

Universidad de Los Andes
Facultad de Ingeniería
Escuela de Sistemas
Grupo CEMISID

Trabajo Especial de Grado:
**MIDDLEWARE PARA AMBIENTES INTELIGENTES BASADO EN
NANOSENSORES**

Autor:
Alberto Jose López Pacheco

Tutor:
Jose Aguilar

Septiembre, 2023

Resumen

En este Trabajo de Grado se presenta el desarrollo de un middleware para nanoestructuras tales como nanosensores y nanoactuadores, que ejecutan tareas específicas en función del contexto donde se despliega. El middleware autónomo y reflexivo para el manejo de nanosensores y nanoactuadores, llamado ARMNANO (por sus siglas en inglés, *Autonomic Reflective Middleware for the management of nanosensors*), está compuesto por 2 niveles horizontales, que son el nivel base y el nivel meta, y un nivel vertical, que es el Sistema Inteligente de Análisis de Datos (DASS). El nivel base contiene todos los componentes físicos, incluyendo los nanoensamblados y las micro y macroestructuras, mientras que el nivel meta contiene todas las capas lógicas, que a su vez incorporan agentes. En el nivel meta, el middleware ARMNANO incorpora 3 componentes, la Capa Lógica de nanosensores (NSLL), La Capa Lógica de Nanoactuadores (NASLL), y el Nanoagente para el Manejo de las Comunicaciones (CmNA). Estos componentes lógicos gestionan las funcionalidades de los nanosensores y nanoactuadores en el despliegue de ARMNANO en sus diferentes aplicaciones en áreas como la salud, el hogar, entre otras. CmNA es el más complejo, e incorpora un protocolo para detección de fallas que establece las reglas para que ARMNANO pueda auto-observarse y auto-diagnosticarse, un protocolo para incorporación de nanodata, y más importante aún, un protocolo para autenticación de nanodatos. De esta forma, solamente un dato autenticado pasará al DASS para la aplicación de técnicas de aprendizaje automático para el análisis de la nanodata. Por su lado, el DASS, está definido por 3 ciclos autónomos, los cuales son, un ciclo para acondicionar el despliegue de ARMNANO (CTM, por sus siglas en inglés), otro ciclo para optimizar su funcionamiento (OTM, por sus siglas en inglés), y otro ciclo para fijar en tiempo real el comportamiento de ARMNANO (FTM, por sus siglas en inglés), los cuales se ejecutan en ese mismo orden. ARMNANO ha sido usado en un sistema de análisis del hígado para medir cuantos nanosensores se utilizan en la determinación de su status y donde en el hígado es más apropiado insertarlos para hacer su evaluación en cuanto a daños, dando resultados prometedores en el contexto de aplicaciones médicas

Índice de Contenido

Capítulo 1. Introducción	6
1.1. Motivación.....	7
1.2. Planteamiento del Problema.....	7
1.3. Objetivos.....	8
1.4. Antecedentes	9
1.5. Alcance y Limitaciones.....	11
1.6. Organización de la Tesis	12
Capítulo 2. Marco Teórico	14
2.1. Nanotecnología	15
2.2. Ambientes Inteligentes.....	17
2.3. Sistemas Middleware.....	21
2.4. Computación Autónoma.....	23
Capítulo 3. Entornos Computacionales de soporte a la nanotecnología	26
3.1. Middleware Reflexivo Autónomo para el Manejo de Nanodispositivos en un Ambiente Inteligente.....	27
3.2. Sistema de Manejo de Nanocomunicación para Ambientes Inteligentes	35
3.3. Un Modelo Ontológico para un Middleware que gestiona Nanodispositivos en un Ambiente Inteligente.....	44
Capítulo 4. Casos de Estudio en Ambientes nanotecnológicos	51
4.1. Sistema Integral de Análisis de Datos en un Middleware Basado En Nanodispositivos (ARMNANO).....	52
4.2. Caso de estudio de Autenticación de nanodatos del Middleware de Manejo de Nanodispositivos	55
4.3. Un Sistema Inteligente de Análisis De Datos para el Despliegue Óptimo de Nanosensores en el Contexto de una Aplicación de eSalud	62
Capítulo 5. Conclusiones y trabajos futuros.....	71
5.1. Conclusiones	72
Referencias.....	74

Índice de Figuras

Figura 1.1. Middleware Reflexivo.....	22
Figura 1.2. Modelo MAPE+K.....	24
Figura 3.1. Arquitectura ARMNANO.....	28
Figura 3.2. Ciclo Autonomico en modo CTM (Ver ref. [3]).....	31
Figura 3.3. Ciclo Autonomico en modo OTM (Ver ref. [3]).....	33
Figura 3.4. Ciclo Autonomico en modo FTM (Ver ref. [3]).....	34
Figura 3.5. Componentes de la Capa CmNA (Ver ref. [1]).....	35
Figura 3.6. Esquema de deteccion de fallas en la capa CmNA reconociendo los estados A, B y C en que puede estar el middleware	37
Figura 3.7. Arquitectura del sistema de autenticación de datos.....	40
Figura 3.8. Capa CIM de la Ontología ARMNANO (Ver ref. [2]).....	45
Figura 3.9. Capa PIM de la ontología ARMNANO (Ver ref. [2]).....	46
Figura 3.10. Ontología de la Analítica de datos para la Ontología ARMNANO (Ver ref. [2]).....	47
Figura 3.11. Ontología de La Computación Autonomica de la Arquitectura ARMNANO (Ver ref. [2]).....	48
Figura 3.12. Dimensiones que compila DASS en su colección de información	50
Figura 4.1. A la izquierda vemos el NS inmovilizado en la superficie. A la derecha se observa la punta del fluorómetro incidiendo la longitud de onda en el rango de 100 a 1700 nm. (ver [54])	55
Figura 4.2. Rendimiento del nanosensor experimental (<i>ExpNS</i>) en las soluciones tampón de PBS y aCSF (Ver ref. [4])	57
Figura 4.3. Cálculo en el modo de funcionamiento (modelado con MLE) para el 2 ^{do} módulo en la zona 1. (ver ref. [4])	60
Figura 4.4. Tareas para el ciclo autonomico CTM	64
Figura 4.5. Hígado seccionado en 4 regiones para inyectar los NSs en cada una (ver ref. [5])	69

Índice de Tablas

Tabla 1. Macroalgoritmo para el módulo 1 en la etapa de Entrenamiento. (Ver ref. [4]).....	41
Tabla 2. Macroalgoritmo para el Módulo 1 en la Etapa de Funcionamiento (Ver ref. [4]).....	42
Tabla 3. Macroalgoritmo para el Módulo 2 en la Etapa de Entrenamiento (Ver ref. [4]).....	43
Tabla 4. Macroalgoritmo para el Módulo 2 en la Etapa de Funcionamiento (Ver ref. [4]).....	43
Tabla 5. Primeros 20 datos del dataset de la fluorescencia I/I ₀ medida en los tampones de PBS y aCSF en el rango de concentraciones de 0.1 a 2 μ M	56
Tabla 6. Métrica estadística de las 3 zonas para los buffers de <i>aCSF</i> y <i>PBS</i> .	58
Tabla 7. Métrica de calidad en el caso de estudio 2.....	59
Tabla 8. Valores de MCV y ALKPHOS en el mejor individuo del caso 1	67
Tabla 9. Mejores individuos encontrados utilizando las bases de datos de personas afectadas en el hígado	67
Tabla 10. Los mejores individuos encontrados en el caso 2 para el dataset [6].....	68
Tabla 11. El mejor individuo encontrado en el caso 2 para el conjunto de datos [6] y diferentes pacientes.	70

Capítulo 1. Introducción

1.1. Motivación

Los sistemas inteligentes son un área que ha tenido auge desde hace más de 50 años, sin embargo, han tenido distintos matices vinculados a los avances de la computación. Por otro lado, la analítica de datos para estudiar grandes volúmenes de datos es muy diversa según la tarea requerida, en las que algunas veces se aplica simplemente estadística y en otros sofisticados algoritmos de aprendizaje automático [7]. La nanotecnología, por su parte, permite percibir el contexto involucrando estructuras como nanodispositivos para medir variables en lugares que antes eran inaccesibles. Ejemplos de ello son los nanosensores (NSs) que miden el pH en la sangre [8] [9], la serotonina [10], o el contenido salino en el sudor [10]. Por otro lado, el área de los nanoactuadores (NAs), a pesar de no estar tan desarrollada, supone también una ventaja dada por la selectividad de acciones que se ejecutan a nivel nanométrico. Estas acciones incluyen cortar, destruir, o liberar un compuesto gradualmente, entre otras cosas [11].

