real, el NWDAF es una herramienta indispensable para enfrentar los desafíos de las redes modernas y asegurar su funcionamiento continuo y eficiente

1.3 Objetivos

El objetivo principal de este trabajo es diseñar y desarrollar un prototipo del NWDAF, siguiendo las especificaciones y estándares establecidos por la 3GPP. Este prototipo se enfocará en dos tipos específicos de eventos: NF_LOAD y NETWORK_PERFORMANCE.

Además, como requisitos tenemos:

- Cumplimiento de Estándares: El prototipo debe adherirse rigurosamente a los estándares definidos por 3GPP para asegurar su interoperabilidad y compatibilidad con otros componentes y dispositivos de la red 5G.
- Eventos Enfocados: El prototipo se centrará en los eventos de NF_LOAD (carga de funciones de red) y NETWORK_PERFORMANCE (rendimiento de la red). Estos eventos son críticos para el análisis y la gestión eficiente de la red.
- Despliegue con Docker: Para facilitar el despliegue y la gestión de los servicios del NWDAF, el prototipo debe ser compatible con Docker. Esto permitirá una mayor flexibilidad y facilidad de implementación en diferentes entornos.

Como objetivo final, realizar una demostración del funcionamiento del prototipo para los eventos desarrollados, donde se crearán suscripciones a dichos eventos para poder recibir los datos analíticos y predictivos.

1.4 Estructura de la memoria

En esta sección, se detalla el contenido de cada apartado de la memoria del proyecto, abarcando desde la lectura de las especificaciones iniciales hasta la validación del prototipo final.

En el punto 2, se describe la planificación llevada a cabo a lo largo de la vida del proyecto. Este apartado incluye un desglose de las tareas realizadas, un análisis exhaustivo de los posibles riesgos y un desglose detallado del presupuesto total del proyecto. Esta planificación es crucial para asegurar el seguimiento adecuado y la mitigación de posibles inconvenientes.

El punto 3 aborda el análisis de las especificaciones técnicas. Aquí, se extrae y analiza toda la información relevante de los estándares 3GPP TS23288 y TS29520, que son esenciales para comprender el funcionamiento del prototipo y los servicios que lo componen. Este análisis es fundamental para garantizar que el desarrollo esté alineado con los estándares internacionales.

En el punto 4, se detalla el diseño y desarrollo del prototipo. Se explica el alcance y los objetivos del proyecto tras la lectura de los estándares, se menciona la tecnología utilizada, se describe la arquitectura

final y se detallan los pasos del desarrollo seguidos. Además, se discuten los riesgos identificados durante el proceso de desarrollo y las soluciones implementadas para mitigarlos.

El punto 5 se centra en la validación del prototipo NWDAF. Se describen las pruebas realizadas para validar el desarrollo de los servicios que componen el NWDAF, asegurando que cumplen con los objetivos establecidos en el punto anterior. Esta validación es crucial para confirmar la funcionalidad y eficiencia del prototipo.

Finalmente, los puntos 6 y 7 contienen las abreviaturas y referencias utilizadas a lo largo de la memoria. Este apartado es esencial para facilitar la comprensión del documento y proporcionar las fuentes de información consultadas.

2 Especificaciones técnicas que implementar

En esta sección se detalla toda la información relacionada con las especificaciones del prototipo del Network Data Analytics Function (NWDAF), incluyendo varios diagramas y casos de uso aportados por la 3GPP.

Las especificaciones de 3GPP son documentos técnicos que describen cómo deben funcionar los sistemas y redes de telecomunicaciones móviles. Estas especificaciones abarcan una amplia gama de aspectos, desde arquitectura de red hasta protocolos de comunicación, codificación de datos, gestión de recursos, seguridad y más. Son desarrolladas por grupos de trabajo dentro de la 3GPP, que incluyen a expertos de industria, proveedores de equipos, operadores de redes móviles y otros interesados en la tecnología móvil.

Las especificaciones de 3GPP son fundamentales para garantizar la interoperabilidad y el funcionamiento adecuado de equipos y redes móviles en todo el mundo. Además, permiten la introducción ordenada de nuevas tecnologías y servicios, asegurando que todos los sistemas compatibles con 3GPP puedan comunicarse de manera efectiva y eficiente, independientemente del fabricante o proveedor de servicios.

2.1 Especificación técnica 3gpp_TS23288

El estándar define la arquitectura para redes 5G que soportan los servicios del Network Data Analytics Function (NWDAF) en el núcleo de la red 5G. Esta arquitectura se enfoca en la optimización y gestión de la red mediante el análisis avanzado de datos, permitiendo la toma de decisiones informadas en tiempo real. Todas las figuras que se muestran en este punto pertenecen a la especificación técnica TS23288.^[13]

2.1.1 Flujo de operaciones

El Network Data Analytics Function (NWDAF) desempeña un papel crucial en la arquitectura de redes modernas al interactuar con diversas entidades para cumplir diferentes propósitos estratégicos. Las interacciones planificadas para este proyecto abarcan varios aspectos fundamentales:

- **Recolección de Datos Basada en Suscripción a Eventos:** El NWDAF establece interacciones con entidades como el Access and Mobility Management Function (AMF), Session Management Function (SMF), Policy Control Function (PCF), Unified Data Management (UDM), Network Slice Anchor Connectivity Function (NSACF), Application Function (AF) y Operations, Administration, and Maintenance (OAM). A través de interfaces específicas, el NWDAF se suscribe a eventos emitidos por estas entidades. Por ejemplo, puede suscribirse a eventos relacionados con cambios en el contexto de sesión, actualizaciones de políticas de red, gestión de datos de usuario, entre otros. Esta suscripción permite al NWDAF recopilar datos en tiempo real para análisis posteriores y optimización de servicios.
- **Recepción de Información sobre los NFs de la Red Mediante el NRF:** Utilizando el Network Repository Function (NRF), el NWDAF accede a información detallada sobre los Network Functions (NFs) disponibles en la red. Esto incluye capacidades, ubicaciones y características específicas de los NFs necesarias para realizar análisis y tomar decisiones informadas. La integración con NRF garantiza que el NWDAF tenga una visión completa de la topología de la red y pueda adaptarse dinámicamente a los cambios en la infraestructura de servicios.

- **Provisión de Análisis Bajo Demanda:** Además de la recolección continua de datos, el NWDAF ofrece capacidades de análisis bajo demanda. Esto implica que las funciones de red y otras entidades pueden solicitar al NWDAF análisis específicos basados en los datos recogidos previamente. Por ejemplo, podrían solicitarse análisis de tráfico por aplicación, evaluaciones de calidad de servicio (QoS) para sesiones particulares, o evaluaciones de carga de red para optimizar la asignación de recursos. Esta flexibilidad permite una respuesta ágil a las necesidades operativas y de optimización de la red.
- **Provisión de Datos en Crudo:** Junto con los análisis procesados, el NWDAF tiene la capacidad de proporcionar datos en su forma cruda a otras entidades de la red. Estos datos sin procesar son útiles para análisis adicionales por parte de las funciones de red o aplicaciones externas que requieran un acceso detallado para mejorar el rendimiento, garantizar la seguridad de la red o optimizar la eficiencia de los recursos.

Estas interacciones se facilitan mediante interfaces estandarizadas y protocolos definidos, asegurando una comunicación eficiente y efectiva entre el NWDAF y las diversas entidades de la red. Esto no solo mejora la gestión de la red y la optimización de recursos, sino que también impulsa la entrega de servicios mejorados y personalizados a los usuarios finales, garantizando así una experiencia de red óptima y confiable en entornos modernos de telecomunicaciones.

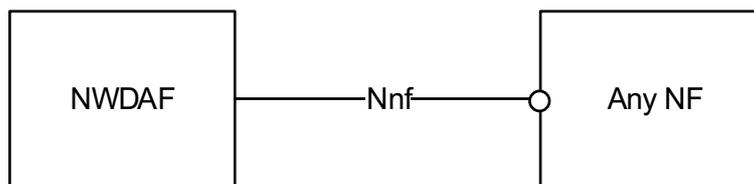


Figura 2. Recolección de datos de un NF

La interfaz Nnf se define específicamente para permitir al NWDAF solicitar y gestionar suscripciones a la entrega de datos relacionados con un contexto específico, así como para cancelar dichas suscripciones cuando sea necesario. Esta interfaz facilita una comunicación estructurada y eficiente entre el NWDAF y otras entidades de la red que manejan información relevante para análisis y operaciones.

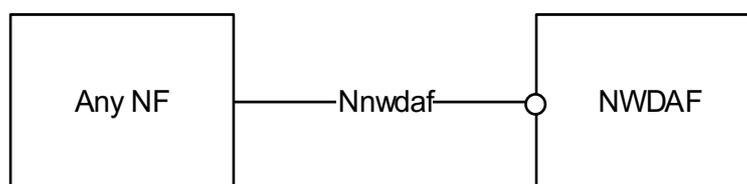


Figura 3. Arquitectura de peticiones del NWDAF

La interfaz NnwdaF está específicamente diseñada para las funciones de red del núcleo 5G, con el propósito de facilitar la suscripción a la entrega de análisis de red para un contexto específico, así como para cancelar estas suscripciones según sea necesario. Este componente será el foco principal de desarrollo en el prototipo, asegurando una comunicación efectiva entre el NWDAF y las funciones de red para la gestión dinámica de datos y análisis.

2.1.2 Descripción de la funcionalidad

El Network Data Analytics Function (NWDAF) desempeña un rol fundamental en la arquitectura 5G al

proporcionar análisis predictivos y estadísticos a otros Network Functions (NFs). Aquí se detallan las funciones principales del NWDAF y su integración con los NFs del núcleo 5G:

2.1.2.1 Funciones Lógicas del NWDAF

Analytics Logical Function (AnLF): Esta función es responsable de realizar inferencias utilizando modelos predictivos entrenados. El AnLF expone servicios como `Nnwdaf_EventsSubscription` y `Nnwdaf_AnalyticsInfo`, permitiendo al NWDAF ofrecer análisis avanzados basados en datos recopilados de los NFs. Por ejemplo, puede realizar predicciones sobre la congestión de red futura o el comportamiento del usuario basado en patrones históricos.

Model Training Logical Function (MTLF): Esta función se encarga del entrenamiento continuo de modelos utilizando técnicas de Machine Learning. A través del MTLF, el NWDAF mejora la precisión de sus modelos predictivos, adaptándolos a cambios en el tráfico de red y comportamiento del usuario. Los servicios expuestos por el MTLF están relacionados con la gestión y optimización de los procesos de entrenamiento de modelos.

2.1.2.2 Configuración del NWDAF

Un NWDAF puede implementar tanto AnLF como MTLF, o solo una de estas funciones lógicas, dependiendo de los requisitos y capacidades específicas de la red 5G. Esta configuración flexible asegura que el NWDAF pueda adaptarse a diferentes escenarios operativos y necesidades de análisis dentro de la red.

Es importante destacar que el entrenamiento y almacenamiento de los modelos utilizados por el NWDAF están fuera del alcance del estándar 5G. Esto significa que estos procesos son gestionados externamente, permitiendo al NWDAF enfocarse en la inferencia y el análisis de datos en tiempo real.

2.1.2.3 Tipo de Información Analizada

La información proporcionada por el NWDAF puede ser tanto predictiva como estadística, derivada de eventos pasados y patrones identificados en los datos de la red. Esto proporciona a los consumidores de los servicios del NWDAF, como los NFs del núcleo 5G, una base sólida para la toma de decisiones informadas y estratégicas.

2.1.2.4 Uso de Datos por Parte de los Consumidores

Los NFs del núcleo 5G y otros consumidores de los servicios del NWDAF tienen la libertad de decidir cómo utilizan los datos emitidos por el NWDAF. Pueden integrar los análisis proporcionados para optimizar operaciones, mejorar la calidad de servicio, o desarrollar nuevos servicios basados en datos analíticos avanzados.

2.1.2.5 Organización Jerárquica y Compartición de Datos

La arquitectura del NWDAF está diseñada para permitir la organización de múltiples instancias en una jerarquía. Esto facilita que diferentes NWDAFs puedan compartir datos entre sí para generar análisis más completos y coordinados. Esta capacidad de organización jerárquica promueve una gestión eficiente de datos y análisis en entornos de red complejos y distribuidos, asegurando la coherencia y la precisión de los insights derivados de los datos de red.

2.1.3 Procedimientos del NWDAF

El estándar 23288 especifica todos los procedimientos que las funciones del NWDAF deben soportar,

permitiéndoles comunicarse directamente con los servicios consumidores o a través de otras funciones de red como el NEF o el DCCF.

Dado que este proyecto se enfoca únicamente en el desarrollo y diseño del NWDAF, las funciones intermedias utilizan los servicios expuestos por el NWDAF de la misma manera que otros servicios consumidores. Por lo tanto, no entraremos en detalle en las interacciones de estas funciones con el NWDAF.

A continuación, se detallan algunos de los procedimientos que debe soportar el NWDAF según el estándar 23288:

2.1.3.1 Suscripción/desuscripción de análisis

El procedimiento de suscripción/desuscripción de análisis es utilizado por los consumidores del NWDAF para gestionar su interacción con los servicios analíticos ofrecidos. Aquí se explica cómo funciona este proceso:

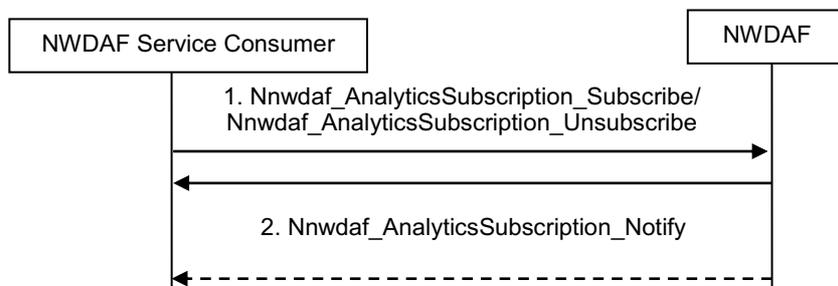


Figura 4. Suscripción/desuscripción del NWDAF

1. Suscripción y Gestión de Información Analítica:

El consumidor del NWDAF se suscribe o desuscribe para recibir información analítica utilizando el servicio de AnalyticsSubscription. Cuando se recibe esta suscripción, el NWDAF evalúa si es necesario recolectar nuevos datos para satisfacer las necesidades específicas de análisis.

2. Notificación de Información Analítica:

El NWDAF notifica al consumidor la información analítica solicitada utilizando la suscripción establecida previamente. Esta notificación puede ser periódica según una programación predeterminada o desencadenada por eventos específicos relevantes para el análisis.

La notificación se repetirá según lo acordado en la suscripción, asegurando que el consumidor reciba actualizaciones regulares y pertinentes hasta que decida cancelar la suscripción.

2.1.3.2 Petición de análisis

Este procedimiento es utilizado por un consumidor para pedir y obtener del NWDAF información analítica.

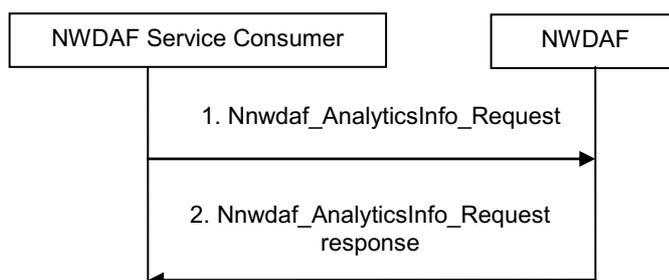


Figura 5. Petición de análisis al NWDAF

1. Solicitud de Información Analítica por el Consumidor:

El consumidor del NWDAF solicita información analítica utilizando el servicio AnalyticsInfo. Cuando esta solicitud es recibida por el NWDAF, este procede a evaluar si es necesario recolectar nuevos datos para satisfacer los requisitos específicos del análisis solicitado.

2. Respuesta del NWDAF con la Información Analítica Solicitada:

El NWDAF responde al consumidor proporcionando la información analítica solicitada. Esta respuesta incluye los resultados de los análisis basados en los datos recopilados y procesados, cumpliendo así con las expectativas del consumidor en términos de precisión y relevancia.

2.1.3.3 Recolección de datos de NFs

Para que el NWDAF pueda suscribirse o desuscribirse de las funciones de red del core 5G y recibir notificaciones con los datos de sus eventos, utiliza los siguientes servicios especificados en el estándar TS 23.502:

Service producer	Service	Reference in TS 23.502 [3]
AMF	Namf_EventExposure (NOTE 3)	5.2.2.3 5.2.3.5
SMF	Nsmf_EventExposure (NOTE 3)	5.2.8.3 5.2.3.5
PCF	Npcf_EventExposure (for a group of UEs or any UE) Npcf_PolicyAuthorization_Subscribe (for a specific UE)	5.2.5.7
UDM	Nudm_EventExposure	5.2.3.5
NEF	Nnef_EventExposure	5.2.6.2
AF	Naf_EventExposure	5.2.19.2
NRF	Nnrf_NFDiscovery	5.2.7.3
	Nnrf_NFManagement	5.2.7.2
NSACF	Nnsacf_SliceEventExposure	5.2.21.4

Table 1. Servicios consumidos por el NWDAF para la obtención de datos

Estos servicios permiten al NWDAF establecer suscripciones a eventos específicos emitidos por cada función de red mencionada. A través de estos mecanismos, el NWDAF puede recibir datos actualizados y relevantes que le permiten realizar análisis avanzados y tomar decisiones informadas dentro de la infraestructura 5G.

El siguiente procedimiento es utilizado por un consumidor para recibir datos históricos del servicio DataManagement del NWDAF.