Este trabajo de grado abarca los aspectos arriba mencionados para definir un middleware que permita el despliegue de nanodispositivos en ambientes inteligentes. En un sistema de computación distribuida, el middleware es crucial porque permite la gestión de sus componentes, tanto a nivel de software y hardware. Así, la novedad de incorporar nanodispositivos implica que middlewares típicos como SACI [12], ARMISCOM [13], o CARMICLOC [14], cambien radicalmente. Ahora se debe gestionar la incorporación de estos nanocomponentes, definir los agentes que manejen sus despliegues y las condiciones para el despliegue autonómico, entre otras cosas.

Esta tesis contempla la utilización de técnicas de inteligencia artificial que, actuando en un middleware, pueden proveer soluciones a aplicaciones basadas en nanotecnologías en ambientes inteligentes, gestionando el set de NSs y NAs adecuados para condiciones específicas. Esto permitirá aprovechar las propiedades de estos nanoensamblajes, para la medición y manejo de datos, con el objeto de dar soluciones en el entorno que se estudia. La carencia de literatura sobre middlewares para el despliegue de nanodispositivos, justifica el interés de esta tesis, así como también, la novedad de fusionar la captación de datos y utilización de estos datos a nivel nanométrico.

1.2. Planteamiento del Problema

El desarrollo de middlewares es uno de los retos de la computación más importantes, de tal manera de conformar un entorno integrado que garantice la interoperabilidad de los sistemas [14], tal que se puedan proveer servicios de forma autónoma a las aplicaciones

en el contexto donde se despliegan, y brindar servicios de forma apropiada en el Ambiente Inteligente (AmI) donde se ejecuta.

En particular, un middleware para la gestión de nanoestructuras como el propuesto en la presente tesis doctoral es un reto, dado por el orden de la nanoescala que se pretende manejar. Estudios como el de la presente tesis doctoral son una novedad dentro de lo reportado en la literatura, y muy pocas publicaciones anteriores han abordado la fusión entre nanotecnología y middlewares para AmIs [8]. Un middleware como el que se avizora en esta tesis doctoral busca resolver las dificultades vinculadas a la gestión de la nanotecnología en AmIs, de tal manera de tener acceso a contextos intrincados (tales como el cuerpo humano, un avión, o un arma), para ámbitos tan diversos tales como salud, militar y telecomunicación, entre otras.

Los sistemas basados en nanodispositivos requieren que se optimicen aspectos como la transmisión de nanoinformación, corrección de fallas en los nanodispositivos, explotación de los nanodatos, entre otras cosas. Adicionalmente, poco se conoce de middlewares adaptados en una estructura real, dando soluciones a través un nodo compuesto de nanosensores o nanoactuadores. Por otro lado, la constitución nanométrica limita los modos físicos en que estos nanodispositivos se interconectan, migrando a modos de comunicación del tipo radiación electromagnética en la banda Terahertz (THz), comunicación molecular, o comunicación utilizando el fenómeno de fluorescencia, donde cada uno tiene sus distintas ventajas y limitaciones en la implementación. Esos y otros aspectos de la nanoescala, deben ser considerados para la construcción de un middleware autónomo de gestión de nanodispositivos como el que desarrollaremos en esta Tesis Doctoral. Estas son las dificultades más relevantes a resolver por nuestro middleware.

1.3. Objetivos

Objetivo General

Especificar un middleware débilmente acoplado a un AmI, para la gestión de los nanodispositivos presentes.

Objetivos Específicos

1. Construir un sistema de gestión de los nanodatos medidos con nanosensores, en un AmI.
2. Concebir un sistema de comunicación entre nanodispositivos que se integre en un

- AmI.
3. Diseñar un sistema autonómico de tareas de análisis de nanodatos, para casos de estudio.
 4. Integrar el sistema autonómico en un middleware de gestión de nanodatos en un AmI.

1.4. Antecedentes

En esta sección, se hará mención a los trabajos previos que sustentan el desarrollo de middlewares similares para entornos con nanomateriales. Particularmente, para esta propuesta interesan los trabajos vinculados a NSs y NAs, y en especial, que usen middleware para la gestión de los nanodatos intercambiados entre ellos. En este sentido, los trabajos previos son los siguientes.

Hesselbach y col., en [15] definen los componentes de un middleware que incorpora nanodispositivos, y en particular, definen las nanoredes que permiten que se comuniquen entre sí a través de diferentes mecanismos. Ellos destacan entre dichos mecanismos las *Redes Definidas por el Software* (en inglés, SDN, *Software Defined Networking*), que consiste en un conjunto de técnicas relacionadas con el área de redes computacionales que busca la ejecución de servicios de comunicación de forma precisa y dinámica para administrar la red. También está el método de *Virtualización de las Funciones de Red* (en inglés, NFV, *Network Function Virtualization*), el cual permite ejecutar parte de la gestión de la red en la nube, lo que facilita la adaptación y escalabilidad del sistema. A partir de esos conceptos (SDN y NFV), definen el concepto del *Internet de las NanoCosas* (en inglés, IoNT, *Internet of NanoThings*), donde las cosas que se comunican son nanodispositivos, los cuales transmiten nanodatos a través en la web. Estas cosas (nanodispositivos) pueden ser nanoantenas, nanoprocesadores, nanoenrutadores, nanosensores, nanoactuadores, etc. Estos autores destacan que el middleware se compone esencialmente de 3 capas, que son, la capa de aplicación, la capa de control y la capa de infraestructura. Respecto a la IoNT, los nanodispositivos se compone de una antena que transmite las señales directamente a la web. Así, los nanodatos se almacenan, se procesan y se intercambian en la web, desencadenando comandos para la red de nanoactuadores.

Chong y col. [16] han establecido un método para monitorear el tiempo de adquisición de datos en un sistema con nanodispositivos, el cual está limitado por el hecho de que tiene un rango para transmitir información muy pequeña (0,06 a 10 THz). Para dicho propósito se propusieron 2 estrategias, basadas en tasa de muestreo baja (en inglés, LSR, *Low Sampling Rate*) o en similitud máxima (en inglés, ML, *Maximum Likelihood*). En [17],

Akkari y col. proponen estrategias para controlar los errores en las nanoredes empleando un análisis de capas entrecruzadas, en un middleware que se comunica en la banda THz. En la *capa física* se considera que las nanomáquinas se comunican implementando la modulación on-off basada en la transmisión de pulsos en el rango de femtosegundos. En la *capa de enlace* se considera que las nanomáquinas acceden a un canal de transmisión de una forma descoordinada, con lo que es necesario controlar la transferencia de pulsos entre distintos nodos en la arquitectura. Abbasi y col. [18] han planteado una revisión de las capas físicas de un middleware para la gestión de la nanocomunicación. Su investigación contempló la instanciación de modelos actuales para controlar la transferencia de datos, la adecuación de varios paradigmas de nanocomunicación, la resolución de problemas de diseño de nanoantenas como dispositivos para controlar la transmisión de datos, entre otras cosas. Kokkonien y col. [19] abordaron la sincronización de la colección de nanodatos en nanoestructuras bajo distintos fenómenos de dispersión. Por un lado, consideraron la dispersión individual (un evento), o la dispersión múltiple (varios eventos) de las nanopartículas. De esta forma, optimizaron la capa física para aprovechar las ventajas de la banda THz en el diseño del canal que transmite los nanodatos.

Fischer S. y col. [10] han propuesto un middleware que involucra nanocomunicación a través de redes instanciadas en el IoNT. La nanocomunicación que ellos plantean se sustenta en redes intracorporales que envían información a arquitecturas de redes de nanosensores corporales (en inglés, BANNET, *Body Area NanoNetworks*). Para esta comunicación, las nanoredes pueden emplear distintos modos de transmisión de datos, tales como comunicación molecular, acústica, de radio frecuencia, con señales que se envían en el rango Terahertz. Estos autores disciernen aspectos importantes a considerar en la implementación con un middleware basado en nanoredes, las cuales incluyen *esquemas de direccionamiento*, donde se establece la ontología para transmitir datos; *nanocomunicación en tiempo real*; *latencia baja*, y *análisis de la factibilidad de la nanocomunicación*. La optimización de estos 4 factores desencadena en la instanciación de una red de nanosensores optimizada.

Una red de nanonodos es propuesto en Granelli F. y col. [20], donde estudian los aspectos relacionados con el intercambio de información en una nanored. Esta investigación apunta a la eficiencia en el manejo comunicacional de nanosensores en una red. Estos nanosensores están interconectados a través de un programa que planifica la frecuencia de conexión, que permite mitigar los efectos atenuantes de la señal cuando ocurre la movilidad de los nanodispositivos o la colisión de paquetes enviados durante las transmisiones. El objetivo de la arquitectura es mejorar el rendimiento de la energía total consumida. Así, proponen un sistema consciente del gasto energético, para un entorno flexible de arreglo de nanosensores.

La integración de datos en la nanotecnología es crucial debido a la creciente información que cada segundo se origina tanto en investigación básica como en aplicaciones específicas, en campos como el biomédico, ciencia de los materiales, agricultura, entre otras. Karcher C. y col. [21] han estudiado una estrategia formal para curar los datos que aparecen en las distintas aplicaciones de los nanomateriales, vinculados a la completitud de los nanodatos, la calidad de los nanodatos, entre otros. Los aspectos considerados en la integración de datos fueron el uso de vocabularios comunes (ontologías), recuperación de datos basada en ontologías, herramientas para analizar y visualizar nanodatos, entre otras cosas. Lograron establecer un planteamiento ontológico de repositorios de datos, y establecieron herramientas para manejar aspectos organizacionales de los datos en la nanoinformática.

El trabajo de Bunakov y col. [22] reporta el esfuerzo del Proyecto de Nanoestructuras y Análisis Fino (en inglés, NFFA, *Nanostructures Foundries and Fine Analysis*) en el diseño de metadata. El propósito principal es reflejar el modelo de metadatos de la NFFA, el cual se sustenta en varios pilares, tales como instalaciones disponibles, instrumentación, muestreo y, mediciones de nanodatos, la organización de los nanodatos y la analítica de nanodatos. En un trabajo realizado por Bunakov y col. [23] explican la instanciación del NFFA en el ciclo de vida de los nanodatos, en el contexto de la nanociencia. Muestran el método para recuperar los metadatos, basado en data recolectada según acciones de los usuarios, instrumentos para monitorear variables del contexto, entre otras cosas.

Los sistemas automatizados en nanotecnología están en auge. Kornilina M. y col. [24] han planteado al respecto una nube de servicios diseñada para analizar volúmenes enormes de nanodatos. Los problemas específicos donde se lo aplicaron incluyen los campos de nanoelectrónica, y nanotecnología laser. Por otro lado, un nuevo modo de comunicación para la arquitectura multimodales ha sido establecido por Liaskos C. y col. [25]. Este sistema instancia una red inalámbrica de nanonodos (nanodispositivos) para áreas como la medicina, procesos industriales, o monitoreo ambiental. Esta investigación estableció un método de diseminación de nanodatos optimizado, que se sustenta en la propia simetría del arreglo de nanodispositivos, los patrones que estos describen, y en su escalabilidad. La ventaja de este proceso radica en la construcción de una arquitectura con una capa física ultra escalable, de baja complejidad, que se optimiza.

1.5. Alcance y Limitaciones

Dentro del alcance de la presente tesis doctoral pueden mencionarse los siguientes:

- Los casos de estudio implementados en la presente tesis emplearon conjunto de nanodatos extraídos de otras publicaciones; o fueron solicitados a otros investigadores que generaron nanodatos medidos con nanosensores.
- ARMNANO ejecutó su condición autonómica con la incorporación de ciclos de tareas de análisis de nanodatos. La capacidad iterativa y la simplicidad de las tareas justifica la habilidad del sistema para establecer su autonomía.
- El middleware ARMNANO es autónomo porque observa el entorno donde se despliega y define acciones a ejecutar sobre él; es reflexivo porque invoca tareas de analítica de datos; y es innovador porque incorpora nanocomponentes que ejecutan tareas en espacios micro.
- El despliegue de ARMNANO fue realizada en entornos genéricos para evaluar su utilidad en contextos variados.

Las limitaciones de la presente tesis doctoral son las siguientes:

- La poca disponibilidad de una cantidad de nanodatos significativa que permita crear modelos basados en nanodatos.
- La falta de nanorutadores que permitan estudiar el tratamiento de ruidos, fallas de comunicación, en las tareas de transmisión de nanodatos.
- La falta de nanosensores para analizar la captura, transmisión y tratamiento de nanodatos en tiempo real, para optimizar aplicaciones reales basadas en nanodatos.

1.6. Organización de la Tesis

Este Trabajo de Grado está organizado de la siguiente forma. En el capítulo 2 está el marco teórico donde se explican las bases conceptuales de la investigación realizada. Este capítulo 2 se subdivide en varios tópicos que son, Nanotecnología, Ambientes inteligentes, Sistemas Middleware y Computación Autonómica. El capítulo 3, por su parte, contempla 3 publicaciones que destacan la composición de ARMNANO; pero además se explican las diferentes capas que componen a ARMNANO. Luego se describen las bases ontológicas del Middleware ARMNANO. En el capítulo 4 se destacan 3 secciones que corresponden de igual forma a 3 publicaciones que presentan Aplicaciones de ARMNANO. La primera de ellas contempla los 3 ciclos autonómicos que se despliegan en ARMNANO en función del status del contexto observado. Luego se presenta un subcapítulo con las funcionalidades de CmNA para autenticar la nanodata a través de un protocolo de autenticación compuesto de 2 capas. Posteriormente, se resalta una implementación de toda la estructura de ARMNANO enfocado a una aplicación de salud, para lo cual se invoca uno de los ciclos autonómicos denominado CTM. Allí se observa a

ARMNANO en todo su esplendor, destacando la utilidad y pragmatismo en dar soluciones mientras la data se mide en la nanoescala.

Capítulo 2. Marco Teórico

A continuación, presentamos la conceptualización y fundamentos que sustentan esta tesis de grado. En este capítulo se desarrollan los tópicos relacionados con Nanotecnología, Ambientes Inteligentes, Sistemas Middleware y Computación Autónoma.

2.1. Nanotecnología

2.1.1. Definición de nanotecnología

Se refiere al campo de estudio de la materia y los materiales a una escala de 10^{-9}m . La nanotecnología permite implementar tecnologías en esta escala, tomando ventaja de las propiedades de estos materiales. A este nivel, una nanomáquina se define como la unidad funcional más básica, integrada por sus nanocomponentes, disponible para ejecutar tareas simples como medir o actuar. Los modos de comunicación son diferenciados a nivel nanométrico debido a las propiedades de la materia en esta escala. Mientras a nivel macro, los dispositivos intercambian información por bluetooth, o infrarrojo, en la escala nanométrica, existen los modos como Comunicación Electromagnética, Comunicación Molecular o Resonancia Fluorescente [26]. A continuación, se describen los modos de comunicación a nivel nanométrico,

a. Comunicación Molecular: está definida como la transmisión y recepción de información codificada en moléculas. Los transreceptores (unidad encargada de recibir-y-enviar datos) moleculares están integrados a los nanodispositivos para aprovechar las ventajas de su tamaño y dominio de operación [27]. Estos transreceptores están dispuestos a reaccionar con moléculas específicas y liberar otras moléculas, como respuesta a un comando interno, o como respuesta de algún tipo de procesamiento externo. La comunicación molecular también se soporta en la ubicación de transreceptores que detectan el gradiente de migración de una molécula u organismo a nivel nano.

b. Comunicación Nanoelectromagnética: este modo de comunicación está definido como la transmisión y recepción de radiación electromagnética de componentes basados en nanomateriales a base de partículas metálicas u óxidos metálicos [18]. Las propiedades en estos materiales decidirán sobre el ancho de banda específico para la emisión de radiación electromagnética, el tiempo de espera de la emisión, o la magnitud de la potencia emitida dada una cantidad “E” de energía de entrada.

2.1.2. Nanodispositivos

Una arquitectura basada en nanodispositivos posee componentes físicos que ejecutan funciones específicas de forma automática o basada en comandos, a nivel nanométrico. De acuerdo con las características a nivel nanométrico, los nanodispositivos se clasifican como sigue,

Nanonodos: son las nanomáquinas más simples y pequeñas. Ellas se perfilan para ejecutar computación simple, tienen una memoria limitada, y pueden transmitir únicamente en pequeñas distancias; debido a su energía reducida y capacidad de comunicación limitada.

Nanosensores: estos nanodispositivos están facultados para medir y transmitir el valor de dicha medición. En general, se apoyan en los nanorutadores para que el dato alcance distancias más largas o destinos muy específicos. Transmiten de acuerdo con el modo de comunicación en el que la arquitectura está trabajando, esto es, molecular o electromagnético.

Nanorutadores: estos nanodispositivos poseen recursos computacionales más grandes que los nodos y son adecuados para agregar y transmitir información que proviene de los nanosensores. Los nanorutadores pueden o no controlar los nanonodos vía emisión de órdenes directas (on/off, sleep, read value, etc.).