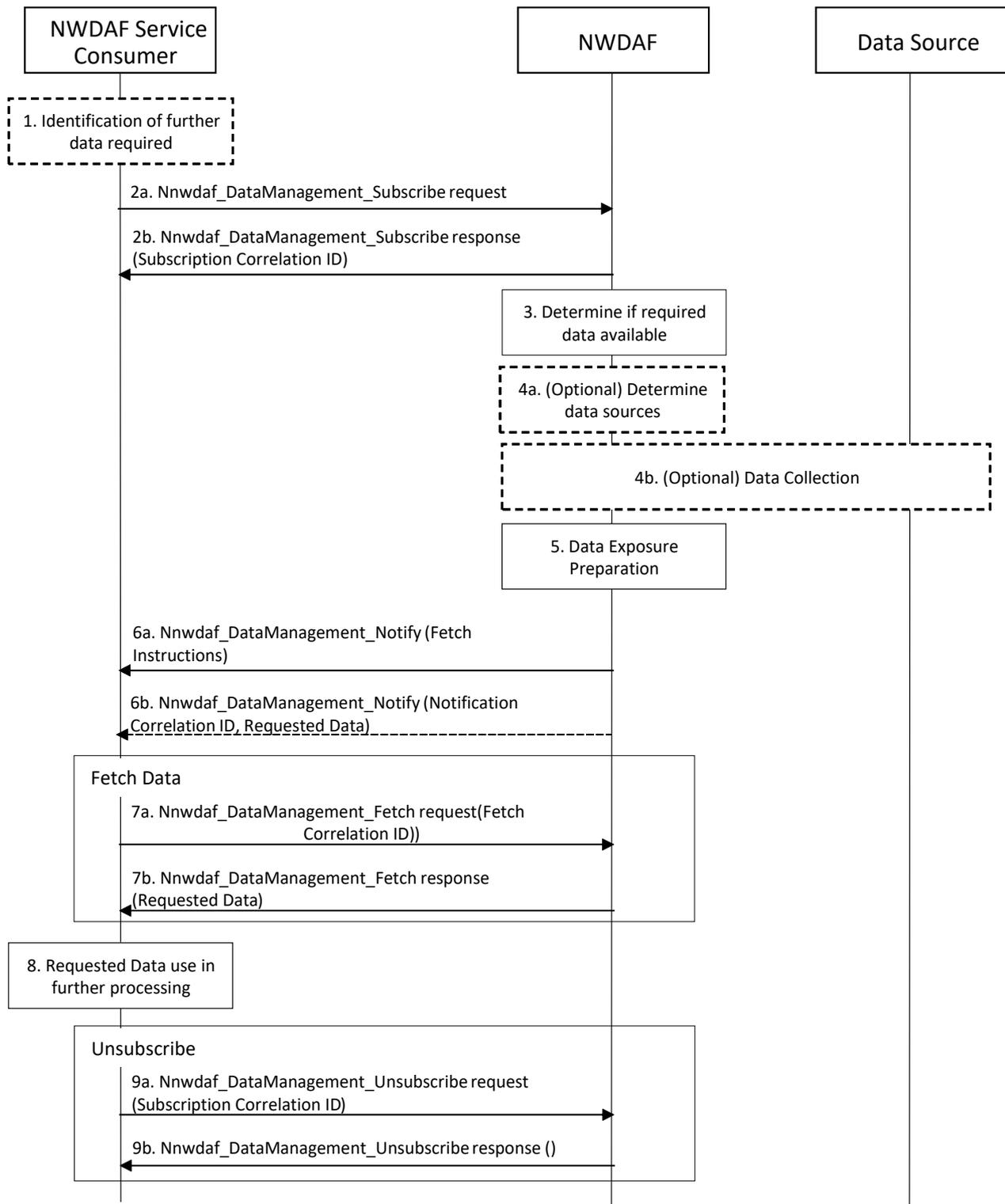


Figura 6. Recolección de datos del NWDaf con el servicio DataManagement

1. El consumidor identifica que se requieren datos adicionales del NWDaf para realizar alguna operación.
2. El consumidor del servicio NWDaf invoca el servicio Nnwdaf_DataManagement_Subscribe para solicitar los datos requeridos. La solicitud incluye especificaciones de datos, instrucciones de formateo y procesamiento de datos, y la dirección del objetivo de notificación junto con un ID de correlación de notificación.

3. El NWDAF comprueba si los datos requeridos están disponibles o no para poder realizar correctamente la petición.
4. Si los datos pedidos no están disponibles, el NWDAF podrá requerirlos usando cualquier mecanismo disponible que posea.
5. Obtenidos los datos en crudo, el NWDAF genera la respuesta a la petición con la información requerida.
6. El NWDAF envía la notificación con la información requerida dependiendo del flag especificado en la petición.
7. El consumidor podrá requerir la notificación si el flag del paso anterior no pedía en envío inmediato de la información.
8. El consumidor del NWDAF utilizará la información obtenida para realizar las operaciones que vea oportunas.
9. Por último, el consumidor podrá eliminar la suscripción a datos del NWDAF.

Este procedimiento asegura que el consumidor pueda acceder de manera eficiente y efectiva a datos en crudo almacenados por el NWDAF.

2.1.3.4 Suscripción a modelos de ML

En esta versión de la especificación, un NWDAF que contiene AnLF está configurado localmente con un conjunto de IDs de NWDAFs que contienen MTLF, así como con los ID(s) de análisis admitidos por cada NWDAF que contiene MTLF para recuperar modelos de aprendizaje automático entrenados.

Cuando un NWDAF que contiene AnLF necesita acceder a un NWDAF que contiene MTLF dentro del conjunto de IDs configurados, puede utilizar el descubrimiento de NWDAF si es necesario. Por otro lado, un NWDAF que contiene MTLF puede determinar que es necesario un entrenamiento adicional para un modelo de aprendizaje automático existente cuando recibe la suscripción o la solicitud del modelo de aprendizaje automático.

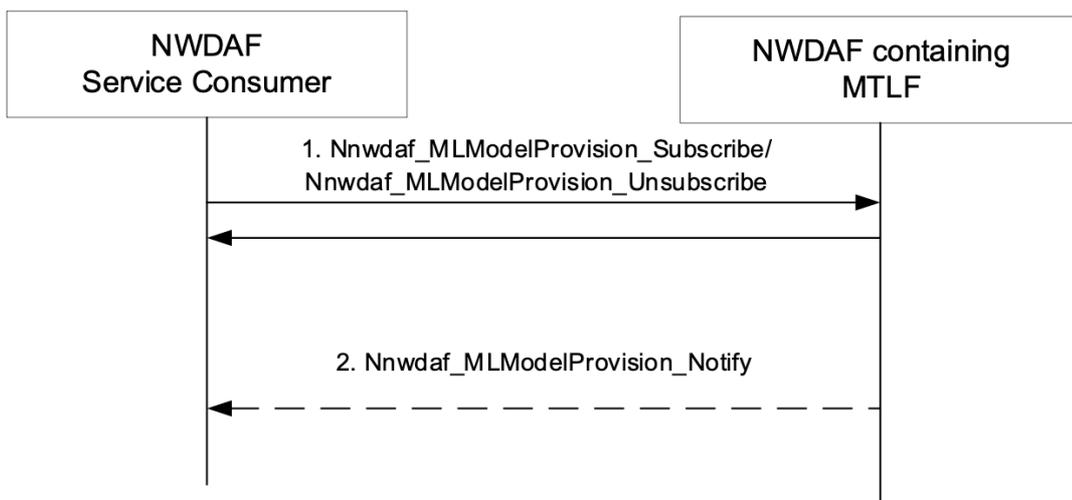


Figura 7. Suscripción a modelos de ML

En este escenario, el consumidor del servicio NWDAF, que típicamente es un NWDAF que contiene AnLF, se encarga de gestionar suscripciones, modificaciones o cancelaciones para modelos de aprendizaje automático entrenados asociados con identificadores específicos de análisis. Esto se logra mediante la invocación de operaciones de servicio como `Nnwdaf_MLModelProvision_Subscribe` o

Nnwdaf_MLModelProvision_Unsubscribe.

Cuando se recibe una solicitud de suscripción, se realiza un proceso crítico de toma de decisiones. En primer lugar, se evalúa si un modelo de ML entrenado existente puede cumplir con los requisitos de la suscripción. Alternativamente, si se determina que es necesario un entrenamiento adicional, el NWDAF que contiene MTLF inicia la recolección de datos desde diversas fuentes. Esta recolección de datos es fundamental para generar o re-entrenar el modelo de ML para satisfacer las necesidades de la suscripción.

Cuando el consumidor del servicio NWDAF se suscribe o modifica suscripciones para modelos de ML entrenados asociados con Analytics IDs, el NWDAF que contiene MTLF informa al consumidor sobre la disponibilidad de la información del modelo de ML entrenado enviando una petición a la url suministrada en la suscripción, adjuntando detalles como las direcciones de archivo de los modelos de ML entrenados

2.1.3.5 Conclusiones de la especificación 23288

El estándar 23288 proporciona una descripción exhaustiva de los flujos de trabajo y las interacciones entre el NWDAF y las diversas entidades de la red en el contexto del núcleo 5G. Esta especificación es esencial para comprender cómo opera y se utiliza el NWDAF dentro del ecosistema de red, permitiendo a los desarrolladores y operadores de red comprender claramente cómo integrar y aprovechar esta función para diversos propósitos analíticos y de gestión de la red.

Al tener esta especificación como referencia, es posible anticipar los objetivos y desafíos del proyecto que involucra el NWDAF. Por ejemplo, se destacan la capacidad de establecer comunicación bidireccional entre el NWDAF y otras funciones de red, como AMF, SMF, PCF, UDM, AF, entre otras, para la recolección y análisis de datos en tiempo real. Además, la capacidad de coordinación entre múltiples instancias de NWDAF para el tratamiento de modelos de ML es crucial, permitiendo escalar y distribuir cargas de trabajo analítico de manera eficiente dentro de la red 5G.

No obstante, el estándar también revela áreas que pueden necesitar mayor claridad o desarrollo futuro. Por ejemplo, el servicio de ML Model Provision podría beneficiarse de actualizaciones adicionales o explicaciones más detalladas en futuras versiones del estándar o en especificaciones complementarias, como TS29520. Esta última especificación se centra en detallar de manera más profunda los servicios específicos que componen el NWDAF, lo cual promete brindar una comprensión más completa y detallada de su funcionalidad y su integración en el entorno de red 5G.

2.2 Especificación técnica 3gpp_TS29520

El estándar TS29520 detalla los servicios de la interfaz Nnwdaf, utilizada por el NWDAF para ofrecer análisis y modelos de aprendizaje automático. Todas las figuras y tablas que se muestran en este punto pertenecen a la especificación técnica TS29520.^[14]

2.2.1 Servicios del NWDAF

El NWDAF ofrece una serie de servicios fundamentales que permiten a las funciones de red (NFs) consumidoras aprovechar al máximo sus capacidades analíticas:

- Nnwdaf_EventsSubscription: Este servicio permite a las NFs suscribirse para recibir notificaciones sobre unos eventos específicos. Estas notificaciones contienen información analítica detallada que puede ser crucial para la optimización y la toma de decisiones en tiempo real. Además, se facilita la transferencia de estas suscripciones a otro NWDAF, asegurando la continuidad del servicio ante cambios de infraestructura.

- Nnwdaf_AnalyticsInfo: Aquí, las NFs pueden solicitar análisis específicos y detallados generados por el NWDAF. Estos análisis abarcan desde métricas de rendimiento hasta patrones de tráfico, proporcionando insights valiosos que pueden utilizarse para mejorar la eficiencia operativa y la calidad del servicio.
- Nnwdaf_DataManagement: Este servicio permite a las NFs gestionar eficazmente los datos generados y procesados por el NWDAF. Las NFs pueden suscribirse para recibir notificaciones automáticas sobre la detección de eventos críticos o acceder de manera estructurada a datos históricos y en tiempo real gestionados por el NWDAF. Esta capacidad facilita la implementación de estrategias proactivas basadas en datos y el monitoreo continuo de la red.
- Nnwdaf_MLModelProvision: En este servicio, las NFs tienen la oportunidad de suscribirse para recibir notificaciones cuando el NWDAF despliega nuevos modelos de aprendizaje automático. Estos modelos están diseñados para mejorar la precisión de las predicciones y optimizar la respuesta ante situaciones variables en la red, ofreciendo una mayor automatización y eficiencia operativa.

Los consumidores de cada servicio NF varían según el endpoint utilizado, pudiendo ser funciones de red objetivo como el AMF, SMF o UPF, o las encargadas de la recolección de datos como el DCCF, MFAF y ADRF.

Además, para facilitar la interacción entre NWDAFs, un NWDAF puede utilizar todos los servicios de otro NWDAF para la obtención de datos y modelos.

A continuación, se presenta una tabla que resume los endpoints de cada servicio junto con ejemplos de posibles consumidores:

Servicio	Método	Endpoint	Consumidor
Nnwdaf_EventsSubscription	POST	/subscriptions	PCF, NSSF, AMF, SMF, NEF, AF, OAM, NWDAF, DCCF
	PUT	/subscriptions/{subscriptionId}	
	DELETE	/subscriptions/{subscriptionId}	
	POST	/transfers	NWDAF
	PUT	/transfers/{transferId}	
	DELETE	/transfers/{transferId}	
Nnwdaf_AnalyticsInfo	GET	/analytics?query_parameters	PCF, NSSF, AMF, SMF, NEF, AF, OAM, NWDAF, DCCF
	GET	/context?query_parameters	NWDAF
Nnwdaf_DataManagement	POST	/subscriptions	NWDAF, DCCF, MFAF, ADRF
	PUT	/subscriptions/{subscriptionId}	
	DELETE	/subscriptions/{subscriptionId}	
	POST	{fetchUri}	
Nnwdaf_MLModelProvision	POST	/subscriptions	NWDAF
	PUT	/subscriptions/{subscriptionId}	
	DELETE	/subscriptions/{subscriptionId}	

Table 2. Servicios del NWDAF

En el prototipo que se desarrollará, se seguirá el estándar mencionado, enfocándose específicamente en dos eventos clave: NF_LOAD y QOS_SUSTAINABILITY. Estos eventos fueron seleccionados debido a su relevancia y amplio impacto dentro del contexto de los servicios proporcionados por el NWDAF.

Todos los modelos de datos utilizados en este proyecto son proporcionados por 3GPP. Para facilitar la comprensión y visualización de los atributos utilizados, se ha creado un Swagger donde se pueden explorar todos estos detalles.

Este enfoque permite una implementación más focalizada y eficiente, asegurando que el prototipo aborde de manera efectiva los aspectos críticos de NF_LOAD y QOS_SUSTAINABILITY según los estándares establecidos por 3GPP.

2.2.2 Nnwdaf_EventSubscription

Este servicio es usado por un NF consumidor para suscribirse a la obtención notificaciones de eventos del NWDAF.

2.2.2.1 Suscripción a eventos

La siguiente figura ilustra el proceso mediante el cual un NF envía una solicitud para suscribirse a notificaciones de eventos del NWDAF.



Figura 8. Suscripción a eventos

El NF realiza una petición detallada, incluyendo en el cuerpo de la solicitud los siguientes datos esenciales:

- notificationURI: Especifica la dirección a la cual el NF desea recibir las notificaciones generadas por el NWDAF. Esta URI puede ser un endpoint HTTP/HTTPS u otro mecanismo de comunicación compatible.
- eventSubscriptions: Consiste en descripciones específicas de los eventos a los cuales el NF desea suscribirse. Cada descripción incluye los siguientes elementos:
 - o event: Identificación del evento al cual se desea suscribir. Este atributo especifica el tipo de evento, como por ejemplo NF_LOAD o QOS_SUSTAINABILITY.
 - o notificationMethod: Indicación del método de notificación deseado. Puede ser "PERIODIC" para recibir notificaciones en intervalos regulares especificados por el NF, utilizando el atributo adicional "repetitionPeriod" para definir la frecuencia de las notificaciones.

- repetitionPeriod: Este atributo es requerido si se elige el método de notificación "PERIODIC". Define el intervalo de tiempo entre cada notificación periódica, permitiendo al NF especificar la frecuencia con la cual desea recibir actualizaciones sobre el evento.
- otros atributos opcionales: La suscripción puede incluir otros parámetros adicionales según sea necesario, como una ventana de tiempo durante la cual la suscripción es válida o el nivel preferido de precisión de los análisis proporcionados por el NWDAF.

Las descripciones de los eventos a los que te suscribes también pueden requerir datos adicionales y opcionales. Para los eventos en los que nos queremos centrar tenemos lo siguiente:

Para el evento NF_LOAD es necesario especificar los siguientes atributos obligatorios:

- Identificación del UE objetivo: Se puede utilizar uno de los siguientes atributos para especificar el UE al cual aplica la suscripción: "supis", "anyUe", o "tgtUe".
- nfLoadLvlThds: Este atributo es necesario si el método de notificación ("notificationMethod") es "THRESHOLD" o si el método de notificación de la solicitud de evento ("evtReq") es "ONE_EVENT_DETECTION"

Como atributos opcionales, se pueden incluir:

- snssais: Lista de IDs de las instancias de NFs a las cuales aplica la suscripción si se identifican todos los UEs.
- nfSetIds: Lista de IDs de los NFs a los cuales aplica la suscripción si se identifican todos los UEs.
- nfTypes: Tipos de las instancias de NF a las cuales aplica la suscripción.
- matchingDir: Dirección a la cual se dirige el evento si se utiliza "nfLoadLvlThds".
- networkArea: Área de interés a la cual aplica la suscripción.

Para suscribirse al evento QOS_SUSTAINABILITY, se deben especificar los siguientes atributos obligatorios:

- networkArea: Área de red en la cual aplica la suscripción.
- qosRequ: Requerimientos de QoS descritos en este atributo.
- qosFlowRetThds: Threshold para el tipo GBR de 5QI, o

- ranUeThrouThds: Threshold para el tipo non-GBR de 5QI, si el método de notificación es "THRESHOLD" o si el método de notificación de la solicitud de evento ("evtReq") es "ONE_EVENT_DETECTION".
- tgtUe: Identificación de los UEs objetivo utilizando el atributo "anyUe" en "tgtUe".

Además, se pueden incluir los siguientes atributos opcionales:

- snssais: Identificación de la slice de red.
- matchingDir: Dirección a la cual se dirige el evento si se utiliza "qosFlowRetThds" o "ranUeThrouThds".

Cuando el NWDAF recibe esta petición, procede a crear la suscripción asignándole un ID único y almacenándola para su gestión.

Si la petición se procesa correctamente, el NWDAF responde con el código de estado "201 Created". El cuerpo de la respuesta contiene la representación detallada de la suscripción creada, mientras que en la cabecera de la respuesta se incluye el campo "Location". Este campo "Location" proporciona la URL donde se puede acceder a la suscripción recién creada, siguiendo el formato:

{apiRoot}/nnwdaf-eventssubscription/v1/subscriptions/{subscriptionId}.

Este enfoque asegura una interacción efectiva entre el NF consumidor y el NWDAF, facilitando la gestión y el acceso a las suscripciones de eventos dentro del entorno de la red.

2.2.2.2 Actualización de suscripción a eventos

La siguiente figura ilustra el escenario en el cual un NF consumidor envía una petición al NWDAF para actualizar su suscripción:

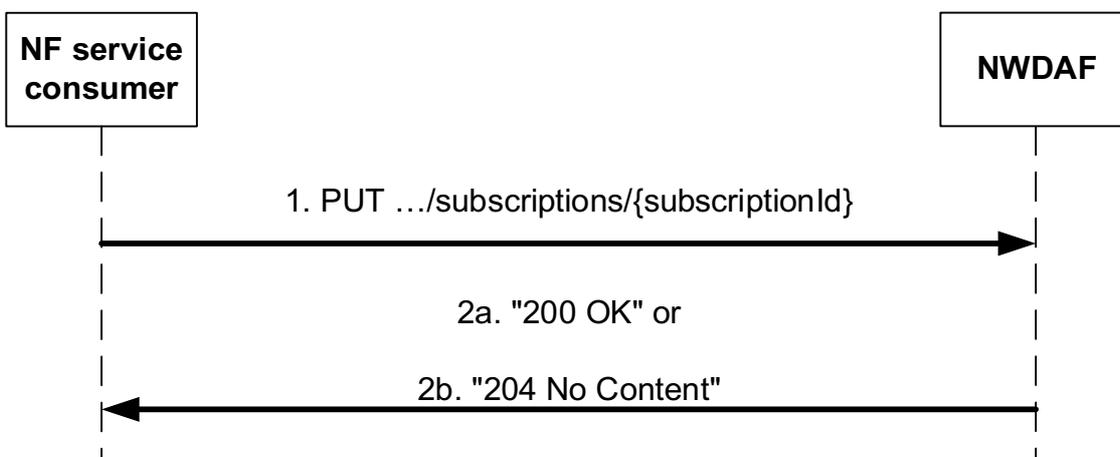


Figura 9. Actualización de suscripción a eventos

El NF consumidor realizará la petición a la URI proporcionada en la cabecera de la respuesta de su

suscripción (campo "Location"), la cual contiene el subscriptionId necesario para identificar la suscripción que se desea modificar. En esta petición, el NF consumidor incluirá el cuerpo con los nuevos detalles de la suscripción.

Una vez que el NWDAF reciba esta petición, procederá a actualizar la suscripción correspondiente al subscriptionId especificado. Si la operación se realiza con éxito, el NWDAF responderá con el código de estado "200 OK". En caso de que no se requiera devolver ningún contenido adicional, el NWDAF puede responder con el código "204 No Content".

2.2.2.3 Eliminación de suscripción a eventos

La siguiente figura ilustra el escenario en el cual un NF consumidor envía una petición al NWDAF para eliminar una suscripción a eventos:

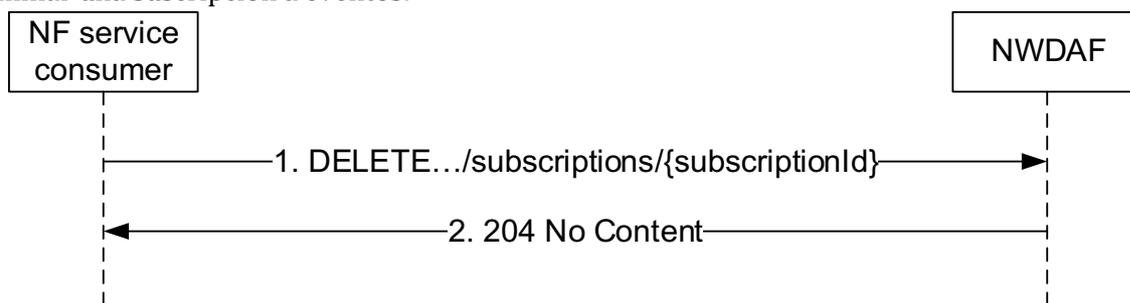


Figura 10. Eliminación de suscripción a eventos

El NF consumidor realizará la petición a la URI proporcionada en la cabecera de la respuesta de su suscripción (campo "Location"), la cual contiene el subscriptionId necesario para identificar la suscripción que se desea eliminar.

Una vez que el NWDAF reciba esta petición, procederá a eliminar la suscripción correspondiente al subscriptionId especificado. Si la operación se realiza con éxito y no se requiere devolver ningún contenido adicional, el NWDAF responderá con el código de estado "204 No Content".

2.2.2.4 Notificación de una suscripción a eventos

La siguiente figura ilustra el escenario en el cual el NWDAF envía una notificación al NF consumidor sobre sus suscripciones a eventos:

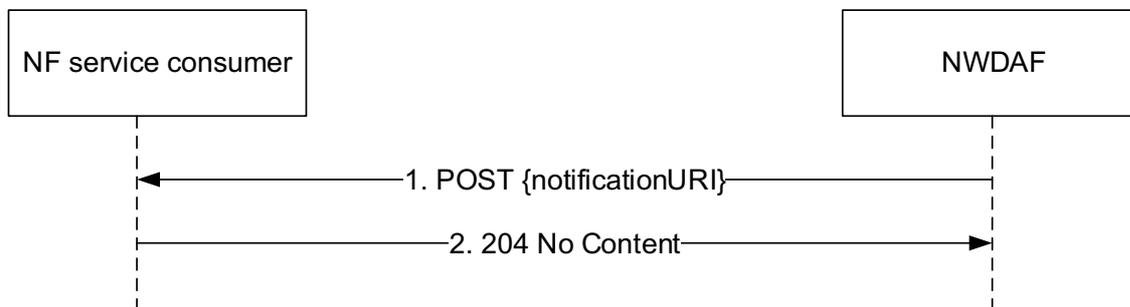


Figura 11. Envío de notificaciones del NWDAF

El NWDAF enviará la notificación a la dirección indicada en el atributo "notificationURI" de la suscripción del NF consumidor. Esta acción se repetirá periódicamente cada X segundos, según lo especificado en el atributo "repPeriod" de la suscripción. Además, puede incluir datos históricos o predictivos utilizando el atributo "offsetPeriod".

Cada notificación contendrá análisis de los eventos suscritos, y cada análisis estará estructurado con los siguientes atributos:

- "event": Indica el tipo de evento al cual hace referencia el análisis.
- "nfLoadLevelInfos": Contiene la información específica del evento de LOAD_LEVEL, si está presente en la notificación.
- "qosSustainInfos": Contiene la información específica del evento de QOS_SUSTAINABILITY, si está presente en la notificación.

Después de recibir la notificación, si la petición de envío ha sido exitosa, el NF consumidor deberá almacenar la notificación y responder al NWDAF con el código de estado "204 No Content".

2.2.2.5 Transferencia de suscripciones a otros NWDAFs

Este servicio facilita al NWDAF la capacidad de solicitar a otro NWDAF el traspaso de la suscripción de un NF. En escenarios donde múltiples NWDAFs están desplegados en una red, este servicio permite la transferencia de responsabilidades de suscripción entre entidades para una gestión más eficiente y distribuida.

Sin embargo, para este proyecto específico, se ha decidido no implementar este servicio. La razón principal es que se presupone la existencia de un único NWDAF desplegado en el entorno de desarrollo. Esto simplifica la arquitectura y reduce la complejidad operativa al concentrar todas las funciones de análisis y gestión de suscripciones en una única instancia del NWDAF.

Al omitir el desarrollo de este servicio, el proyecto puede centrarse de manera más efectiva en los objetivos prioritarios, como la implementación y optimización de los análisis específicos de eventos como NF_LOAD y QOS_SUSTAINABILITY dentro del estándar TS29520.

2.2.3 Nnwdaf_AnalyticsInfo

El servicio Nnwdaf_AnalyticsInfo facilita a diversos NF la capacidad de solicitar y recibir una amplia gama de información relacionada con el análisis de eventos,

2.2.3.1 Petición de análisis

La siguiente figura muestra el escenario donde un NF consumidor envía una petición al NWDAF para obtener información analítica.

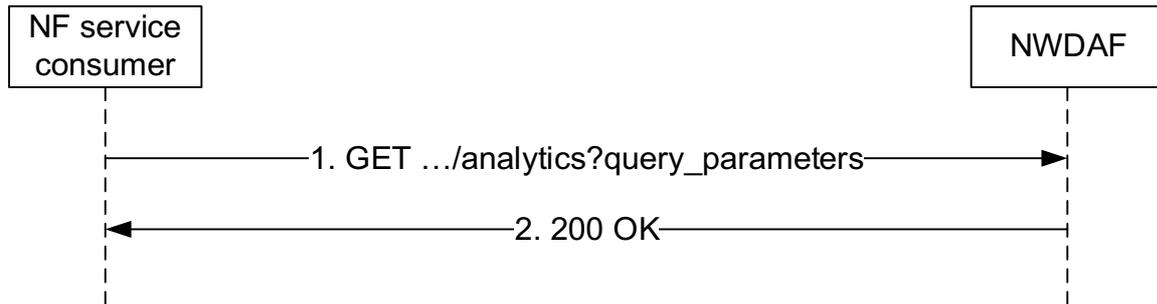


Figura 12. petición de información analítica al NWDAF

El NF realiza la petición especificando los parámetros necesarios en la query de la solicitud, dependiendo del evento establecido en el campo "event-id". Algunos de estos parámetros son los siguientes:

- "ana-req": Contiene los requisitos comunes como la ventana de tiempo para aplicar la solicitud o el nivel de precisión del análisis.
- "event-filter": Incluye atributos opcionales específicos para cada evento:
 - Para el evento NF_LOAD:
 - "nfInstanceIds": Lista de IDs de las instancias de NFs a las que se aplica la solicitud si se identifican todos los UEs.
 - "nfSetIds": Lista de IDs de los NFs a los que se aplica la solicitud si se identifican todos los UEs.
 - "nfTypes": Tipos de las instancias de NF a las que se aplica la solicitud.
 - "snssais": Identificación de la slice de red.
 - "networkArea": Área de interés a la que se aplica la solicitud.
 - Para el evento QoS_SUSTAINABILITY:
 - "networkArea": Área de red donde se aplica la suscripción.
 - "qosRequ": Requisitos de QoS descritos.
 - "snssais": Identificación de las slices de red.
- "tgt-ue": Contiene atributos necesarios y opcionales dependiendo del evento:
 - Para el evento NF_LOAD:
 - Identifica el UE objetivo utilizando "supis" o "anyUe".
 - Para el evento QoS_SUSTAINABILITY:
 - Identifica el UE objetivo utilizando "anyUe".

Cuando el NWDAF recibe la petición, analiza los datos analíticos solicitados conforme a los parámetros establecidos en la petición. Posteriormente, responde al NF con el código de estado "200 OK" y el cuerpo de la respuesta contiene los análisis solicitados.

2.2.4 Nnwdaf_DataManagement

El servicio de Data Management facilita a los consumidores el crear suscripciones para recibir

notificaciones de datos.

2.2.4.1 Creación de suscripción de datos

La siguiente figura muestra el escenario donde un NF consumidor envía una petición al NWDAF para realizar una suscripción con el objetivo de obtener datos.

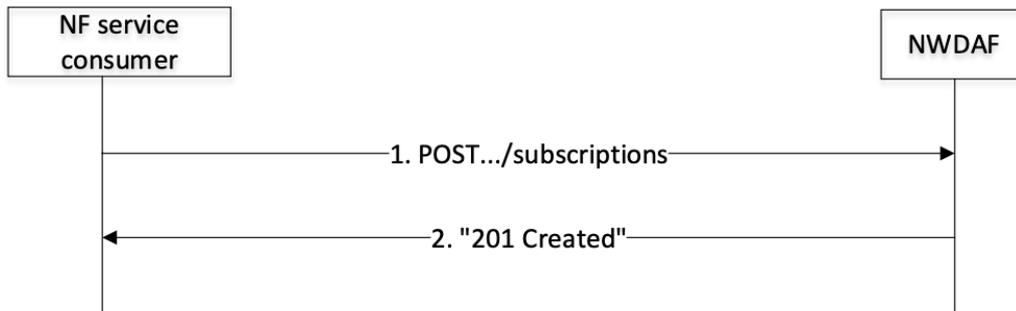


Figura 13. Petición de suscripción de datos al NWDAF

El NF realiza una petición detallada, incluyendo en el cuerpo de la solicitud los siguientes datos esenciales:

- notifURI: Dirección donde se enviarán las notificaciones con los datos al consumidor.
- notifCorrId: Identificador de correlación de las notificaciones.
- Y unos de los siguientes atributos:
 - o anaSub: información sobre una suscripción de datos de análisis que se usara para determinar los datos necesarios para ser recogido.
 - o dataSub: suscripción con la información de los datos pedidos.

Además de esos atributos obligatorios, se pueden incluir:

- formatInstruct: instrucciones para el formateo de los datos.
- procInstruc: instrucciones para el procesamiento de los datos.
- timePeriod: Ventana de tiempo en la cual se piden los datos.
- dataCollectPurposes: El propósito de los datos que se están pidiendo.

Cuando el NWDAF recibe esta petición, procede a crear la suscripción asignándole un ID único y almacenándola para su gestión.

Si la petición se procesa correctamente, el NWDAF responde con el código de estado "201 Created". El cuerpo de la respuesta contiene la representación detallada de la suscripción creada, mientras que en la cabecera de la respuesta se incluye el campo "Location". Este campo "Location" proporciona la URL donde se puede acceder a la suscripción recién creada, siguiendo el formato:

`{apiRoot}/nnwdaf-datamanagement/v1/subscriptions/{subscriptionId}`.

2.2.4.2 Actualización de suscripción a datos

La siguiente figura ilustra el escenario en el cual un NF consumidor envía una petición al NWDAF para actualizar su suscripción de datos:

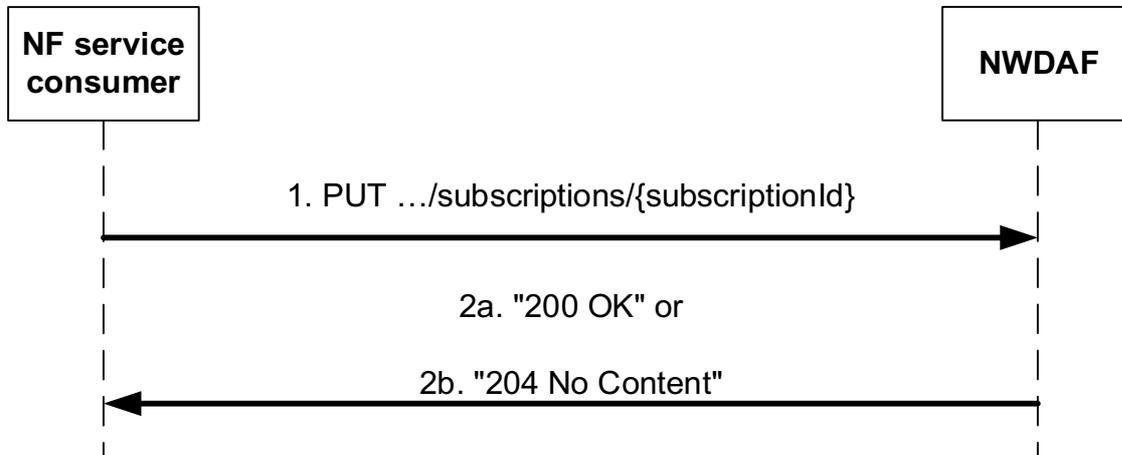


Figura 14. Actualización de suscripción a notificaciones de datos

El NF consumidor realizará la petición a la URI proporcionada en la cabecera de la respuesta de su suscripción (campo "Location"), la cual contiene el subscriptionId necesario para identificar la suscripción que se desea modificar. En esta petición, el NF consumidor incluirá el cuerpo con los nuevos detalles de la suscripción.

Una vez que el NWDAF reciba esta petición, procederá a actualizar la suscripción correspondiente al subscriptionId especificado. Si la operación se realiza con éxito, el NWDAF responderá con el código de estado "200 OK". En caso de que no se requiera devolver ningún contenido adicional, el NWDAF puede responder con el código "204 No Content".