Nanoactuadores: estos nanodispositivos se encargan de ejecutar tareas específicas que son ordenadas por la unidad de procesamiento de información. Esta unidad está conectada a la arquitectura, pero ejecuta el procesamiento fuera y envía, la decisión a través de una estrategia in-bound (de afuera hacia adentro). Las tareas que ejecuta dependen de las capacidades técnicas y de procesamiento con las cuales está provisto este nanodispositivo. En el campo de la salud, por ejemplo, puede cortar tejido orgánico, piel, una arteria, inyectar una dosis de un medicamento, localizar específicamente una estructura biocompatible, etc.

Nanointerfaz: juntan la información que proviene de los nanorutadores, para transmitirla a niveles subsiguientes. Son dispositivos híbridos provistos de un transductor que permite la comunicación entre las instancias que miden y las instancias que procesan la nanodata.

Gateway: este dispositivo es un paso de control de nanodata. Están encargados de detectar la presencia de entes que son observados y reportarlos, tal como ocurre en el modo de Comunicación Molecular. Puede actuar como un observador de gradiente molecular cuando se instancia en el campo de la salud, análisis climatológico, control de calidad en la industria, etc.

2.1.3. Nanoarquitectura

La nanoarquitectura está compuesta de nanomáquinas que ejecutan funciones diversas. Estas nanomáquinas forman redes donde todos los dispositivos están interconectados para generar un clúster funcional en un espacio inteligente. La nanoarquitectura en consecuencia, puede desplegarse in situ, permitiendo un ciclo cerrado de transferencia de información para diagnosticar, analizar y determinar las acciones en el ambiente. A continuación, se explican algunas características de la nanoarquitectura:

- *Canal de los Nanodispositivos:* El canal de banda típico para la transferencia de nanodata es la banda Terahertz dado que su alta frecuencia de 10^{-14} s permite alcanzar niveles de transmisión por el orden de los Terabits por segundo, en distancias muy pequeñas de menos de 1 metro [28].
- *Control de Acceso al Medio* (en inglés, MAC, Medium Access Control): es una capa de acceso que controla el ancho de banda que transmite el canal. En la nanoarquitectura, la banda Terahertz provee anchos de banda más grandes, los cuales resultan en tasas de bits más altas, periodos de transmisión cortos y probabilidad de colisión mínima. A pesar del poder de transmisión bajo de los nanodispositivos, de la alta pérdida de recorrido de la banda Terahertz, y de la limitada energía de las nanomáquinas, el uso del protocolo MAC es necesario para modular el acceso al canal y coordinar las transmisiones entre nanomáquinas.
- *Consumo de Energía de los Nanodispositivos:* una nanomáquina funciona con una nanobatería, que, dado su tamaño, almacena muy poca energía. Esto indica que deben establecerse modos de recarga externos o in situ. Un modo de recarga externa funciona a través de un campo magnético o un campo eléctrico ubicado externamente al set de nanomáquinas, de modo que se recarguen tanto como necesiten. Un modo de recarga in situ, implica la transducción de un proceso local para convertir una fuente de energía cercana en energía eléctrica útil a las nanomáquinas. Por ejemplo, en procesos dentro del cuerpo humano, la nanomáquina puede tomar la energía química de azúcares, o la energía cinética de la sangre fluyendo, o la energía de la tensión muscular, etc. [28].

2.2. Ambientes Inteligentes

2.2.1. ¿Qué es un AmI?

Se define como un sistema sensible, reactivo, y adaptativo a los cambios del contexto [29]. Este sistema es embebido, consciente del contexto (monitorea variables relevantes en función de la línea temporal), personalizado (se adapta a los requerimientos del usuario), adaptativo (es flexible a los cambios y moldea sus capacidades y alcance), y anticipatorio (entiende el contexto y prevé su comportamiento, estableciendo mecanismos autocorrectivos) [30]. Los AmIs se construyen sobre la base de los siguientes paradigmas, Computación Pervasiva, Computación Ubicua, conciencia de Contexto, y Computación

Centrada en el Humano. En resumen, un AmI conforma un sistema inteligente que brinda servicios a una persona (o varias personas) en un espacio delimitado, para lo cual hace uso de recursos específicos [31], lógicos o tangibles. Un sistema AmI, contempla un sistema distribuido, no intrusivo y se apoya en la ejecución de software inteligentes y reactivos a agentes externos [32].

2.2.2. Diseño de un AmI

Para explotar las ventajas del AmI como sistema informático, este debe sincronizar el desempeño de los agentes lógicos y físicos que participan en el sistema. Dada estas características, un AmI necesita responder 5 interrogantes que definan el alcance de su diseño e implementación, que son:

¿Qué?, define las propiedades y características de los hechos que ocurren en el sistema. En el caso de un sistema con nanodispositivos, el AmI debería, por ejemplo, identificar las variables que mide la red de nanosensores, entre otras.

¿Quién?, identifica los usuarios del sistema y el rol de cada uno. En este sentido, el sistema reconoce, los actores que participan.

¿Dónde?, se encarga de rastrear el lugar donde un usuario o un objeto se localizan geográficamente durante la operación de un sistema. Esto puede requerir una mezcla de tecnologías que combina telecomunicación con informática, etc.

¿Cuándo?, se refiere a la instanciación de las actividades en el tiempo, determinando la secuencia de acciones para construir el hecho en la línea temporal.

¿Por qué?, destaca la capacidad de inferir y entender las intenciones que se encubren detrás de los usuarios. Intenta inferir necesidades, acciones y problemas de los usuarios.

2.2.3. Componentes de un AmI

Los componentes de un AmI deben permitir que el mismo pueda funcionar en diferentes contextos, para lo cual se deben considerar las siguientes capacidades en él: Sensor, Razonar, Actuar, Interactuar. A continuación, se define cada uno de estos parámetros.

Sensar: Los AmIs requieren componentes para medir datos del mundo real. Para ello se diseñan software que perciben el ambiente. Varios tipos de sensores se emplean en la tarea de captura de información. Estos pueden incluir sensores químicos (miden CO₂, NO_x), de humedad (miden cantidad de H₂O en el aire), de luz (mide la cantidad de fotones incidente), de radiación (mide la cantidad de partículas alfa, beta o gamma), de temperatura (mide la cantidad de calor), de velocidad (mide el desplazamiento por unidad de tiempo), etc. En el caso que atañe a la presente tesis doctoral, los sensores propuestos van a estar en escala nanométrica.

Las Redes de Sensores Inalámbricos (RSI) constituyen un área de investigación de mucho interés en la actualidad para AmIs. Las RSI tienen aplicaciones directas en la caracterización de sistemas colectivos y determinación de comportamientos sociales. Estas variables requieren de observadores no invasivos, que reporten instantáneamente, y que alimenten una base central de procesamiento. La base central de procesamiento está compuesta de nodos que les permite intercambiar información con rapidez, mejorar la eficiencia en el manejo de recursos y mantener la subsistencia a largo plazo en la colección de datos.

Razonar: El razonamiento de acuerdo a Bettini C. y col. [33], se define como “la destreza de un AmI en diagnosticar un contexto mediante la realización de operaciones autónomas enraizadas en el análisis de sus datos”. El razonamiento, de acuerdo Patkos T. y col. [34], se aplica al “entendimiento de eventos que indican cambios en el ambiente, y permite describir e interpretar el flujo de estos eventos, con propiedades que varían en el tiempo”.

En [35] el autor define el razonamiento como “la capacidad de interpretar datos” a través de lo cual establece que “el razonamiento deja que los números se expresen por sí mismos, permitiendo que la data revele las tendencias escondidas y genere un modelo descriptivo”. Todas las definiciones anteriores entran en la definición general de razonamiento, entendido como la facultad que permite resolver problemas, extraer conclusiones, aprender de los hechos, para establecer conexiones causales y lógicas entre ellos.

Actuar: Los dispositivos inteligentes están provistos de distintos mecanismos bajo el cual ejecutan comandos en un contexto determinado. Para sistemas nanometrizados se usan nanoactuadores. Por ejemplo, un AmI en el campo de la salud, dispondrá de un nanoactuador que elimine la grasa del torrente sanguíneo, o que seccione una masa fibrosa intracorporal. Un AmI en el campo de la Telecomunicación estará provisto con un sensor que mide la intensidad de la señal y la frecuencia de sintonización, de forma que sintoniza automáticamente si la señal es de baja resolución.

Interactuar: Los ambientes inteligentes se enfocan mucho en la detección de los humanos en el espacio contentivo. Esto permite ubicar la posición de cada humano y monitorear su desplazamiento para ejecutar la correcta aplicación de los servicios a los que se propone el AmI.

La interacción Humano-Maquina (H-M) puede ser de diferentes formas: comando de voz, comunicación en una plataforma común, interacción en la App, entre otras. La información corresponde a órdenes del humano a la máquina, suministro de un input solicitado por la arquitectura, etc.