2.2.4.3 Eliminación de suscripción a datos

La siguiente figura ilustra el escenario en el cual un NF consumidor envía una petición al NWDAF para eliminar una suscripción a datos:

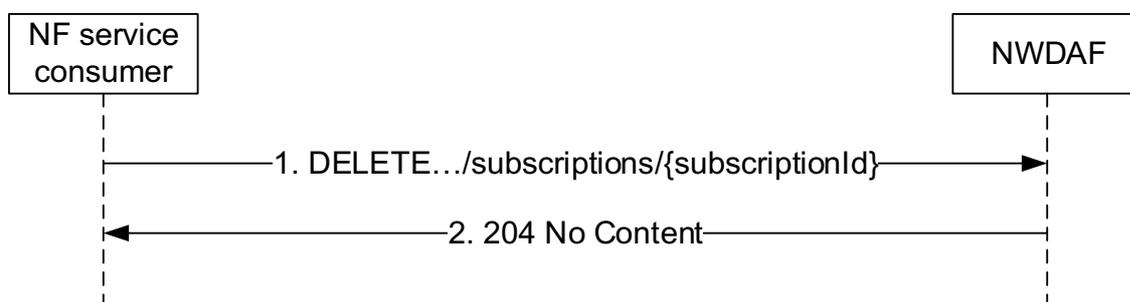


Figura 15. Eliminación de suscripción a notificaciones de datos

El NF consumidor realizará la petición a la URI proporcionada en la cabecera de la respuesta de su suscripción (campo "Location"), la cual contiene el subscriptionId necesario para identificar la suscripción que se desea eliminar.

Una vez que el NWDAF reciba esta petición, procederá a eliminar la suscripción correspondiente al subscriptionId especificado. Si la operación se realiza con éxito y no se requiere devolver ningún contenido

adicional, el NWDAF responderá con el código de estado "204 No Content".

2.2.4.4 Notificación de una suscripción de datos

La siguiente figura ilustra el escenario en el cual el NWDAF envía una notificación al NF consumidor sobre sus suscripciones de datos:

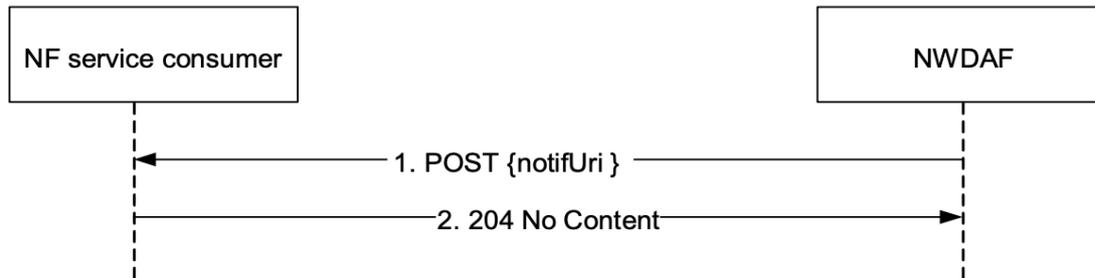


Figura 16. Envío de notificaciones del NWDAF

El NWDAF enviará la notificación a la dirección indicada en el atributo "notifURI" de la suscripción del NF consumidor. La notificación deberá incluir los siguientes atributos:

- notifCorrId: el identificador de correlación de la suscripción.
- notifTimestamp: la fecha de la creación de la notificación
- uno de los siguientes atributos:
 - o dataNotification: datos recogidos de fuentes de datos como funciones de red.
 - o dataReports: Resumen de datos derivados de las instrucciones de procesamiento y formateo de la suscripción.

Además de estos atributos, el NWDAF puede indicar en "terminationReq" la petición de una terminación de la suscripción.

Cuando el consumidor recibe esta petición del NWDAF, deberá de devolver el código "204 No Content" y almacenar la notificación.

2.2.5 Nnwdaf_MLModelProvision

El servicio de ML Model provision facilita a los consumidores el crear suscripciones para recibir notificaciones sobre modelos de Machine Learning. Este servicio es expuesto por el NWDAF que contenga la función lógica MTLF. Los consumidores de este servicios serán otros NWDAFs que contengan la función lógica de AnLF.

2.2.5.1 Creación de suscripción a modelos ML

La siguiente figura muestra el escenario donde un NF consumidor envía una petición al NWDAF para realizar una suscripción con el objetivo de obtener la dirección de un modelo de Machine Learning.

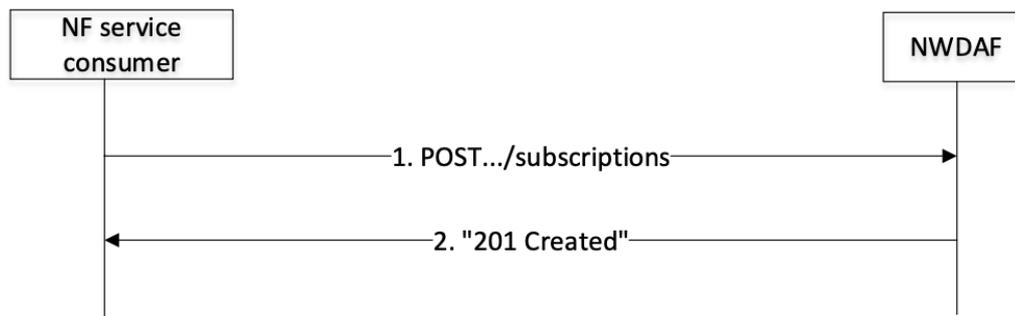


Figura 17. Petición de suscripción de modelos ML al NWDAF

El NF realiza una petición detallada, incluyendo en el cuerpo de la solicitud los siguientes datos esenciales:

- notifURI: Dirección donde el NWDAF enviará la notificación de la suscripción.
- mLEventSubscs: Descripción sobre los datos del evento del que se quiere el modelo de ML. Este atributo debe contener los siguientes atributos:
 - o mLEvent: Identificador del evento al que nos queremos suscribir para obtener el modelo de ML.
 - o mLEventFilter: Información de filtrado que se utiliza para seleccionar el modelo específico que se pide para la suscripción.

Además, podrá incluir los siguientes atributos:

- notifCorrelId: Identificador de correlación asignado por el consumidor para las notificaciones pedidas.
- eventReq: Los requerimientos del envío de las notificaciones.

Para los diferentes tipos de eventos que se pueden pedir en “mLEvent”, se pueden requerir distintos atributos obligatorios u opcionales que se especificaran en “mLEventFilter”.

Cuando el NWDAF recibe esta petición, procede a crear la suscripción asignándole un ID único y almacenándola para su gestión.

Si la petición se procesa correctamente, el NWDAF responde con el código de estado "201 Created". El cuerpo de la respuesta contiene la representación detallada de la suscripción creada, mientras que en la cabecera de la respuesta se incluye el campo "Location". Este campo "Location" proporciona la URL donde se puede acceder a la suscripción recién creada, siguiendo el formato:

- {apiRoot}/nnwdaflmlmodelprovision/v1/subscriptions/{subscriptionId}.

2.2.5.2 Actualización de suscripción de modelos ML

La siguiente figura ilustra el escenario en el cual un NF consumidor envía una petición al NWDAF para actualizar su suscripción de modelos de Machine Learning:

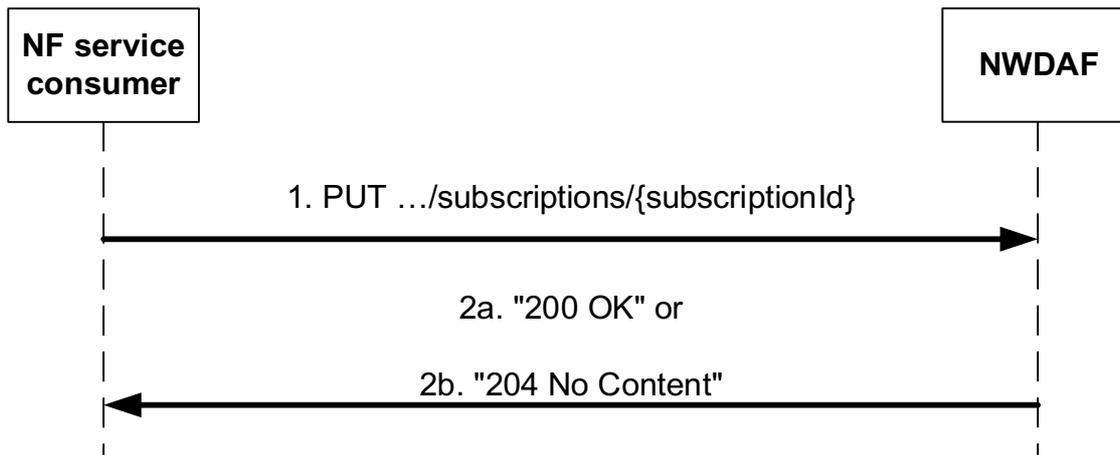


Figura 18. Actualización de suscripciones a notificaciones de modelos ML

El NF consumidor realizará la petición a la URI proporcionada en la cabecera de la respuesta de su suscripción (campo "Location"), la cual contiene el subscriptionId necesario para identificar la suscripción que se desea modificar. En esta petición, el NF consumidor incluirá el cuerpo con los nuevos detalles de la suscripción.

Una vez que el NWDAF reciba esta petición, procederá a actualizar la suscripción correspondiente al subscriptionId especificado. Si la operación se realiza con éxito, el NWDAF responderá con el código de estado "200 OK". En caso de que no se requiera devolver ningún contenido adicional, el NWDAF puede responder con el código "204 No Content".

2.2.5.3 Eliminación de suscripción de modelos de ML

La siguiente figura ilustra el escenario en el cual un NF consumidor envía una petición al NWDAF para eliminar una suscripción de modelos de Machine Learning:

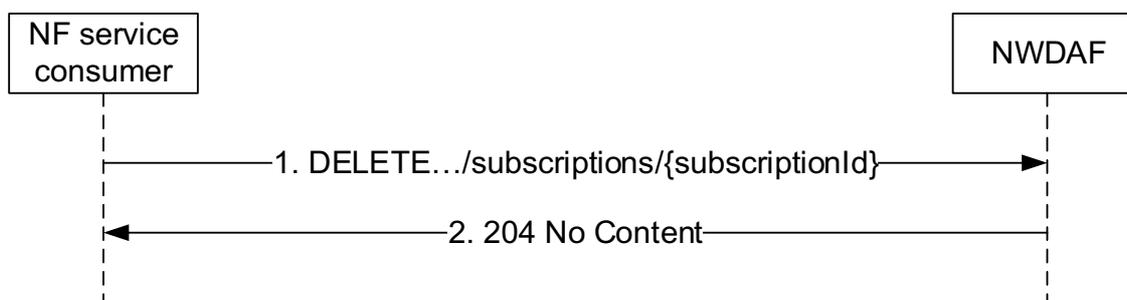


Figura 19. Eliminación de suscripciones a notificaciones de modelos de ML

El NF consumidor realizará la petición a la URI proporcionada en la cabecera de la respuesta de su suscripción (campo "Location"), la cual contiene el "subscriptionId" necesario para identificar la suscripción que se desea eliminar.

Una vez que el NWDAF reciba esta petición, procederá a eliminar la suscripción correspondiente al "subscriptionId" especificado. Si la operación se realiza con éxito y no se requiere devolver ningún contenido adicional, el NWDAF responderá con el código de estado "204 No Content".

2.2.5.4 Notificación de una suscripción de modelos de ML

La siguiente figura ilustra el escenario en el cual el NWDAF envía una notificación al NF consumidor

sobre sus suscripciones de modelos de ML:

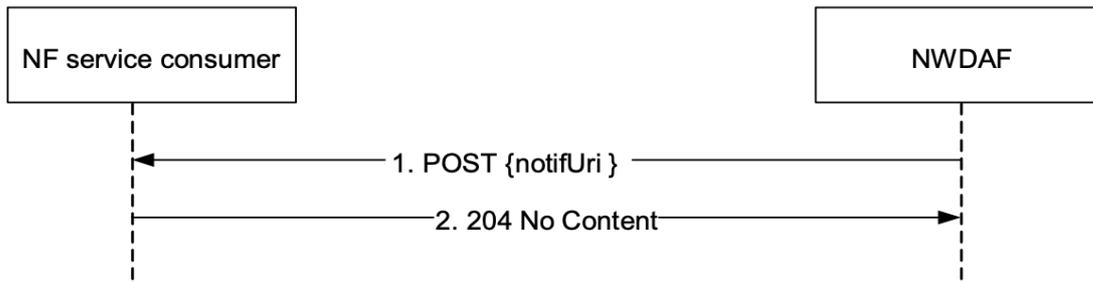


Figura 20. Envío de notificaciones del NWDAF

El NWDAF enviará la notificación a la dirección indicada en el atributo "notifURI" de la suscripción del NF consumidor. La notificación deberá incluir los siguientes atributos:

- subscriptionId: Identificador de la suscripción a la cual pertenece la notificación.
- eventNotifs: una descripción sobre los modelos de los eventos pedidos en la suscripción, que contiene:
 - o event: evento al que hace referencia el modelo.
 - o mLFileAddr: dirección donde se encuentra el modelo.
 - o notifCorrelId: identificador de correlación de la suscripción.
 - o validityPeriod: periodo de tiempo en el cual aplica el modelo.
 - o spatialValidity: Área en la cual aplica el modelo

Cuando el consumidor recibe esta petición del NWDAF, deberá de devolver el código "204 No Content" y almacenar la notificación para poder hacer uso de los modelos.

2.2.6 Opiniones sobre la especificación 29520

La especificación 29520 complementa a la 23288 al proporcionar una descripción más detallada de los servicios de NWDAF. En esta especificación, se presentan todos los endpoints de cada servicio, así como los datos necesarios que deben ser suministrados y los resultados que se pueden obtener.

En relación con los eventos de interés, como NF_LOAD y QOS_SUSTAINABILITY, la especificación ofrece toda la información pertinente para cada servicio vinculado a estos eventos. Esto simplifica el desarrollo del prototipo, ya que nos permite omitir atributos no relevantes en la implementación.

Adicionalmente, el estándar incluye los archivos YAML de OpenAPI para los servicios de NWDAF. Estos archivos son valiosos porque nos permiten generar el esqueleto de los servicios y visualizar las peticiones con todos sus atributos mediante el uso de Swagger. Esta capacidad de visualizar y organizar las peticiones facilita enormemente la tarea de desarrollo y asegura que todas las interacciones con los servicios sean claras y bien documentadas.

3 Planificación

3.1 Tareas del proyecto

En el desarrollo del proyecto, se han identificado una serie de tareas fundamentales que deben llevarse a cabo para completar con éxito el proyecto. Estas tareas son:

- Lectura y análisis de las especificaciones del NWDAF
- Diseño del prototipo
- Desarrollo del prototipo
- Validación del prototipo

Cada una de estas tareas tiene fechas límite claramente definidas para asegurar un progreso ordenado y eficiente:

- 1 de mayo de 2023: Finalización del análisis de las especificaciones
- 2 de junio de 2023: Diseño del prototipo del NWDAF
- 16 de agosto de 2023: Desarrollo del prototipo de NWDAF
- 31 de agosto de 2023: Validación del prototipo de NWDAF

Para llevar a cabo el proyecto, se ha establecido una jornada laboral de media jornada, lo que implica una dedicación de 4 horas diarias de lunes a viernes. Con estas horas de trabajo y las fechas límite en mente, se han subdividido las tareas principales en subtareas más manejables, todas organizadas en un detallado diagrama de Gantt:

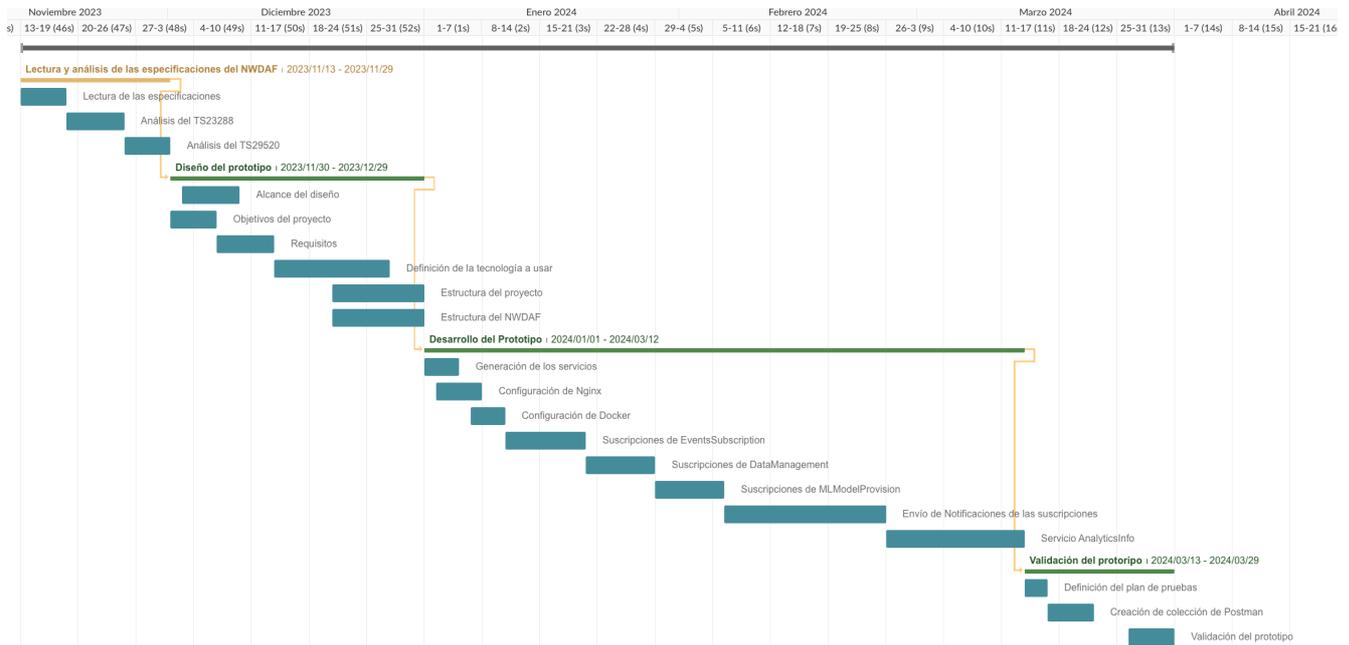


Figura 1. Diagrama de Gantt del proyecto.