2.2.4. Aplicaciones de un AmI

Son muchísimos los ámbitos de aplicación de los AmI tales como en el hogar, en los centros de salud, en las instituciones de educación, en la industria armamentística, en las empresas de producción masiva, entre otras. Cada AmI, en función de su aplicación, libera diferentes servicios a los humanos contenidos en el espacio inteligente. [32]. A continuación, se describe algunas aplicaciones de interés,

En la Casa. La instanciación de un AmI en la casa tiene que ver con el concepto de casas inteligentes, donde todos los objetos dentro de la misma están conectados parcial o totalmente. En este rango se incluye para la interconexión de datos, técnicas como IoT o el IoNT para hacer la transferencia de información. Su uso se relaciona con la instanciación de espacios inteligentes. Por un lado, un espacio individual para ofrecer un servicio específico a una o varias personas, como, por ejemplo, medir la azúcar en sangre (cuarto de salud), ofrecer recomendaciones de libros para leer en función de su perfil en redes sociales (cuarto de lectura), o recomendar recetas de comida si la persona quiere cocinar en el hogar (cuarto de cocina). Por otro lado, un AmI establece la interconexión de todos los dispositivos para proveer esos servicios específicos. Esto implica que la casa está provista de distintos componentes físicos que se integran para ofrecer servicios de tal manera de gestionar la casa en aspectos, como la temperatura, cantidad de iluminación adecuada, etc.

En la Salud, en este campo se incluyen todos los AmI orientados al servicio de diagnóstico, recomendación y ejecución de actividades paliativas a nivel de salud en las personas. Este tipo de sistema implica que el espacio donde existe el AmI contiene NSs (y nanodispositivos en general) específicos para la salud. Por ejemplo, para medir variables típicas como la temperatura corporal, grado de sudoración, nivel de hidratación, actividad cerebral. También puede configurarse un sistema que se denomina Redes BANNET, que reside en el cuerpo del paciente para ofrecer una observación individualizada que detecta

variables más sensibles y precisas. Un BANNET puede sincronizarse con el AmI, y generar un diagnóstico más amplio en las personas [36].

2.3. Sistemas Middleware

2.3.1. Middleware Reflexivo

La Reflexión es la habilidad que tiene un programa de autoexaminarse, e incluso de examinar a su entorno inmediato, con la finalidad de autoadaptar su comportamiento según sea necesario [37]. Por su parte, [29] define un Middleware Reflexivo como un sistema distribuido que actúa como capa intermedia entre aplicaciones y servicios, el cual es capaz de cambiar el comportamiento de las aplicaciones de acuerdo a sus necesidades de ejecución, por medio de la autoconciencia y autoreferencia.

Los autores de [38] afirman que un Middleware permite esconder la heterogeneidad de los sistemas distribuidos, soportar modelos avanzados de coordinación entre entidades distribuidas, y realizar la distribución de los cálculos de forma transparente. Las funciones más resaltantes de un Middleware son:

- Esconder la distribución de los usuarios y de las aplicaciones que se ejecutan sobre él.
- Esconder la heterogeneidad de los componentes de hardware, sistemas operativos y protocolos de comunicación.
- Proveer una interfaz estándar de alto nivel para el desarrollo e integración de aplicaciones, de tal manera que una aplicación pueda ser fácil de reusar, portar y mantener la comunicación con otras aplicaciones.
- Suministrar un conjunto de servicios comunes para el desarrollo de funciones en entornos distribuidos, con el objeto de evitar la duplicación de esfuerzos y facilitar actividades colaborativas.

En otras palabras, un Middleware debe proveer al usuario los elementos necesarios para que la tarea de desarrollar aplicaciones se realice de una forma simple, haciendo menos complejo el uso de lenguajes de programación, sistemas operativos y características inherentes al hardware como son la heterogeneidad y distribución. Algunas plataformas de Middleware conocidas son CORBA, J2EE, DCOM, .NET.

Por otro lado, la capacidad de reflexión de un programa se lleva a cabo por medio de dos procesos que son:

Introspección: Es la habilidad del programa de observar y razonar sobre su estado [39]. Durante este proceso no se realizan cambios en la aplicación, solo se efectúa un razonamiento sobre su propia estructura y comportamiento.

Intersección: es la capacidad del programa de cambiar su propio estado de ejecución. Por medio de la intersección es posible modificar la secuencia de ejecución de un sistema, sin que sea necesario realizar cambios en el código de la aplicación [39].

Un Middleware Reflexivo está compuesto por dos o más niveles (ver **Figura 2.1**):

- Nivel base: es donde se ejecutan las funcionalidades propias del sistema (aplicación en ejecución), y contiene los objetos que resuelven el problema dado;
- Nivel meta: implementa la capacidad de reflexión, y verifica que el funcionamiento del sistema (aplicación) sea el esperado o requerido.

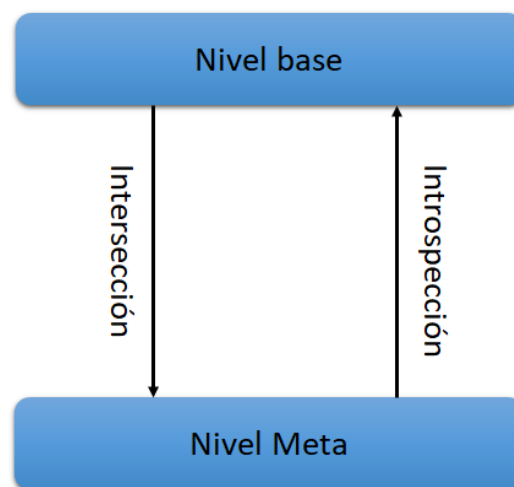


Figura 2.1. Middleware Reflexivo

En el nivel meta es donde se ejecutan los procesos de introspección, mientras que la intersección requiere de los dos niveles [40]. La intersección es realizada en dos fases; en la primera fase la entidad base realiza una operación que es interceptada por el nivel meta. Luego, en la segunda fase la entidad meta toma el control y aplica el razonamiento asociado a la operación pedida en el nivel base. De esta manera, la entidad meta razona sobre lo que está pasando en el nivel base, y en función de esto modifica su estado para adaptarse a los requerimientos del nivel base. Posteriormente, los resultados se regresan al nivel base, donde se ejecutan las nuevas funcionalidades [41].

2.4. Computación Autonómica

2.4.1. ¿Qué es la computación autonómica?

Es un concepto a través del cual un sistema de componentes físicos y lógicos indaga constantemente sobre cómo mejorar su rendimiento y eficiencia [42]. Un sistema autonómico tiene una configuración automatizada para la colección, análisis y transmisión de información, de forma que se ajusta sin intervención externa. Un sistema autonómico posee las siguientes características generales [43]:

Consciente del contexto: una aplicación autonómica debería ser consciente de su espacio de ejecución y estar disponible a reaccionar a los cambios.

Abierto: una aplicación autonómica debe funcionar en un mundo heterogéneo y debería ser portable en múltiples plataformas físicas.

Anticipatorio: una aplicación autonómica debería anticipar en toda la extensión posible sus necesidades y comportamientos, con el objeto de estar dispuesta a manejarlos proactivamente.

La computación autonómica es un concepto que trae inmerso la posibilidad de generar sistemas de automanejo. Los conceptos que definen la computación autonómica se anclan en las siguientes subáreas: la autoadaptación, el autocontrol, y la autoobservación. La computación autonómica radica en la capacidad del sistema de observar un contexto, analizarlo, y tomar decisiones sobre el mismo, para mitigar un problema existente. [44]

2.4.2. Propiedades Autonómicas

Las propiedades generales de un sistema autonómico pueden ser resumidas en 4 objetivos: Autoconfiguración, Autosanado, Autooptimización y Autoprotección, y cuatro atributos: autoconsciente, consciente del ambiente, automonitoreo y autoajuste. A continuación, se describen los atributos:

Autoconfiguración: es la habilidad del sistema para reajustarse automáticamente, bien sea cuando el contexto se modifica, o para asistir a los procesos de autosanamiento, autooptimización, o autoprotección.

Autosanamiento: en un modo reactivo, es un mecanismo que se ejecuta en el contexto cuando se detecta la ocurrencia de una falla. De este modo, identifica las causas de la falla, donde ocurre, modos de reparación. Además, en un modo proactivo monitorea el funcionamiento de los sistemas, buscando predecir y evitar los problemas presentes en ellos.

Autooptimización: significa que un sistema está consciente de su comportamiento ideal; esto es, detecta su comportamiento actual, compara con el comportamiento ideal y establece políticas para mejorarlo.

Autoprotección: el sistema se defenderá a si mismo de ataques externos maliciosos o accidentales. Esto significa que el sistema está consciente de las amenazas, y establece modos para amilanarlas.

2.4.3. Lazo autónómico MAPE+K

La estructura MAPE+K es la base para la implementación de sistemas autónomos, y se refiere a las tareas de: Monitorear (**M**onitor), Analizar (**A**nalyze), Planificar (**P**lan), y Ejecutar (**E**xecute), basado en el Conocimiento (**K**nowledge) que el sistema posee (ver **Figura 2.2**). Esta estructura identifica y clasifica los 4 tipos de tareas que el ciclo autónomo debe ejecutar en el contexto.

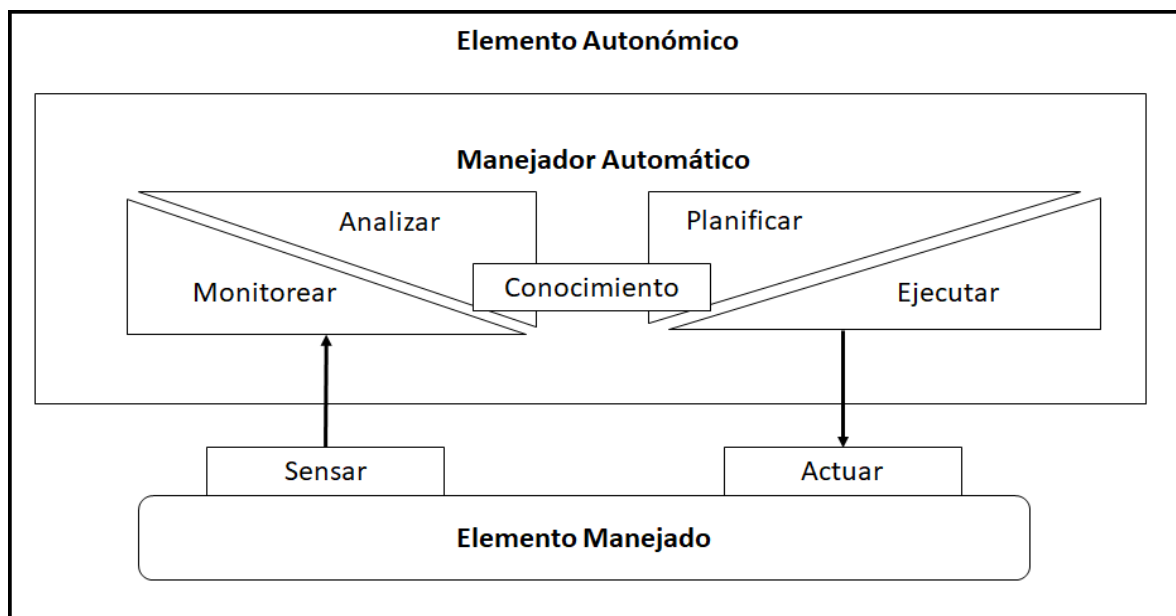


Figura 2.2. Modelo MAPE+K

En estos sistemas autonómicos se contemplan varios componentes relevantes que son:

- *Recurso Gestionado*: puede ser cualquier tipo de recursos (hardware o software) que puede ser gestionado. El recurso gestionado se controla a través de sus sensores y actuadores.
- *Punto de Enlace*: Enlaza a los sensores y/o actuadores con los recursos requeridos, para su gestión.
- *Gestor Autonómico*: Implementa los lazos de control inteligentes que automatizan las tareas de autorregulación/autogestión de las aplicaciones.
- *Orquestador de Gestores Autonómicos*: Debido a que un sistema autonómico puede contar con varios gestores autonómicos, que necesitan trabajar en conjunto para garantizar el funcionamiento correcto de los recursos, este nivel proporciona la coordinación entre ellos.
- *Gestor Manual*: Permite a los humanos configurar los gestores autonómicos para realizar su tarea de autogestión, proveyendo para esto una interfaz hombre-máquina que permite conectar al hombre con el gestor autonómico.
- *Fuentes de Conocimiento*: Proporciona los conocimientos requeridos para la gestión autonómica del sistema.

Capítulo 3. Entornos Computacionales de soporte a la nanotecnología

La definición del middleware y sus componentes es crucial para conocer el potencial de nuestra propuesta. A continuación, exponemos el contenido de 3 publicaciones en revistas internacionales que destaca las características de ARMNANO, y como sus componentes interactúan entre sí para generar sus funcionalidades. A continuación, presentamos el middleware que vamos a trabajar, así como sus propiedades.

3.1. Middleware Reflexivo Autonomico para el Manejo de Nanodispositivos en un Ambiente Inteligente

ARMNANO es un middleware capaz de medir, analizar, actuar y aprender al nivel de los nanodispositivos en un AmI [45]. En este sentido, ARMNANO es ubicuo, no invasivo y responde en tiempo real. Es un middleware reflexivo que incorpora la nanotecnología como actor clave en el nivel base, con un conjunto de mecanismos para su gestión en el contexto de la AmI, los cuales son ofrecidos por ARMNANO como servicios (Ver Figura 3.1.). Para el despliegue de estos servicios, nuestro middleware explota el paradigma de la computación en la nube.

Nuestro middleware cuenta con componentes específicos, para permitir la comunicación entre nanodispositivos, y adicionalmente, ciclos autónomos de procesamiento de datos que permiten la captura de los datos por parte de los nanosensores, y la ejecución de comandos por parte de los nanoactuadores. Esta plataforma se caracteriza por los siguientes conceptos:

- **Autonomía:** es propiedad del sistema para monitorear, comprender y actuar. En nuestro caso, los ciclos autónomos de las tareas de análisis de datos cumplen esta propiedad.
- **Reflexiva:** Es la capacidad de la arquitectura para pensar y adaptarse al contexto de operación de la AmI. En nuestro caso, los ciclos autónomos determinan las acciones a ejecutar por los nanoactuadores, luego de analizar la información del contexto, para adaptarla.
- **Autoaprendizaje:** es la capacidad del sistema para aprender nuevos conocimientos. En nuestro middleware, cada ciclo autónomo procesa datos para generar conocimiento, lo que significa que el sistema es capaz de aprender.
- **Tiempo real:** a pesar de la capacidad reflexiva, el sistema reacciona de inmediato, para descifrar las situaciones y lograr la resolución de problemas en tiempo real.

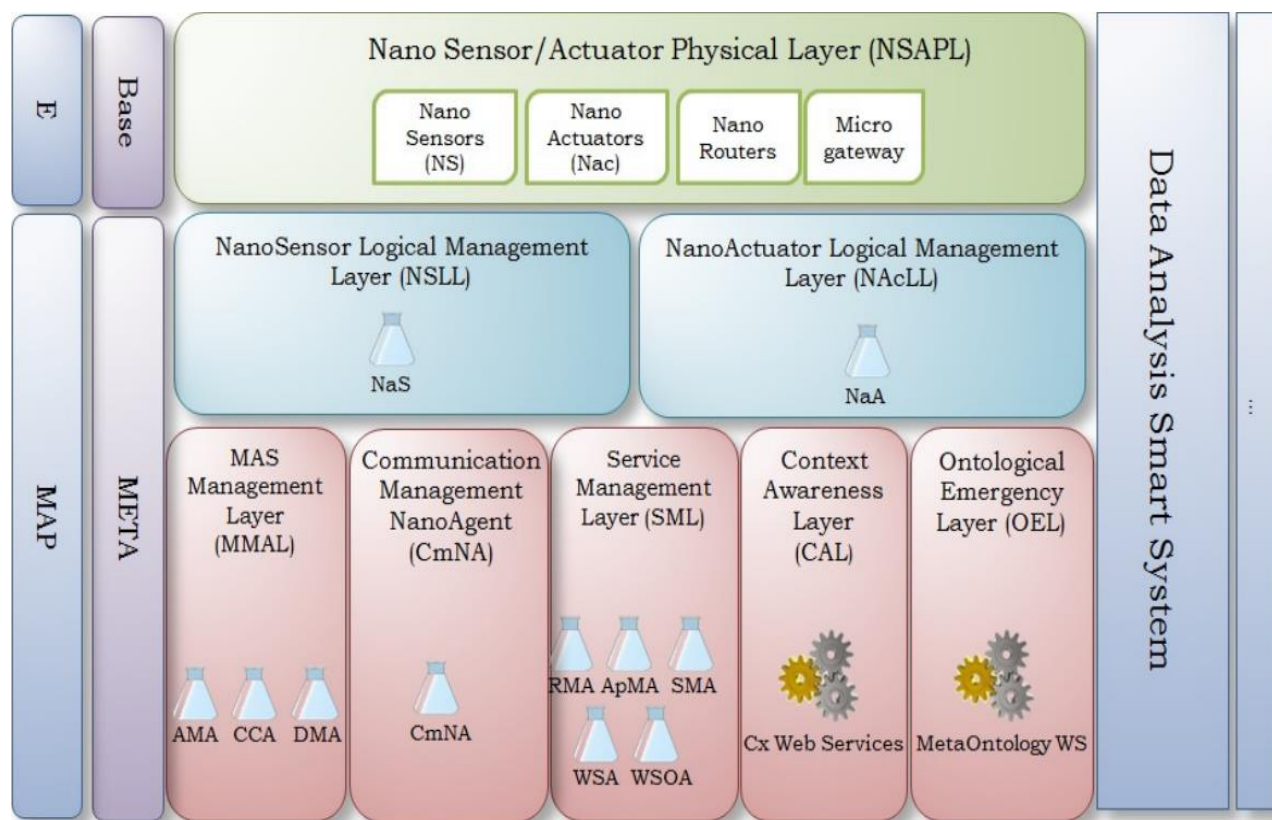


Figura 3.1. Arquitectura ARMNANO

Como se muestra en la **Figura 3.1**, ARMNANO es un Middleware Reflexivo dividido en dos niveles principales, el *nivel base* (donde se lleva a cabo la fase de Ejecución del lazo MAPE) y el *nivel meta* (donde se realiza el Monitoreo, Análisis y Planificación del lazo MAPE).