Las subtareas, representadas en el diagrama de Gantt, se pueden observar en la siguiente tabla. En esta tabla se muestra a qué tarea principal pertenece cada subtarea, las fechas de inicio y finalización planificadas, así como el número total de horas estimadas para cada una. Al planificar estas subtareas, se han considerado posibles retrasos, como la imposibilidad de trabajar las 4 horas diarias completas en ciertos días o demoras imprevistas:

Número de EDT	Nombre de tarea / Título	Fecha de inicio planificada	Fecha de finalización planificada	Duración (horas)
1	Lectura y análisis de las especificaciones del NWDAF	13/11/23	29/11/23	52
1.1	Lectura de las especificaciones	13/11/23	16/11/23	16
1.2	Análisis del TS23288	17/11/23	23/11/23	20
1.3	Análisis del TS29520	24/11/23	29/11/23	16
2	Diseño del prototipo	30/11/23	29/12/23	88
2.1	Alcance del diseño	1/12/23	7/12/23	20
2.2	Objetivos del proyecto	30/11/23	5/12/23	16
2.3	Requisitos	6/12/23	12/12/23	20
2.4	Definición de la tecnología a usar	13/12/23	26/12/23	40
2.5	Estructura del proyecto	20/12/23	29/12/23	32
2.6	Estructura del NWDAF	20/12/23	29/12/23	32
3	Desarrollo del Prototipo	1/1/24	12/3/24	208
3.1	Generación de los servicios	1/1/24	3/1/24	12
3.2	Configuración de Nginx	2/1/24	5/1/24	16
3.3	Configuración de Docker	5/1/24	9/1/24	12
3.4	Suscripciones de EventsSubscription	10/1/24	18/1/24	28
3.5	Suscripciones de DataManagement	19/1/24	26/1/24	24
3.6	Suscripciones de MLModelProvision	29/1/24	5/2/24	24
3.7	Envío de Notificaciones de las suscripciones	6/2/24	23/2/24	56
3.8	Servicio AnalyticsInfo	26/2/24	12/3/24	48
4	Validación del prototipo	13/3/24	29/3/24	52
4.1	Definición del plan de pruebas	13/3/24	14/3/24	8
4.2	Creación de colección de Postman	15/3/24	20/3/24	16
4.3	Validación del prototipo	26/3/24	29/3/24	16

Tabla 1. Tareas del proyecto

Tras realizar una planificación exhaustiva, se ha calculado que el proyecto requerirá aproximadamente 400 horas de trabajo, incluyendo un margen para posibles interrupciones o imprevistos que puedan surgir a lo largo del desarrollo. Esta planificación detallada nos permite anticipar y gestionar de manera eficiente cualquier contratiempo, asegurando así el éxito del proyecto.

3.2 Presupuesto

Para la ejecución del proyecto, se requiere la contratación de un programador que se encargue de llevar a cabo todas las tareas definidas en la planificación. Este profesional deberá dedicar aproximadamente 580 horas al proyecto, que es la duración total estimada teniendo en cuenta los posibles retrasos.

El sueldo promedio de un programador en España es de 14,62€ por hora, lo que nos lleva a calcular el costo total de su contratación de la siguiente manera:

$$400 \text{ horas} \times 14,62\text{€/hora} = 5848\text{€}$$

Además, es necesario proporcionar un portátil al programador para el desarrollo de sus tareas. El coste estimado del portátil es de aproximadamente 1000€. Pensando que un portátil tiene un periodo de

amortización de 3 años, obtenemos que por año el portátil cuesta 333€, obtenemos que el portátil cuesta al mes:

$$333 \text{ €} / 12 \text{ meses} = 27,75 \text{ €/mes}$$

Todo esto nos lleva a la siguiente distribución presupuestaria con un periodo de vida del proyecto de 5 meses:

Concepto	Cantidad	Precio Unitario	Coste Total
Programador	400h	14,62 €/h	5848 €
Portátil	5 meses	27,75 €/mes	138,75 €
Total			5986,75€

Tabla 2. Coste total del proyecto

El coste final estimado del proyecto es de 5986,75€. Este presupuesto incluye una provisión para posibles pérdidas de horas de trabajo, asegurando que cualquier imprevisto pueda ser cubierto sin comprometer la finalización del proyecto.

3.3 Riesgos

Para asegurar el éxito del proyecto, es crucial identificar y gestionar los posibles riesgos que podrían afectar su desarrollo, incluyendo aquellos que podrían extender las fechas límite. A continuación, se presenta un análisis de los riesgos potenciales, evaluando su probabilidad e impacto:

Código	Nombre	Descripción	Probabilidad	Impacto
01	Enfermedad	El programador contratado no puede trabajar en el proyecto debido a que está enfermo.	Baja	Alto
02	Problemas con las especificaciones	Al analizar las especificaciones surgen problemas que ralentizan el proyecto.	Media	Alto
03	Perdida del entorno de trabajo	Problemas en el hardware o software que se utiliza en el proyecto.	Baja	Alto
04	Retrasos en la entrega del hardware	El portátil o componentes necesarios no llegan a tiempo, retrasando el inicio del proyecto.	Baja	Medio
05	Cambios en los requisitos	Solicitudes de cambios o ajustes en los requisitos, afectando la planificación original.	Media	Alto
06	Problemas de conectividad	Fallos en la conexión a internet que impiden el acceso a recursos en la nube.	Baja	Medio
07	Falta de conocimientos técnicos	El programador enfrenta dificultades técnicas que no puede resolver de manera eficiente.	Media	Alto
08	Problemas de integración	Dificultades al integrar diferentes componentes del sistema, lo que puede causar retrasos significativos.	Media	Alto
09	Problemas con la obtención de datos	Dificultades para obtener datos de otras funciones de red para las notificaciones del NWDAF	Media	Alto

Tabla 3. Posibles riesgos del proyecto

Al identificar y evaluar estos riesgos, podemos implementar estrategias de mitigación adecuadas, asegurando así que el proyecto avance de manera eficiente y se completen dentro de los plazos establecidos.

Código	Solución	Tipo	Riesgo Residual
01R1	Corrección de horas en la planificación y aumentar hora de dedicación a las tareas.	Correctora	Medio
01R2	Posibilidad de buscar otro programador.	Correctora	Alto
02R1	Analizar el problema y corregir las horas de la planificación si es necesario	Correctora	Medio
02R2	Dedicar más tiempo al análisis de las especificaciones.	Mitigadora	Medio
03R1	Realizar copias de seguridad regulares.	Mitigadora	Bajo
03R2	Tener un portátil de reemplazo disponible.	Mitigadora	Bajo
04R1	Ordenar el portátil con anticipación.	Mitigadora	Bajo
04R2	Confirmar fechas de entrega con proveedores.	Mitigadora	Bajo
05R1	Establecer un proceso formal de gestión de cambios.	Mitigadora	Bajo
06R1	Tener una conexión de respaldo.	Correctora	Bajo
07R1	Acceso a recursos técnicos y expertos.	Mitigadora	Bajo
08R1	Realizar pruebas de integración frecuentes.	Mitigadora	Bajo
09R1	Buscar soluciones para la obtención de datos	Correctora	Medio

Tabla 4. Soluciones planteadas para mitigar o corregir los riesgos

Con estas soluciones en marcha, hemos reducido significativamente la probabilidad e impacto de los riesgos identificados. Por lo tanto, podemos considerar que los riesgos están controlados y podemos proceder con las tareas planificadas según el cronograma establecido.

Este enfoque proactivo en la gestión de riesgos nos permitirá avanzar con mayor seguridad y confianza, asegurando que el proyecto se complete dentro de los plazos y presupuestos acordados.

4 Diseño y desarrollo del prototipo

Una vez entendido las especificaciones y realizada la planificación y el análisis de riesgos, pasamos a planear el desarrollo del prototipo.

4.1 Alcance del diseño

El NWDAF se compone de actividades o sucesos denominados eventos, que implican la recopilación, análisis y generación de datos.

Debido a la gran cantidad de eventos y la complejidad de algunos de ellos, en este proyecto nos centraremos en el desarrollo de dos eventos específicos:

- **NF_LOAD**: Este evento utiliza los datos correspondientes a los niveles de carga de las funciones de la red.
- **QOS_SUSTAINABILITY**: Este evento se enfoca en utilizar los datos necesarios para mantener y garantizar la calidad de servicio de la red.

Utilizaremos estos dos eventos como base para los distintos servicios del NWDAF, prestando especial atención a los componentes de EventsSubscription y AnalyticsInfo.

Para los servicios de DataManagement y MLModelProvision, es necesaria la comunicación con otras funciones de red en un entorno de red 5G, así como el uso de modelos predictivos de Machine Learning.

Es posible que el NWDAF no pueda comunicarse con otras NFs o que estas NFs no estén preparadas para responder a nuestras solicitudes. En tales casos, se desarrollará la capacidad de suscripción a datos del servicio DataManagement sin incluir la funcionalidad de solicitar datos directamente a otras NFs.

En cuanto al servicio MLModelProvision, el entrenamiento de modelos de Machine Learning está fuera del alcance del estándar. No obstante, implementaremos la capacidad de suscripción a modelos de ML, aunque sin la funcionalidad de recibir la localización de dichos modelos.

Este es el alcance propuesto al inicio del proyecto, el cual puede variar a lo largo de su desarrollo. La flexibilidad para adaptarse a los cambios y nuevas necesidades es fundamental para el éxito del proyecto.

4.2 Objetivos del proyecto

Para este proyecto, me he centrado en conseguir los siguientes objetivos:

Tabla 5. Objetivos del proyecto

OBJ-1	Suscripción NF_LOAD analítica
Descripción	El servicio deberá permitir suscribirse para obtener la información correspondiente al evento NF_LOAD utilizando el servicio de EventsSubscription y siguiendo el estándar.
Prioridad	Alta
Asequibilidad	Alta

OBJ-2	Suscripción QOS_SUSTAINABILITY analítica
Descripción	El servicio deberá permitir suscribirse para obtener la información correspondiente al evento QOS_SUSTAINABILITY utilizando el servicio de

	EventsSubscription y siguiendo el estándar.
Prioridad	Alta
Asequibilidad	Alta

OBJ-3	Obtener análisis de NF_LOAD
Descripción	El servicio deberá permitir suscribirse para obtener la información correspondiente al evento NF_LOAD utilizando el servicio de AnalyticsInfo y siguiendo el estándar.
Prioridad	Alta
Asequibilidad	Alta

OBJ-4	Obtener análisis de QOS_SUSTAINABILITY
Descripción	El servicio deberá permitir suscribirse para obtener la información correspondiente al evento QOS_SUSTAINABILITY utilizando el servicio de AnalyticsInfo y siguiendo el estándar.
Prioridad	Alta
Asequibilidad	Alta

OBJ-5	Obtener predicciones de NF_LOAD
Descripción	El servicio deberá permitir suscribirse para obtener la predicción correspondiente al evento NF_LOAD utilizando el servicio de AnalyticsInfo y siguiendo el estándar.
Prioridad	Baja
Asequibilidad	Baja
Comentario	Debido a que el entrenamiento de modelos está fuera del estándar no se contempla que se realice en el proyecto, sin embargo, se intentará realizar el objetivo con modelos pre-entrenados.

OBJ-6	Obtener predicciones de QOS_SUSTAINABILITY
Descripción	El servicio deberá permitir suscribirse para obtener la predicción correspondiente al evento QOS_SUSTAINABILITY utilizando el servicio de AnalyticsInfo y siguiendo el estándar.
Prioridad	Baja
Asequibilidad	Baja
Comentario	Debido a que el entrenamiento de modelos está fuera del estándar no se contempla que se realice en el proyecto, sin embargo, se intentará realizar el objetivo con modelos pre-entrenados.

OBJ-7	Suscribirse al servicio DataManagement
Descripción	El servicio deberá permitir suscribirse a datos utilizando el servicio de DataManagement y siguiendo el estándar.
Prioridad	Alta
Asequibilidad	Alta

OBJ-8	Suscribirse al servicio DataManagement y obtener datos de un NF
Descripción	El servicio deberá permitir suscribirse a datos de un NF en una red 5G utilizando el servicio de DataManagement y siguiendo el estándar.
Prioridad	Baja
Asequibilidad	Baja
Comentario	Debido a que no sé si dispondré de una red 5G con alguna función de red

	que permita la petición de datos.
--	-----------------------------------

OBJ-9	Suscribirse al servicio MLModelProvision
Descripción	El servicio deberá permitir suscribirse a un modelo de ML utilizando el servicio de MLModelProvision y siguiendo el estándar.
Prioridad	Alta
Asequibilidad	Alta

OBJ-10	Suscribirse al servicio MLModelProvision y devolver modelo ML
Descripción	El servicio deberá permitir suscribirse a un modelo de ML y responder con la localización de dicho modelo utilizando el servicio de MLModelProvision y siguiendo el estándar.
Prioridad	Baja
Asequibilidad	Baja
Comentario	Debido a que no se si dispondré de algún modelo ML que devolver.

4.3 Tecnología usada

Para el desarrollo del prototipo del NWDAF, se han utilizado diversas tecnologías y herramientas que han facilitado tanto la implementación como la gestión del proyecto. A continuación, se detallan las tecnologías empleadas y las razones por las que se han seleccionado:

4.3.1 Visual Studio Code

- Descripción: Visual Studio Code es un editor de código fuente altamente personalizable y extensible.
- Razones de Uso: Se ha seleccionado debido a su interfaz de usuario personalizable y a su amplia gama de extensiones, que permiten utilizar herramientas adicionales como Docker, mejorando así la productividad y la eficiencia en el desarrollo del prototipo.

4.3.2 GitHub

- Descripción: GitHub es una plataforma de desarrollo colaborativo basada en la web que utiliza el sistema de control de versiones Git.
- Razones de Uso: Se ha utilizado para alojar el código del proyecto, facilitando la colaboración entre desarrolladores y el control de versiones, permitiendo un seguimiento detallado de los cambios y una fácil integración de nuevas funcionalidades.

4.3.3 Python

- Descripción: Python es un lenguaje de programación versátil y ampliamente utilizado en el desarrollo de software, especialmente en áreas como la inteligencia artificial y el análisis de datos ^[11].
- Razones de Uso: Se ha seleccionado debido a su vasta biblioteca de frameworks y herramientas que amplían las capacidades del lenguaje, facilitando el desarrollo de componentes del NWDAF especializados en inteligencia artificial y análisis de datos.

4.3.4 Flask

- Descripción: Flask es un microframework web ligero para Python que facilita la creación de aplicaciones web ^[12].

- Razones de Uso: Se ha utilizado para construir aplicaciones web de manera rápida y sencilla, ofreciendo flexibilidad para combinar distintas bibliotecas y añadir funcionalidades como la autenticación de usuarios y el acceso a bases de datos.

4.3.5 OpenAPI Generator

- Descripción: OpenAPI Generator es una herramienta de código abierto que permite generar código cliente y servidor a partir de especificaciones OpenAPI [7].
- Razones de Uso: Se ha utilizado para facilitar la creación de los servicios del NWDAF, permitiendo generar automáticamente el cuerpo del estándar con todos los modelos de datos necesarios, lo que acelera el proceso de desarrollo y asegura la conformidad con las especificaciones.

4.3.6 MongoDB

- Descripción: MongoDB es un sistema de gestión de bases de datos NoSQL orientado a documentos, que permite almacenar documentos JSON de manera flexible [5].
- Razones de Uso: Se ha elegido debido a su flexibilidad y capacidad para manejar datos dinámicos, lo que es ideal para las necesidades de almacenamiento del NWDAF, además de ser una solución común y bien soportada en proyectos modernos.

4.3.7 Docker

- Descripción: Docker es una plataforma para automatizar el despliegue de aplicaciones en contenedores, garantizando consistencia en cualquier entorno [9].
- Razones de Uso: Se ha utilizado para simplificar y estandarizar el despliegue de los servicios del NWDAF, asegurando que el prototipo se ejecute de manera consistente y fiable en diferentes entornos de desarrollo y producción.

4.3.8 Swagger

- Descripción: Swagger es una suite de herramientas para diseñar, construir, documentar y consumir APIs. Facilita la creación de documentación interactiva y probada de las APIs [2].
- Razones de Uso: Swagger se ha utilizado para diseñar y documentar las APIs del NWDAF, permitiendo a los desarrolladores y stakeholders comprender y probar las funcionalidades de las APIs de manera eficiente y precisa.

4.3.9 Postman

- Descripción: Postman es una plataforma de colaboración para el desarrollo de APIs que permite realizar pruebas, monitorear, documentar y diseñar APIs de manera colaborativa [8].
- Razones de Uso: Postman se ha utilizado para probar y verificar las APIs del NWDAF, asegurando que las peticiones y respuestas funcionen correctamente, y proporcionando un entorno para pruebas automatizadas y colaborativas.