Nivel Base: En este nivel se encuentran los nanodispositivos desplegados en el AmI. El middleware garantiza una respuesta en tiempo real, a través de las tareas individuales ejecutadas por los nanodispositivos. Está compuesto por Nanosensores y Nanoactuadores que se enfocan principalmente en monitorear el AmI y actuar de acuerdo a las decisiones tomadas en el nivel meta. Estas nanoestructuras miden las variables de interés, así como también realizan las tareas que se les ordenan.

El nivel base se denomina NanoSensor/Actuator Physical Layer (NSAPL), y sus componentes principales son: (i) nanosensores (NSs): estos nanodispositivos observan las variables de interés in situ (por ejemplo, en el cuerpo humano), de manera ubicua [46]; (ii) nanoactuadores (NAs): son nanodispositivos que reciben órdenes de actuar en el AmI. Los nanoactuadores son agentes a nivel de base, que actúan de acuerdo a los

comandos que son enviados, normalmente por el DASS. Los NAs representan el primer contacto para la ejecución de tareas que utilizará el sistema; (iii) nanorutadores: representan dispositivos que enrutan la información entre nanodispositivos en el AmI. Están ubicados entre los agentes Nanosensor y Nanoactuador y el microgateway (Mc), para controlar el enrutamiento de datos. Este nanodispositivo une la transmisión de datos a la unidad de micropuerta de enlace con menos tráfico; (iv) micropuerta de enlace: se dedica a enrutar y autenticar la información de manera adecuada. Este agente enruta la información desde los nodos nanosensores al DASS, así como los datos desde el DASS a los agentes nanoactuadores.

Nivel Meta: comprende el segundo nivel del middleware reflexivo. En este nivel se realiza la introspección, analizando el intercambio de mensajes en la capa superior, particularmente entre los nanosensores/actuadores. Nuestro middleware tiene 7 capas en este nivel. Algunas de estas capas han sido definidas en trabajos previos para la gestión de AmIs [7] [47], pero otras son específicas debido a la inclusión de la Nanotecnología en un AmI. Los componentes definidos en trabajos anteriores son:

- **Capa de gestión multiagente (MMAS):** esta capa está definida por el estándar FIPA (ver [7] para más detalles). Está compuesto por agentes como Agentes de gestión de agentes (AMA), Agente de control de comunicaciones (CCA) y Agente de gestión de datos (DMA). Estos agentes se definen en [7].
- **Capa de Gestión de Servicios (SML):** esta capa es trascendental para la arquitectura ARMNANO, ya que permite la conexión entre los paradigmas MAS y SOA [13]. Significa que los agentes pueden consumir servicios web en la nube y que las tareas del agente pueden concebirse como servicios web. Esto permite utilizar el modelo SaaS de computación en la nube en ARMNANO. Los agentes de esta capa son: el Agente de Gestión de Servicios (SMA), el Agente de Servicios Web (WSA), el Agente Orientado a Servicios Web (WSOA), el Agente de Gestión de Recursos (RMA) y el Agente de Gestión de Aplicaciones (ApMA). Estos agentes se definen en detalle en [47] [48].
- **Capa de conocimiento del contexto (CAL):** esta capa ofrece servicios de contexto a través de un ciclo que se compone de descubrimiento, modelado, razonamiento y distribución del contexto. Esta capa se basa en los servicios ofrecidos en [18], donde se propone un middleware contextual basado en la computación en la nube.
- **Capa de Emergencia Ontológica (OEL):** esta capa ofrece servicios ontológicos basados en [49]. Algunos de los servicios son: actualización automática de ontologías para adaptarlas al entorno, y la Emergencia de Ontologías.

Nuestro middleware extiende esas capas para poder gestionar nanodispositivos. Las capas específicas que ejecutarán tareas relevantes en la transmisión y procesamiento de nanodatos de la arquitectura, se describen a continuación,

1. **Capa Lógica de Nanosensores** (NSLL en inglés, Nanosensor Logical Layer): esta capa caracteriza a cada nanosensor como un agente. Esta capa guarda metadatos que describen el comportamiento del agente. Posee el agente NaS que caracteriza a cada nanosensor en su conjunto y las aplicaciones que se ejecutan en él [14]. Estos agentes son agentes conscientes del contexto, y se coordinan y cooperan entre sí, para medir las variables de interés de una manera no invasiva. El NaS recibe datos de los sensores individualmente y luego los procesa.
2. **Capa Lógica de Nanoactuadores** (NAcLL, en inglés, NanoActuator Logical Layer): esta capa es donde se describen los nanoactuadores. Esta capa será operada por los agentes de la NaA. Ontológicamente, los nanoactuadores ejecutan órdenes, por lo tanto, no poseen algoritmos elaborados, sino para reconocer las órdenes y desencadenar acciones inmediatas. Los comandos se analizan y envían desde el DASS a los nanoactuadores. Los protocolos se implementan aquí para controlar la acción de los agentes nanoactuadores in situ.
3. **NanoAgente de Manejo de la Comunicación** (CmNA en inglés, Communication Management Nanoagent): esta capa se encarga de conectar los nanodispositivos. Los mecanismos de comunicación pueden ser Comunicación Molecular, Comunicación Electromagnética, entre otros. La comunicación y transmisión de datos son manejadas por los agentes en ella. Esta capa define los protocolos de comunicación para recibir y entregar datos, desde los nanosensores a DAAS, y desde DAAS a los nanoaccionadores [49]. Garantiza que el sistema se ejecute automáticamente para detectar y actuar.

Por otro lado, existe una capa transversal a las anteriores, que se describe a continuación:

Sistema Inteligente de Análisis de Datos (DASS en inglés, Data Analysis Smart System): es un sistema transversal a los niveles base y meta, que se encarga del procesamiento de los datos. En particular, en él se definen los ciclos autónomos de análisis de datos [50], [51]. Es un ciclo de supervisión de procesos basado en tareas de análisis de datos, que aprenden en base a la experiencia. Estos ciclos autónomos pueden ejecutar tareas de minería de datos, y pueden utilizar diferentes tipos de representaciones de conocimiento.

Particularmente, la plataforma ARMNANO es un sistema distribuido autónomo que proporciona servicios de análisis de datos soportados en nanodispositivos. La mayor parte de este análisis de datos se lleva a cabo por el DASS. En particular, DASS gestiona un conjunto de Ciclos Autónomos (CAs) de tareas de análisis de nanodatos para la toma de decisiones [3]. DASS ejecutará el conjunto de CAs como servicios que dependen del contexto. En particular, en DASS se definen 3 CAs genéricos, que son, el Ciclo de

Acondicionamiento (CTM en inglés, *Conditioning Thinking Mode*), el Ciclo de Optimización del Funcionamiento (OTM en inglés, *Optimizing Thinking Mode*), y el Ciclo de Fijación de Comportamiento del Sistema (FTM en inglés, *Fixing Thinking Mode*). Los Ciclos Autónomos se describen a continuación,

1. Ciclo de Acondicionamiento (CTM en inglés, *Conditioning Thinking Mode*): Esta CA tiene como objetivo definir el conjunto de actividades para establecer los valores de las variables en sus setpoints, en base al contexto definido por los nanodatos, como se muestra en la **Figura 3.2** Este CA se activa una vez instalada la arquitectura en un contexto dado.