4.4 Requisitos del proyecto

En esta sección, se detallan los requisitos específicos que hemos recibido para llevar a cabo el proyecto. Entre ellos, se destaca la necesidad de emplear diversas herramientas durante el desarrollo del Network Data Analytics Function (NWDAF), las cuales son fundamentales para facilitar su implementación y

despliegue efectivos. Estas herramientas no solo optimizan el proceso de desarrollo, sino que también aseguran que el NWDAF cumpla con los estándares requeridos y pueda integrarse sin problemas en el entorno de red existente.

4.4.1 OpenAPI Generator

El estándar 29520 proporciona archivos YAML que contienen todas las definiciones de modelos de datos necesarios para generar el esqueleto de servicios.

Para llevar a cabo esta generación, emplearemos OpenAPI Generator, una herramienta que nos permite seleccionar el lenguaje de programación y el framework que deseamos utilizar para exponer los servicios.

En nuestro caso, hemos optado por Python debido a su versatilidad y legibilidad, así como por su extensa colección de librerías y frameworks disponibles. Entre los frameworks disponibles, hemos elegido Flask para implementar los servicios, gracias a su diseño minimalista, flexibilidad y su capacidad para integrarse fácilmente con otros componentes del ecosistema Python.

Los servicios se generarán con la siguiente estructura:

```
├── Dockerfile
├── README.md
├── git_push.sh
├── openapi_server
│   ├── _init_.py
│   ├── _main_.py
│   ├── controllers
│   ├── encoder.py
│   ├── models
│   ├── openapi
│   ├── test
│   ├── typing_utils.py
│   └── util.py
├── requirements.txt
├── setup.py
├── test-requirements.txt
└── tox.ini
```

Dentro del directorio `openapi_server` encontramos todos los elementos necesarios para el funcionamiento del servicio:

- El archivo `main.py` es el punto de entrada principal encargado de realizar todas las operaciones iniciales necesarias antes de levantar el servicio.
- El directorio `controllers` contiene archivos Python que definen las funciones a ser llamadas cuando se realiza una petición al servicio.
- El directorio `models` alberga todos los modelos de datos necesarios para el servicio.

- El archivo `encoder.py` permite transformar los modelos de datos en formato JSON para facilitar su envío.
- El directorio `openapi` contiene el archivo YAML del servicio, crucial para gestionar las peticiones y respuestas según la especificación OpenAPI.
- Los archivos `typing_utils` y `utils` son indispensables para manejar los modelos de datos de manera efectiva.

Además de generar el código base del servicio, OpenAPI Generator facilita el despliegue de este en Docker. Esto se logra mediante la generación automática de un Dockerfile y archivos con todos los requerimientos básicos para el despliegue, simplificando así el proceso de implementación y asegurando la portabilidad del servicio entre diferentes entornos.

4.4.2 Nginx

Para exponer todos los servicios, implementaremos NGINX como un proxy inverso. NGINX se elige por su alto rendimiento y escalabilidad, capaz de manejar un gran número de conexiones concurrentes de manera eficiente. Además, proporciona equilibrio de carga, es flexible, sencillo de configurar y cuenta con soporte integrado para SSL/TLS, lo cual es crucial para futuras necesidades de seguridad ^[6].

En nuestro proyecto, configuraremos NGINX con 4 endpoints específicos que redirigirán el tráfico hacia cada uno de los servicios, utilizando los nombres propuestos por los estándares:

- `/nnwdaf-eventssubscription`
- `/nnwdaf-datamanagement`
- `/nnwdaf-mlmodelprovision`
- `/nnwdaf-analyticsinfo`

Estos endpoints están alineados con los estándares establecidos, facilitando la integración y el entendimiento dentro del contexto del proyecto. NGINX actuará como un punto centralizado de acceso, gestionando de manera eficiente las solicitudes hacia cada servicio correspondiente, lo cual optimiza tanto el rendimiento como la administración de la infraestructura de servicios.

4.4.3 Docker

OpenAPI Generator automatiza la generación de Dockerfiles para cada servicio, lo que simplifica el despliegue individual de cada uno. Sin embargo, componentes adicionales como MongoDB y NGINX no son gestionados automáticamente de la misma manera.

Para facilitar el despliegue conjunto de todos los servicios junto con MongoDB y NGINX, crearemos un archivo Docker-compose. Este archivo será responsable de instanciar y coordinar todos los contenedores necesarios para el NWDAF.

El Docker-compose incluirá:

- Configuración de contenedores para cada servicio generado por OpenAPI, así como para MongoDB y NGINX.
- Un archivo de variables de entorno donde se almacenarán configuraciones específicas como usuarios, contraseñas de MongoDB, nombres de colecciones, etc.

- Creación de una red Docker para permitir la comunicación entre todos los servicios desplegados, facilitando así la interacción entre NGINX y MongoDB, y garantizando la operatividad del NWDAF en su conjunto.

Este enfoque no solo simplificará el proceso de despliegue, sino que también mejorará la gestión y la escalabilidad del entorno, asegurando una integración fluida y eficiente de todos los componentes del sistema NWDAF.

4.4.4 Estructura de los servicios y del proyecto

Cada servicio generado por OpenAPI Generator se organizará en carpetas separadas, resultando en la siguiente estructura de directorios:

```

├── Analytics_Info
├── Data_Management
├── Events_Subscription
├── ML_Model_Provision
├── _env
├── docker-compose.yml
├── nginx
└── run.sh

```

En detalle:

- Analytics_Info, Data_Management, Events_Subscription, ML_Model_Provision: Cada una de estas carpetas contendrá el código generado por OpenAPI Generator para cada servicio respectivo.
- _env: Esta carpeta almacenará archivos de variables de entorno que serán utilizados por los servicios y configuraciones relacionadas.
- docker-compose.yml: Archivo principal de Docker-compose que orquestrará todos los servicios junto con NGINX y configurará la red para la comunicación entre ellos.
- nginx: Aquí se encontrarán todos los archivos y configuraciones relacionadas con NGINX, que actuará como proxy inverso para dirigir el tráfico a los servicios adecuados.
- run.sh: Este script facilitará las operaciones necesarias durante el despliegue. Incluirá comandos como la ejecución del Docker-compose para iniciar todos los servicios y configuraciones asociadas.

En el nivel superior del proyecto, la estructura de directorios se organiza de la siguiente manera:

```

├── services
├── tests
├── utils
└── README.MD

```

Detallando cada carpeta y su propósito:

- services: Aquí se encuentra el núcleo del NWDAF, que incluye todos los servicios generados por OpenAPI Generator y necesarios para el funcionamiento del sistema. Cada servicio tiene su propia carpeta como se describió anteriormente: Analytics_Info, Data_Management, Events_Subscription, ML_Model_Provision, etc.
- tests: Esta carpeta almacena todo lo necesario para la validación y pruebas de los servicios. Incluirá colecciones de Postman u otros archivos necesarios para realizar pruebas automatizadas o manuales de los endpoints y funcionalidades de los servicios.
- utils: Contiene utilidades adicionales que facilitan el uso del NWDAF. En este caso, podría incluir scripts o configuraciones para levantar servicios básicos necesarios para recibir notificaciones de los servicios del NWDAF. Esto es crucial para validar la información que se envía y asegurar el funcionamiento correcto del sistema en entornos de prueba o desarrollo.
- README.md: Este archivo proporciona documentación básica sobre el proyecto, incluyendo una descripción general, instrucciones para configurar y desplegar el sistema, así como cualquier otra información relevante para los desarrolladores y usuarios del NWDAF.

Este diseño estructurado no solo facilita la gestión y mantenimiento del proyecto, sino que también mejora el modularidad y la escalabilidad al separar claramente cada componente y sus configuraciones correspondientes.

4.5 Arquitectura del NWDAF

La arquitectura del Network Data Analytics Function (NWDAF) se integra de manera eficiente con varios componentes tecnológicos, incluyendo NGINX, Docker y MongoDB, para proporcionar análisis de datos y notificaciones de eventos en redes de telecomunicaciones.

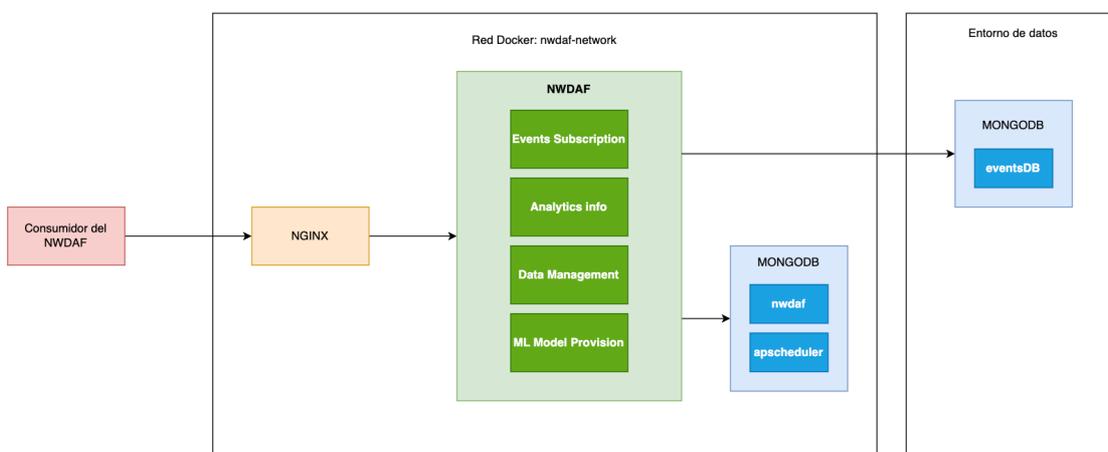


Figura 21. Arquitectura del NWDAF

A continuación, se detallan las conexiones y relaciones que tiene el NWDAF con otros elementos:

El NWDAF se compone de varios servicios clave que facilitan la suscripción a eventos, el acceso a información analítica, la gestión de datos y la provisión de modelos de aprendizaje automático (ML). Estos servicios incluyen el manejo de suscripciones a eventos, la provisión de información analítica basada en los datos recopilados y procesados, la gestión y almacenamiento de datos, y la implementación y acceso a modelos de aprendizaje automático.

Dentro de la red de Docker denominada “nwda-network”, el NWDAF y sus servicios están contenidos y administrados, garantizando un entorno aislado y controlado. Además del NWDAF, esta red incluye un servidor NGINX que actúa como un proxy inverso, exponiendo los servicios del NWDAF fuera de la red interna y enrutando las peticiones a los servicios correspondientes dentro del NWDAF. También se incluye una instancia de MongoDB interna que contiene dos bases de datos críticas: una base de datos llamada nwda que almacena información relacionada con las suscripciones a servicios, y otra base de datos llamada “apscheduler” que contiene la programación de trabajos periódicos necesarios para el envío de notificaciones.

Los consumidores del NWDAF, que pueden ser otras aplicaciones o servicios, realizan peticiones al NWDAF a través del servidor NGINX. Este se encarga de enrutar las solicitudes adecuadamente hacia los servicios internos del NWDAF, garantizando una comunicación fluida y eficiente.

Además, el NWDAF está conectado a una instancia externa de MongoDB que contiene una base de datos denominada “eventDB”. Esta base de datos proporciona los datos necesarios que el NWDAF utiliza para enviar notificaciones basadas en las suscripciones a eventos. Esta conexión asegura que el NWDAF pueda acceder a datos externos relevantes y oportunos para sus análisis y notificaciones.

Esta arquitectura modular y robusta del NWDAF permite gestionar grandes volúmenes de datos y proporciona capacidades avanzadas de análisis y notificación, esenciales para las redes de telecomunicaciones modernas. La utilización de contenedores Docker y bases de datos distribuidas asegura una alta disponibilidad, escalabilidad y eficiencia en la gestión de datos y la entrega de servicios analíticos.

4.6 Metodología

En el desarrollo de los servicios, se ha adoptado una metodología iterativa que incluye pruebas continuas de los servicios a medida que se desarrollaban. Este enfoque metodológico asegura la estabilidad y confiabilidad del sistema. La documentación detallada de los modelos de las APIs desarrolladas se presenta en el anexo de esta documentación.

Se han considerado algunos aspectos claves en el desarrollo del NWDAF que se detallan a continuación:

- Gestión de errores: La gestión de errores es fundamental para garantizar que una API sea robusta y fácil de usar. Esto implica diseñar respuestas coherentes y claras para los distintos tipos de errores que puedan ocurrir, como errores de validación, autenticación, autorización y errores del servidor siguiendo el estándar de códigos.
- Elaboración de documentación utilizando Swagger: Swagger es una herramienta ampliamente utilizada para la creación de documentación interactiva de APIs. Utilizar Swagger permite generar documentación comprensible y accesible. Con Swagger, se pueden describir todos los endpoints de la

API, incluyendo sus métodos HTTP, parámetros de entrada, respuestas y posibles errores. La documentación generada facilita la comprensión y uso de la API por parte de otros desarrolladores y puede servir como guía para el desarrollo y mantenimiento continuo de la API. Además, Swagger permite probar los endpoints directamente desde la documentación, lo que mejora la eficiencia en el desarrollo y la integración. Se ha utilizado SwaggerHub (SwaggerHub) como herramienta para la generación de la documentación de las APIs.

- Realización de pruebas: Las pruebas son esenciales para asegurar que una API funcione correctamente y cumpla con los requisitos especificados.
- Versionado de la API: El versionado de la API es fundamental para mantener la compatibilidad con aplicaciones cliente a lo largo del tiempo. Al introducir cambios significativos en la API, como la adición de nuevas funcionalidades o la modificación de existentes, es importante hacerlo de manera que no se rompa la compatibilidad con versiones anteriores. Esto se puede lograr mediante la creación de diferentes versiones de la API, indicando claramente en la URL o en los encabezados de las solicitudes cuál versión se está utilizando. El versionado permite a los desarrolladores de clientes actualizar sus aplicaciones de manera controlada y gradual.

4.7 Generación de los servicios

Para generar el esqueleto de los servicios, hemos optado por utilizar OpenAPI Generator CLI Tool que facilita la generación automática de servicios a partir de archivos YAML. Específicamente, hemos configurado OpenAPI Generator para utilizar el lenguaje Python con el framework Flask.

Inicialmente, al intentar generar los servicios a partir de los archivos YAML de especificación, nos encontramos con un problema. Estos archivos hacían referencia a otras especificaciones para obtener sus modelos, lo cual generaba un error indicando la falta de estos archivos adicionales.

Como solución, descubrimos un repositorio en GitHub que contiene todos los archivos YAML de 3GPP [3]. Este repositorio no solo facilita el acceso a las especificaciones completas, sino que también incluye enlaces directos al swagger de los servicios, lo cual fue invaluable para entender mejor las estructuras y relaciones entre los diferentes componentes.

Con la ayuda de este repositorio, logramos generar todos los servicios del NWDAF de manera sencilla utilizando el siguiente comando de OpenAPI Generator:

```
openapi-generator generate -i 5GC_APIs/TS29520_Nnwdaf_EventsSubscription.yaml -g python-flask -o NWDAF/Events_Subscriptions
```

En este comando, especificamos el archivo de entrada con el flag `-i`, indicamos el lenguaje y framework deseado con `-g`, y definimos la carpeta de salida con `-o`. Cada servicio generado se colocó en su directorio correspondiente dentro de la estructura del proyecto, lo cual nos permitió construir de manera progresiva el esqueleto completo del proyecto NWDAF.

Este enfoque nos aseguró una implementación organizada y coherente de los servicios, facilitando la gestión y mantenimiento a medida que avanzamos en el desarrollo del prototipo.

4.8 Configuración del proxy inverso con NGINX

Después de generar todos los servicios, nos encontramos con que cada uno estaba configurado para escuchar en el puerto 8080, lo cual generó conflictos durante el despliegue. Para resolver este problema y garantizar un funcionamiento adecuado, implementamos Nginx como un proxy inverso.

El objetivo principal del proxy inverso Nginx es facilitar el enrutamiento de las peticiones entrantes. Configuramos Nginx para que escuche en el puerto estándar HTTP (generalmente el puerto 80) y luego redirija las solicitudes entrantes a los servicios correspondientes, los cuales estaban configurados para escuchar en diferentes puertos en el mismo servidor.

El proceso implicó configurar Nginx para cada servicio específico, utilizando directivas como `proxy_pass` para redirigir las solicitudes entrantes a la dirección y puerto donde cada servicio estaba escuchando. Por ejemplo, si un servicio estaba escuchando en `localhost:8080`, Nginx lo redirigiría desde `localhost` al puerto 8080 interno.

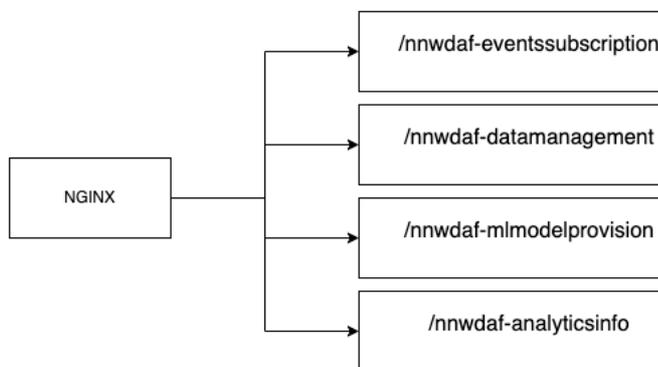


Figura 22. NGINX como proxy inverso

Esta configuración no solo resolvió los problemas de conflictos de puertos, sino que también mejoró la organización y la gestión del tráfico hacia nuestros servicios backend. Además, Nginx proporciona funcionalidades adicionales como balanceo de carga y caching, lo cual puede ser aprovechado para mejorar el rendimiento del sistema en general.

Con estas medidas implementadas, pudimos asegurar un despliegue más eficiente y escalable de los servicios generados, utilizando Nginx como un componente crucial en la arquitectura de nuestra aplicación.

4.9 Configuración de Docker Compose

Para simplificar y optimizar el proceso de despliegue, especialmente con el crecimiento de servicios y la introducción de componentes como Nginx y MongoDB como base de datos, hemos adoptado la utilización de Docker Compose.

Docker Compose nos permite definir y gestionar aplicaciones multi-contenedor de manera declarativa. Hemos creado un archivo `docker-compose.yml` donde especificamos todos los servicios necesarios para el NWDAF, todos ellos conectados a una red Docker llamada "naas-network". Esta configuración asegura que todos los servicios puedan comunicarse internamente y que los puertos estén expuestos según nuestras necesidades:

- Los servicios del NWDAF están configurados para ser expuestos internamente a través del puerto 8080.
- Nginx se expone externamente en el puerto 8070 y internamente en el puerto 80, actuando como proxy inverso para dirigir el tráfico hacia los servicios del NWDAF.
- MongoDB está expuesto internamente en el puerto 27017, proporcionando la capa de persistencia necesaria para nuestra base de datos.

Cada servicio se configura con su propio volumen en Docker. Por ejemplo, Nginx utiliza un volumen para cargar su configuración específica, lo que facilita la gestión y personalización del servidor proxy según nuestras necesidades operativas.

Además, hemos especificado archivos de variables de entorno para cada servicio dentro del archivo docker-compose.yml, asegurando una configuración flexible y modular.

Con el archivo Docker Compose creado, ahora podemos desplegar todos los componentes necesarios para el NWDAF con un simple comando. Para mejorar aún más la automatización y asegurar un despliegue robusto, hemos desarrollado un script llamado run.sh. Este script no solo ejecuta el comando de despliegue a través de Docker Compose, sino que también incluye comprobaciones para garantizar que todos los servicios se hayan iniciado correctamente después del despliegue.

Esta estrategia no solo simplifica el proceso de despliegue y reduce la posibilidad de errores, sino que también mejora la escalabilidad y la administración de nuestra aplicación NWDAF en entornos de desarrollo y producción.

Hasta el momento, hemos organizado nuestro proyecto según el diseño propuesto, asegurando una estructura clara y modular:

- └─ **Analytics_Info**
- └─ **Data_Management**
- └─ **Events_Subscription**
- └─ **ML_Model_Provision**
- └─ **_env**
- └─ **docker-compose.yml**
- └─ **nginx**
- └─ **run.sh**

Cada carpeta corresponde a un servicio del NWDAF y contiene todos los archivos generados por OpenAPI Generator. La carpeta _env almacena los archivos de configuración de variables de entorno necesarios para la correcta configuración de nuestros servicios. La carpeta nginx contiene la configuración específica de Nginx que actúa como proxy inverso para nuestros servicios.

El archivo docker-compose.yml define la configuración de todos los servicios necesarios para nuestro proyecto, organizados dentro de la red Docker "naas-network". Este archivo asegura que todos los contenedores puedan comunicarse entre sí y estén expuestos correctamente a través de los puertos definidos.

Además, hemos creado un script llamado run.sh que automatiza el proceso de despliegue. Este script ejecuta el comando de despliegue usando Docker Compose y realiza comprobaciones para verificar que todos los servicios se hayan iniciado correctamente después del despliegue, garantizando así una puesta en marcha sin problemas.

Con esta estructura y configuración establecidas, estamos preparados para avanzar hacia el desarrollo

de los servicios. Esta modularidad y organización nos permitirá implementar, probar y escalar cada servicio de manera eficiente y controlada.

4.10 Gestión de suscripciones

Los servicios de Events_Subscription, ML_Model_Provision y Data_Management están diseñados para gestionar suscripciones y obtener datos mediante notificaciones. Cada uno de estos servicios expone tres endpoints principales que permiten crear, borrar y actualizar suscripciones de manera uniforme y coherente.

4.10.1 Creación de la suscripción

Para la creación de la suscripción, seguimos estos pasos:

1. Recepción de la petición: Cuando recibimos la solicitud, extraemos los datos de la suscripción del cuerpo del mensaje.
2. Validación de datos: A partir de los datos obtenidos, realizamos una validación exhaustiva para asegurarnos de que todos los campos necesarios estén presentes y tengan el formato correcto. Si encontramos algún error en la validación, respondemos con un código de estado correspondiente (por ejemplo, 400 Bad Request) junto con un mensaje detallado explicando el problema encontrado.
3. Generación del identificador único: Si la validación de datos es exitosa, procedemos a generar un identificador único (ID) para la suscripción. Este ID nos permite identificar de manera única y realizar operaciones de actualización o eliminación sobre la suscripción en el futuro.
4. Almacenamiento en MongoDB: Una vez validados los datos y generado el ID único, almacenamos la información de la suscripción en una colección de MongoDB. Esto asegura que los datos estén persistentemente disponibles para futuras consultas y operaciones.
5. Respuesta: En la respuesta a la solicitud de creación de la suscripción, incluimos en el encabezado (headers) el campo Location, el cual contiene la URL que apunta al recurso recién creado. Esta URL incluye el ID único generado, permitiendo a los usuarios acceder directamente a la suscripción mediante esta dirección.

Este proceso asegura que la creación de suscripciones sea robusta y segura, garantizando la integridad de los datos y proporcionando una buena experiencia al usuario al proporcionarles una referencia clara y accesible al recurso creado.

4.10.2 Eliminación de la suscripción

Para la eliminación de la suscripción seguimos estos pasos:

1. Recepción de la petición: Cuando recibimos la solicitud de eliminación, obtenemos el ID de la suscripción desde los parámetros de la solicitud o desde el cuerpo del mensaje, según el método de la solicitud (GET, POST, DELETE, etc.).
2. Validación del ID: Verificamos si el ID proporcionado en la solicitud pertenece a alguna suscripción almacenada en la base de datos. Para esto, consultamos la colección de MongoDB donde se guardan las suscripciones. Si el ID no corresponde a ninguna suscripción válida, respondemos con un código de estado apropiado (por ejemplo, 404 Not Found) y un mensaje indicando que la suscripción no existe.
3. Eliminación de la suscripción: Si el ID se encuentra en la base de datos y corresponde a una suscripción válida, procedemos a eliminar la entrada correspondiente en la colección de MongoDB. Esto implica ejecutar una operación de borrado que elimine todos los datos asociados con la suscripción identificada por el ID.
4. Respuesta: Después de realizar con éxito la eliminación de la suscripción, respondemos con un código de estado 204 No Content. Este código indica que la solicitud se procesó correctamente y que no hay contenido para devolver en la respuesta. Esto confirma al cliente que la suscripción ha sido eliminada satisfactoriamente del sistema.

Este proceso garantiza que la eliminación de suscripciones sea segura y eficiente, proporcionando retroalimentación clara al usuario sobre el estado de la operación y manteniendo la integridad de los datos en la base de datos MongoDB.

4.10.3 Actualización de la suscripción

Para la actualización de la suscripción seguimos estos pasos:

1. Recepción de la petición: Al recibir la solicitud de actualización, obtenemos del cuerpo del mensaje los nuevos datos de la suscripción que se desea modificar, así como el ID único que identifica la suscripción específica que queremos actualizar.
2. Validación del ID: Verificamos si el ID proporcionado en la solicitud pertenece a alguna suscripción almacenada en la base de datos. Si el ID no corresponde a ninguna suscripción válida, respondemos con un código de estado apropiado (por ejemplo, 404 Not Found) y un mensaje indicando que la suscripción no existe.
3. Validación de datos: Si la validación del ID es exitosa y encontramos la suscripción correspondiente, procedemos a validar los datos proporcionados en el cuerpo de la solicitud. Verificamos que todos los campos necesarios estén presentes y que tengan el formato correcto. Si hay datos faltantes o incorrectos, respondemos con un código de estado (por ejemplo, 400 Bad Request) junto con un mensaje detallado indicando el error encontrado.

4. Comparación y preparación de datos: Si la validación de datos es exitosa, comparamos los datos existentes de la suscripción en la base de datos con los datos proporcionados en la solicitud de actualización. Creamos un diccionario que contiene las diferencias entre los datos actuales y los nuevos datos de la suscripción. Esta comparación nos permite identificar exactamente qué campos han sido modificados y qué valores deben ser actualizados.
5. Almacenamiento de la actualización: Utilizando el ID único de la suscripción, aplicamos las diferencias encontradas para actualizar los datos de la suscripción en la base de datos. Esto implica ejecutar una operación de actualización que garantice que los cambios se reflejen correctamente en el registro de la suscripción.
6. Respuesta: Una vez completada la operación de actualización con éxito, respondemos con un código de estado 200 OK para indicar que la actualización se ha realizado satisfactoriamente. En la respuesta también podríamos incluir detalles adicionales según sea necesario, como, por ejemplo, los datos actualizados de la suscripción.

Este proceso asegura que la actualización de suscripciones sea precisa y segura, manteniendo la consistencia de los datos y proporcionando una retroalimentación clara al usuario sobre el estado de la operación realizada.

4.10.4 Problemas y limitaciones

Después de implementar la administración de suscripciones para los servicios de Data Management y ML Model Provision, nos enfrentamos a un problema, la falta de datos y funcionalidades específicas requeridas por los estándares para completar completamente la funcionalidad de estos servicios.

En el servicio de Data Management, el objetivo era suscribirse para obtener información de diversas funciones de la red. Sin embargo, encontramos que no existe una implementación funcional de estas funciones de red según los estándares, lo que impidió obtener la información necesaria. Como resultado, solo pudimos implementar la administración de suscripciones sin poder cumplir completamente con el objetivo OBJ-8 relacionados con este servicio.

Similarmente, en el servicio de ML Model Provision, se planeaba suscribirse para recibir notificaciones sobre la ubicación de modelos de aprendizaje automático necesarios. El desafío surgió debido a la falta de especificaciones sobre cómo crear y ubicar estos modelos, lo que hizo imposible completar esta parte del servicio según el OBJ-10. Por lo tanto, también se limitó a implementar la gestión de suscripciones.

A pesar de las limitaciones en Data Management y ML Model Provision, pudimos dar funcionalidad al servicio de Events Subscription. Este servicio ahora se utiliza para enviar notificaciones basadas en información obtenida de una base de datos MongoDB. Esto nos permitió mantener la funcionalidad esencial del servicio de Events Subscription, cumpliendo con los objetivos relacionados OBJ-1 y OBJ-2.

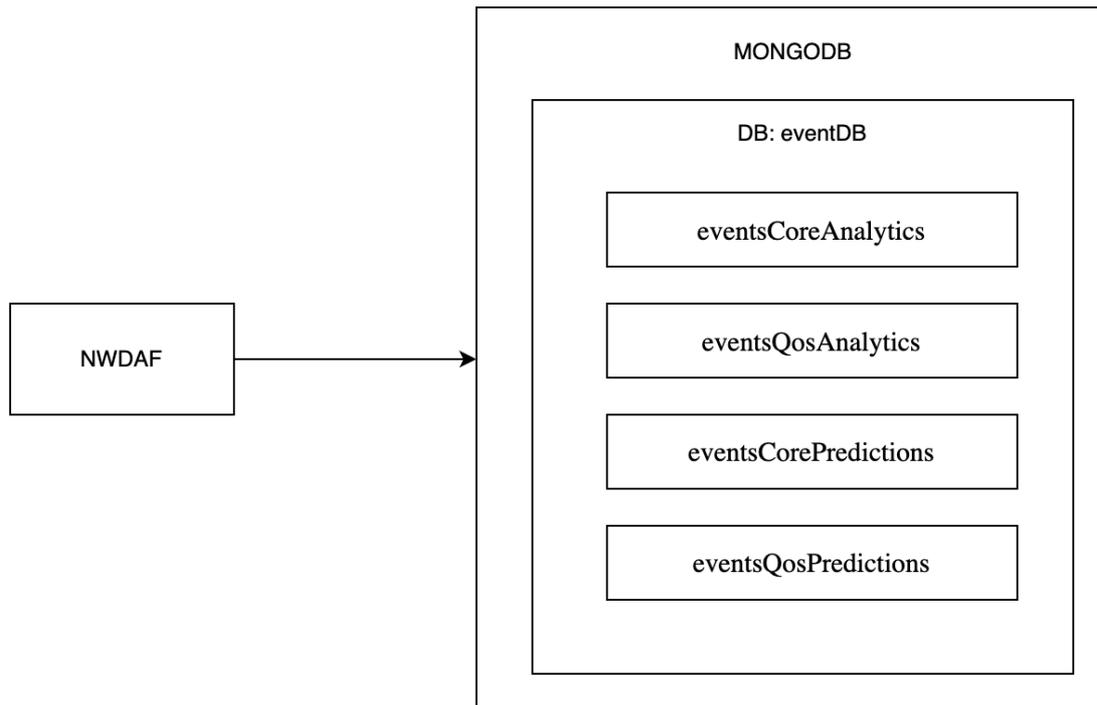


Figura 23. Base de datos "eventDB"

Con esto, el proyecto utilizara dicha base de datos llamada "eventsDB" que contiene las siguientes colecciones de datos:

- eventsCoreAnalytics: Contiene datos analíticos esenciales relacionados con el evento NF_LOAD.
- eventsQosAnalytics: Almacena datos analíticos necesarios para el evento QOS_SUSTAINABILITY.
- eventsCorePredictions: Contiene datos predictivos específicos para el evento NF_LOAD.
- eventsQosPredictions: Almacena datos predictivos relevantes para el evento QOS_SUSTAINABILITY.

Estas colecciones están organizadas con documentos que proporcionan información detallada con una granularidad de 15 minutos. Esto nos permite acceder a todos los datos necesarios para realizar notificaciones a través del servicio Events Subscription, específicamente para los eventos NF_LOAD y QOS_SUSTAINABILITY.

4.11 Envío de notificaciones

El servicio de Events Subscription tiene la responsabilidad de enviar notificaciones periódicas cuando un consumidor se suscribe para recibir datos. En el cuerpo de la suscripción se especifica la URL a la cual deben enviarse las notificaciones, así como el intervalo de tiempo entre cada envío.

Para manejar esta tarea en segundo plano, hemos implementado un scheduler utilizando Flask-APScheduler. Esta extensión para Flask integra el programador de tareas APScheduler de manera eficiente y sencilla, ideal para automatizar trabajos periódicos o basados en eventos dentro de la aplicación Flask. Flask-APScheduler se combina perfectamente con MongoDB para persistir y almacenar las tareas

programadas. Al guardar las tareas en MongoDB, aseguramos que persistan incluso después de reiniciar la aplicación, permitiendo una gestión eficaz a lo largo del tiempo.

El primer paso es inicializar el scheduler junto con la aplicación Flask en el archivo `__main__.py`. Durante esta inicialización también establecemos la conexión con MongoDB y creamos una nueva colección para almacenar las tareas programadas.

Con esta configuración en su lugar, ajustamos la administración de las suscripciones para crear una nueva tarea cada vez que se registre una nueva suscripción. Además, nos aseguramos de actualizar o eliminar la tarea correspondiente cuando se modifique o elimine la suscripción respectiva.

Cada tarea programada se ejecuta periódicamente según el intervalo especificado en la suscripción de Events Subscription, llamando a una función específica con los datos de la suscripción como parámetros. Además de almacenar los detalles de la tarea en MongoDB, también guardamos el ID de la tarea asociada a cada suscripción para facilitar su gestión y mantenimiento.

Este enfoque garantiza que las notificaciones se envíen de manera precisa y oportuna, cumpliendo con los requisitos del servicio Events Subscription de manera efectiva y eficiente.

La función que ejecutan las tareas programadas sigue una serie de pasos estructurados:

1. Comprobación del "flag" de envío:

Primero, se verifica el parámetro de la suscripción que indica si se deben recibir notificaciones. Este "flag" puede estar configurado en estado "ACTIVATE" para indicar que las notificaciones están activas y deben ser enviadas.

2. Obtención de los datos del evento:

Si el "flag" está activado, se procede a obtener los datos del evento requerido según la suscripción. Estos datos son recuperados de las colecciones de la base de datos "eventDB". Cada suscripción puede especificar qué tipo de datos o eventos desea recibir notificaciones.

3. Creación de la notificación:

Una vez obtenida toda la información necesaria del evento, se transforma y estructura para cumplir con las especificaciones del estándar. Esta transformación asegura que la notificación esté correctamente formateada para su envío.

4. Envío de la notificación:

Con la notificación creada y formateada, se realiza una solicitud POST a la URL suministrada en la suscripción. Se espera recibir un código de respuesta 204, que indica que la petición ha sido exitosa y que la notificación se ha entregado correctamente.

Este proceso se repite automáticamente cada X tiempo, donde X es el periodo especificado en la configuración de la suscripción. Si durante la ejecución de este proceso surge algún error al acceder a la información en las colecciones de la base de datos o si la petición de envío de notificación no se completa satisfactoriamente, el "flag" de envío se cambia a "DEACTIVATE". Esto detiene el envío de notificaciones, y la suscripción puede ser modificada para activar nuevamente el envío cuando se resuelvan los problemas.

Con la finalización del servicio de Events Subscription, hemos logrado cumplir con los objetivos: OBJ-1 y OBJ-2. Ahora podemos crear suscripciones y obtener datos de manera efectiva utilizando este servicio.

4.12 Servicio Analytics Info

Para el servicio de Analytics Info, a diferencia de los mencionados anteriormente que requieren suscripciones para recibir datos periódicamente, los usuarios pueden realizar peticiones directas y recibir información inmediata. Sin embargo, nos enfrentamos al desafío recurrente de obtener los datos necesarios para responder a estas solicitudes.

Como solución, hemos adoptado un enfoque similar al utilizado para el servicio de Events Subscription. Es decir, utilizamos la base de datos "eventsDB" para acceder a las colecciones relevantes y obtener los datos necesarios. Esto nos permite garantizar que la información solicitada esté disponible de manera oportuna y eficiente, utilizando la misma infraestructura robusta de MongoDB para el almacenamiento y gestión de datos.

Este enfoque no solo facilita la implementación del servicio de Analytics Info, sino que también asegura la coherencia en la obtención y entrega de datos críticos, cumpliendo con los requisitos del proyecto de manera efectiva.

En el servicio de Analytics Info, se utilizan varios parámetros en las solicitudes para precisar qué datos se desean obtener:

- **Event_id:** Este parámetro identifica el tipo específico de evento del cual se solicitan análisis o predicciones. Por ejemplo, los eventos pueden ser NF_LOAD (carga de función de red) o QOS_SUSTAINABILITY (sostenibilidad de calidad de servicio).
- **Ana_req:** Contiene información crucial para la generación y reporte de análisis y predicciones. Incluye detalles como el intervalo de fechas para el cual se requieren los análisis, el número máximo de documentos que deben ser devueltos, y otros criterios relevantes para la solicitud.
- **Event_filter:** Este parámetro permite al consumidor especificar qué tipo de análisis desea obtener dentro del evento seleccionado. Por ejemplo, para el evento NF_LOAD, se podría especificar el tipo específico de función de red (NF) del cual se desean obtener los resultados analíticos.
- **Supported_features:** Se utiliza como filtro para determinar si el evento seleccionado está soportado o no por el sistema. Proporciona información crucial sobre las capacidades disponibles para el evento en cuestión, lo cual es fundamental para tomar decisiones informadas sobre qué eventos se pueden consultar.
- **Tgt_ue:** Identifica el equipo de usuario objetivo para el cual se desean obtener los análisis. Este parámetro es útil para personalizar los resultados según el equipo específico involucrado en el contexto del evento seleccionado.

En cada solicitud al servicio de Analytics Info, es obligatorio especificar el Event_id y dependiendo del evento seleccionado, algunos de los otros parámetros pueden ser obligatorios para refinar y especificar los detalles exactos de la consulta.

Una vez realizada una petición al servicio de Analytics Info, se sigue un proceso estructurado para determinar y responder con los datos requeridos:

1. Soporte del Evento: Se verifica inicialmente si el evento solicitado y los datos requeridos son admitidos por el NWDAF. En este caso, solo se aceptan los eventos NF_LOAD y QOS_SUSTAINABILITY.
2. Tipo de Datos Solicitados: Se determina si se solicitan análisis o predicciones basándose en las fechas proporcionadas en el parámetro "ana_req", específicamente en "start_ts" y "end_ts":
 - Si ambas fechas son anteriores a la fecha actual, se requieren análisis.
 - Si ambas fechas son posteriores a la fecha actual, se solicitan predicciones.
 - Si las fechas son mixtas (pasado y futuro), se genera un error indicando que la petición no puede completarse.
3. Cumplimiento de Requerimientos del Evento: Una vez determinado el tipo de datos solicitados, se verifica si la solicitud cumple con los requisitos específicos del evento seleccionado.
4. Obtención de Datos: Si la solicitud es válida y cumple con todos los criterios, se procede a obtener los datos requeridos de las colecciones correspondientes en la base de datos "eventDB".
5. Transformación y Respuesta: Los datos obtenidos se transforman según las especificaciones del servicio para asegurar que cumplan con los formatos y estructuras necesarias. Luego, estos datos transformados se envían como respuesta al consumidor que realizó la petición.

Con estos pasos implementados, se completa el servicio de Analytics Info de manera efectiva, cumpliendo con los objetivos establecidos (OBJ-3, OBJ-4, OBJ-5 y OBJ-6).

Esto garantiza que los usuarios puedan acceder de manera eficiente a los análisis y predicciones relevantes, utilizando la infraestructura robusta de la base de datos para gestionar y entregar los datos solicitados de manera oportuna.

5 Validación de servicios

Una vez finalizado el prototipo de NWDAF para los eventos de NF_LOAD y QOS_SUSTAINABILITY, el siguiente paso crucial es validar si se han cumplido los objetivos establecidos para el proyecto.

Durante el desarrollo del prototipo, se ha adoptado una metodología iterativa que incluye pruebas continuas de los servicios a medida que se desarrollaban. Para estas pruebas, se ha utilizado Postman, una herramienta de gran utilidad que permite realizar peticiones específicas para cada servicio de manera efectiva.

Además, se han implementado servicios sencillos de escucha que están diseñados para recibir las notificaciones enviadas por el NWDAF cuando se activa una suscripción en cualquiera de los servicios. Estos servicios de escucha se encuentran ubicados en la carpeta "utils" del directorio raíz del proyecto y se desplegarán cada vez que se realice una prueba, garantizando así la correcta recepción y procesamiento de las notificaciones.

El objetivo de estos servicios de escucha es asegurar que las notificaciones sean recibidas y gestionadas adecuadamente, lo cual es fundamental para la validación del prototipo. Esta fase de validación no solo comprobará el cumplimiento de los objetivos, sino que también permitirá identificar y corregir posibles fallos o áreas de mejora, asegurando la robustez y eficiencia del sistema antes de su implementación definitiva.

5.1 Plan de pruebas

Para la validación de los servicios del NWDAF, hemos diseñado un plan de pruebas exhaustivo utilizando Postman. Este plan realiza todas las peticiones importantes necesarias para verificar si se han cumplido los objetivos del proyecto. Las peticiones incluidas en este plan son las siguientes:

Tabla 6. Plan de pruebas

Nombre		create QOS sub
Detalles		Crea una suscripción para el evento QOS_SUSTAINABILITY utilizando el servicio Events Subscription
Método		POST
URL		/nnwdaf-eventssubscription/v1/subscriptions
Código de estado esperado	de	201

Nombre		update QOS sub
Detalles		Actualiza una suscripción ya creada para el evento QOS_SUSTAINABILITY utilizando el servicio Events Subscription
Método		PUT
URL		nnwdaf-eventssubscription/v1/subscriptions/{{QOS_SUSTAINABILITY_ID}}
Código de estado esperado		204

Nombre		delete QOS sub
Detalles		Elimina una suscripción ya creada para el evento QOS_SUSTAINABILITY utilizando el servicio Events Subscription
Método		DELETE
URL		nnwdaf-

	eventssubscription/v1/subscriptions/{{QOS_SUSTAINABILITY_ID}}
Código de estado esperado	204

Nombre	create NF LOAD sub
Detalles	Crea una suscripción para el evento NF_LOAD utilizando el servicio Events Subscription
Método	POST
URL	/nnwdaf-eventssubscription/v1/subscriptions
Código de estado esperado	201

Nombre	update NF LOAD sub
Detalles	Actualiza una suscripción ya creada para el evento NF_LOAD utilizando el servicio Events Subscription
Método	PUT
URL	nnwdaf-eventssubscription/v1/subscriptions/{{NF_LOAD_ID}}
Código de estado esperado	204

Nombre	Delete NF LOAD sub
Detalles	Elimina una suscripción ya creada para el evento NF_LOAD utilizando el servicio Events Subscription
Método	DELETE
URL	nnwdaf-eventssubscription/v1/subscriptions/{{NF_LOAD_ID}}
Código de estado esperado	204

Nombre	create data management sub
Detalles	Crea una suscripción utilizando el servicio Data management
Método	POST
URL	/nnwdaf-datamanagement/v1/subscriptions
Código de estado esperado	201

Nombre	update data management sub
Detalles	Actualiza una suscripción ya creada utilizando el servicio Data Management
Método	PUT
URL	/nnwdaf-datamanagement/v1/subscriptions/{{DATA_MANAGEMENT_ID}}
Código de estado esperado	200

Nombre	delete data management sub
Detalles	Elimina una suscripción ya creada utilizando el servicio Data Management
Método	DELETE
URL	/nnwdaf-datamanagement/v1/subscriptions/{{DATA_MANAGEMENT_ID}}
Código de estado esperado	204

Nombre	create model provision sub
Detalles	Crea una suscripción utilizando el servicio ML Model Provision

Método	POST
URL	/nnwdaf-mlmodelprovision/v1/subscriptions
Código de estado esperado	201

Nombre	update model provision sub
Detalles	Actualiza una suscripción ya creada utilizando el servicio ML Model Provision
Método	PUT
URL	/nnwdaf-mlmodelprovision/v1/subscriptions/{{ML_MODEL_PROVISION_ID}}
Código de estado esperado	200

Nombre	delete model provision sub
Detalles	Elimina una suscripción ya creada utilizando el servicio ML Model Provision
Método	DELETE
URL	/nnwdaf-mlmodelprovision/v1/subscriptions/{{ML_MODEL_PROVISION_ID}}
Código de estado esperado	204

Nombre	Get QOS SUSTAINABILITY analytics
Detalles	Se obtiene información de análisis sobre el evento QOS_SUSTAINABILITY con el servicio Analytics Info
Método	GET
URL	/nnwdaf-analyticsinfo/v1/analytics?
Código de estado esperado	200

Nombre	Get QOS SUSTAINABILITY prediction
Detalles	Se obtiene predicciones de análisis sobre el evento QOS_SUSTAINABILITY con el servicio Analytics Info
Método	GET
URL	/nnwdaf-analyticsinfo/v1/analytics?
Código de estado esperado	200

Nombre	Get NF LOAD analytics
Detalles	Se obtiene información de análisis sobre el evento NF_LOAD con el servicio Analytics Info
Método	GET
URL	/nnwdaf-analyticsinfo/v1/analytics?
Código de estado esperado	200

Nombre	Get NF LOAD prediction
Detalles	Se obtiene predicciones de análisis sobre el evento NF_LOAD con el servicio

	Analytics Info
Método	GET
URL	/nnwdaf-analyticsinfo/v1/analytics?
Código de estado esperado	200

Además, se utilizan variables de entorno que almacenan información crucial para las peticiones, como la dirección IP donde se ubica el NWDAF y el puerto de escucha correspondiente. Entre estas variables, también se puede especificar la URL a la cual se enviarán las notificaciones de las suscripciones.

Cada vez que se genera una suscripción, el ID de dicha suscripción se almacena en las variables de entorno una vez que se valida que la petición ha sido resuelta correctamente. Esto permite la modificación y eliminación de la suscripción en peticiones posteriores, facilitando una gestión eficiente y dinámica de las suscripciones activas en el sistema.

Para verificar la llegada de las notificaciones con la información solicitada, se puede utilizar el servicio ubicado en la carpeta "utils". Este servicio está diseñado para comprobar que las notificaciones se reciban de manera adecuada y contengan los datos esperados, asegurando así la correcta funcionalidad del sistema de notificaciones del NWDAF.

La implementación y uso de estas variables de entorno mejoran significativamente la flexibilidad y eficiencia del proceso de validación y monitoreo, permitiendo adaptaciones rápidas y precisas a diferentes configuraciones y entornos de prueba. Esto asegura que todas las notificaciones y suscripciones funcionen como se espera, contribuyendo al éxito general del proyecto.

5.2 Resultados

Una vez creado el plan de pruebas con Postman, es momento de comprobar si todos los servicios funcionan correctamente y cumplen con los objetivos propuestos.

Para iniciar el plan de pruebas, levantamos el servicio de la carpeta "utils" para poder recibir las notificaciones de las suscripciones realizadas en el servicio de Events Subscription. En las variables de entorno de Postman, debemos especificar la IP y el puerto donde se ubica el NWDAF, además de la URL en la que se encuentra escuchando el servicio de la carpeta "utils".

Con esto realizado, podemos ejecutar la colección completa de Postman para comprobar si todas las peticiones se completan satisfactoriamente. Podemos omitir las comprobaciones de las peticiones de borrado para verificar que llegan las notificaciones al servicio de la carpeta "utils", o no omitirlas y comprobar que funcionan correctamente.

El resultado de las peticiones es el siguiente:

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	NWDAF	1	678ms	16	10 ms

RUN SUMMARY

					1
▶	POST	create QOS sub	1	0	
▶	PUT	update QOS sub	1	0	
▶	DELETE	delete QOS sub	1	0	
▶	POST	create NF LOAD sub	1	0	
▶	PUT	update NF LOAD sub	1	0	
▶	DELETE	delete NF LOAD sub	1	0	
▶	POST	create data management sub	1	0	
▶	PUT	update data management sub	1	0	
▶	DELETE	delete data management sub	1	0	
▶	POST	create model provision sub	1	0	
▶	PUT	update model provision sub	1	0	
▶	DELETE	delete model provision sub	1	0	
▶	GET	get QOS SUSTAINABILITY analytics	1	0	
▶	GET	get QOS SUSTAINABILITY prediction	1	0	
▶	GET	get NF LOAD analytics	1	0	
▶	GET	get NF LOAD prediction	1	0	

Figura 23. Resultados de la colección de Postman

Después de realizar las pruebas correspondientes, confirmamos que todas las peticiones cumplen con éxito los criterios establecidos para el proyecto. Por lo tanto, podemos asegurar que se han alcanzado los objetivos planteados. Todas las pruebas han sido superadas de manera satisfactoria, demostrando que el proyecto está en condiciones de cumplir con sus metas.

6 Conclusiones

Durante el desarrollo del proyecto, hemos destacado la importancia del NWDAF como una función crucial en la mejora futura de la calidad de red para los usuarios. Específicamente, nos hemos enfocado en la implementación de las funciones relacionadas con los eventos NF_LOAD y QOS_SUSTAINABILITY, siguiendo meticulosamente las directrices establecidas por la 3GPP.

Si bien adherirse a estas especificaciones ha facilitado considerablemente el desarrollo y el diseño, es importante destacar que algunas áreas no están completamente definidas y están sujetas a cambios continuos. Esta flexibilidad nos ha permitido abordar creativamente las partes no especificadas y prepararnos para posibles ajustes en futuras actualizaciones de las especificaciones.

A lo largo del proyecto, nos hemos enfrentado a varios desafíos, entre los cuales el más significativo fue la obtención de datos para las notificaciones de servicios, que inicialmente representó un obstáculo considerable. Sin embargo, hemos logrado resolver parcialmente este problema.

A pesar de los contratiempos, podemos decir que he alcanzado satisfactoriamente los objetivos principales del proyecto. Quedan pendientes únicamente dos objetivos de baja prioridad, cuya viabilidad estaba condicionada por la incertidumbre sobre su realización.

En resumen, el proyecto ha representado un paso crucial hacia la implementación efectiva del NWDAF, destacando la capacidad para adaptarme a los desafíos en curso y para encontrar soluciones innovadoras en un entorno dinámico y cambiante. A través de este trabajo, he fortalecido mi entendimiento de las necesidades de red emergentes y he sentado las bases para futuras mejoras y desarrollos.

7 Abreviaturas

NWDAF	Network Data Analytics Function
3GPP	3rd Generation Partnership Project
NF	Network Function
SMF	Session Management Function
UPF	User Plane Function
PCF	Policy Control Function
AMF	Access and Mobility Management Function
OAM	Operations, Administration, and Maintenance
AF	Application Function
AnLF	Analytics Logical Function
MTLF	Model Training Logical Function

8 Referencias

- ^[1] 3GPP – the mobile broadband standard. (n.d.). 3GPP. Retrieved July 1, 2024, from <http://3gpp.org>
- ^[2] API Documentation & Design Tools for Teams. (n.d.). Swagger.io. Retrieved July 1, 2024, from <https://swagger.io/>
- ^[3] de Gregorio, J. (n.d.). *5GC APIs at Rel-17*.
- ^[4] Salario para Programador en España - Salario Medio. (s/f). Talent.com. Recuperado el 3 de julio de 2024, de <https://es.talent.com/salary?job=programador>
- ^[5] MongoDB: The developer data platform. (n.d.). MongoDB. Retrieved July 1, 2024, from <https://www.mongodb.com/>
- ^[6] nginx. (n.d.). Nginx.org. Retrieved July 1, 2024, from <https://nginx.org/en/>
- ^[7] OpenAPI generator. (n.d.). Openapi-Generator.Tech. Retrieved July 1, 2024, from <https://openapi-generator.tech/>
- ^[8] Postman API platform. (n.d.). Postman.com. Retrieved July 1, 2024, from <https://www.postman.com/>
- ^[9] Ratliff, S. (2022, May 10). *Docker: Accelerated container application development*. Docker. <https://www.docker.com/>
- ^[10] Salazar, J. A. R. (2021, October 15). *NWDAF: la piedra angular de 5G*. LinkedIn.com. <https://www.linkedin.com/pulse/nwdaf-la-piedra-angular-de-5g-jos%C3%A9-alberto-rodr%C3%ADguez-salazar/>
- ^[11] Welcome to. (n.d.). Python.org. Retrieved July 1, 2024, from <https://www.python.org/>
- ^[12] Welcome to flask — flask documentation (3.0.X). (n.d.). Palletsprojects.com. Retrieved July 1, 2024, from <https://flask.palletsprojects.com/en/3.0.x/>
- ^[13](N.d.-a). 3gpp.org. Retrieved July 1, 2024, from <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579>
- ^[14](N.d.-b). 3gpp.org. Retrieved July 1, 2024, from <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3355>
- ^[15] (N.d.-c). Ericsson.com. Retrieved July 1, 2024, from <https://www.ericsson.com/en/core-network/5g-core/network-data-analytics-function>
- ^[16] Marappan, S. K. (2019, agosto 22). *5G Analytics - NWDAF explained*. LinkedIn.com. <https://www.linkedin.com/pulse/5g-analytics-nwdaf-satheesh-kumar-marappan/>