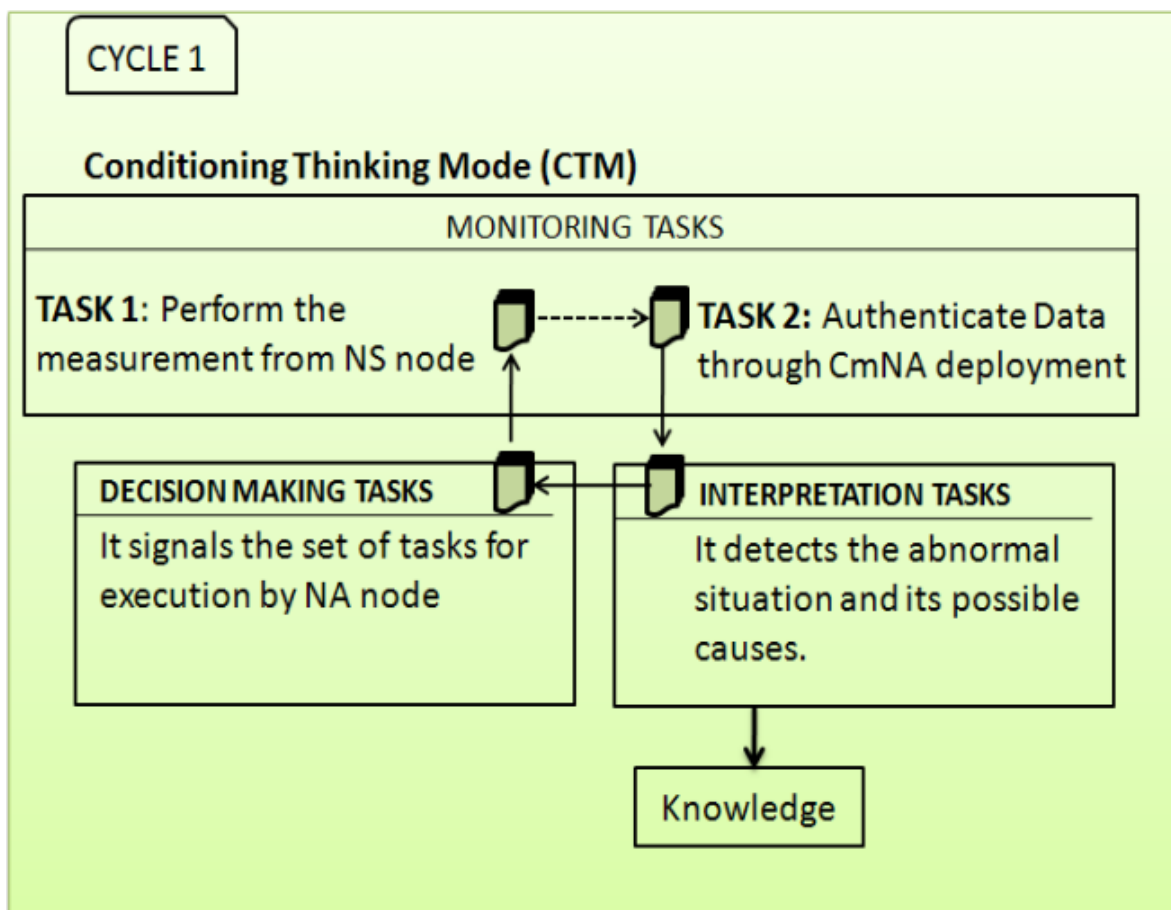


Figura 3.2. Ciclo Autónomo en modo CTM (ver ref. [3])

- i) **Monitoreo:** mide las variables utilizando el nodo de NSs.
- ii) **Autenticación de Datos:** en esta tarea se detecta y verifica la situación actual. Interactúa con los agentes NaS y el mecanismo de autenticación de CmNA.
- iii) **Tareas de Interpretación:** determina la situación actual en cuanto al nuevo

contexto o la situación anómala, y sus posibles causas.

- iv) **Tareas de Toma de Decisiones (Autoacondicionamiento):** determina un conjunto de tareas que deben ejecutar los nodos NA, con el fin de restablecer los valores de las variables en el AmI que no están en sus setpoints.

2. Ciclo Autónomo de Optimización (OTM en inglés, *Optimization Thinking Mode*): Este CA se utiliza cuando hay una degradación del rendimiento en el AmI o cuando el middleware detecta una nueva restricción en el AmI. Las causas típicas para ejecutar esta CA son:

- a) El tiempo de respuesta es muy grande,
- b) Hay mucho consumo de recursos,
- c) No se alcanzan las metas generales en el AmI,
- d) Hay nuevas restricciones sobre los nanodispositivos, el contexto, entre otros.

En la **Figura 3.3** observamos las tareas que se ejecutan en este ciclo autónomo de optimización.

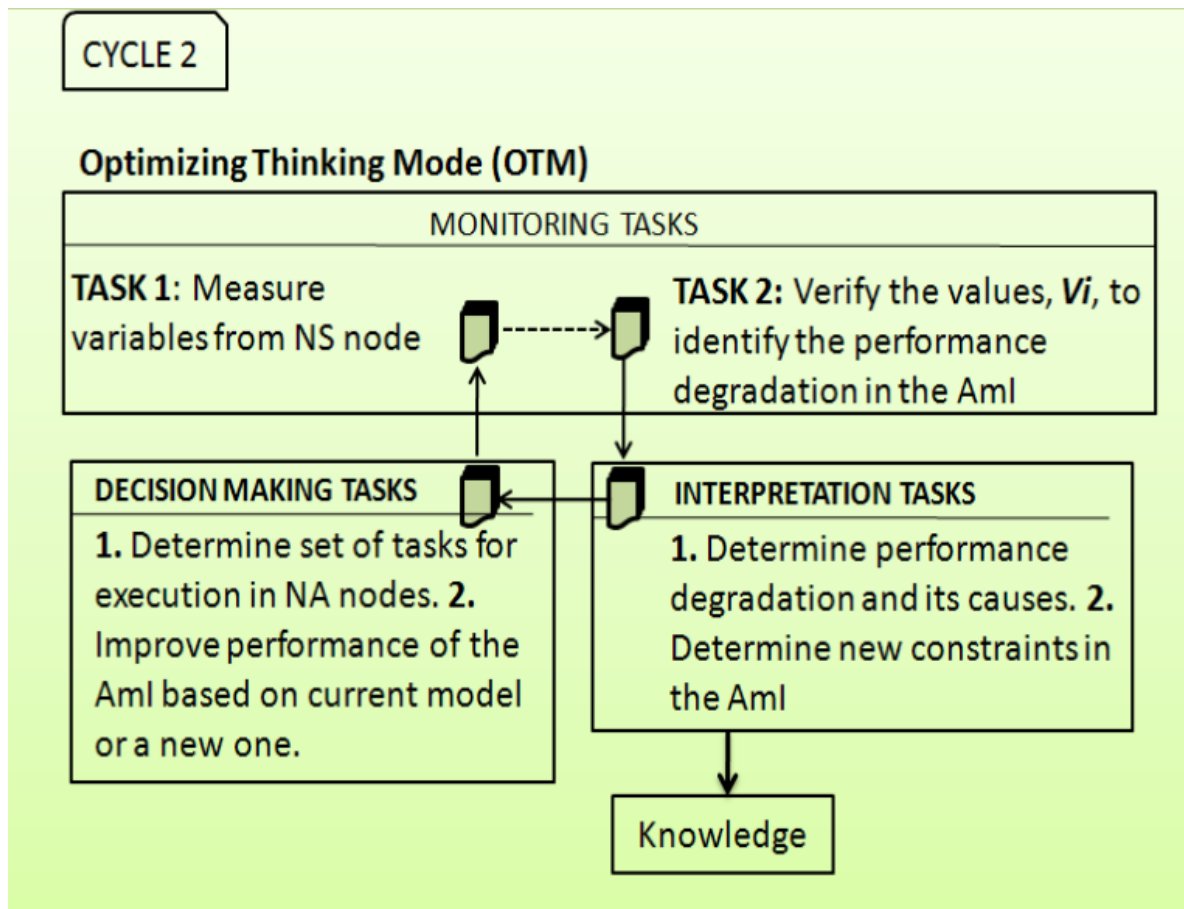


Figura 3.3. Ciclo Autónomo en modo OTM (ver ref. [3])

3. Ciclo de Fijación de Parámetros en Tiempo Real (FTM en inglés, Fixing Thinking Mode): Este CA actúa cuando hay un cambio repentino en el Aml y se requiere actuar rápidamente para restablecer las condiciones de desempeño de los nanodispositivos en el sistema como se evidencia en la **Figura 3.4** De esa forma, primero es necesario determinar los objetivos a seguir y las variables vinculadas a ellos, y luego, condicionar los diferentes nanodispositivos para seguirlos. Es decir, en este caso se deben determinar los puntos de referencia que deben seguir las variables, para restablecer las condiciones ideales de la condición supervisada.

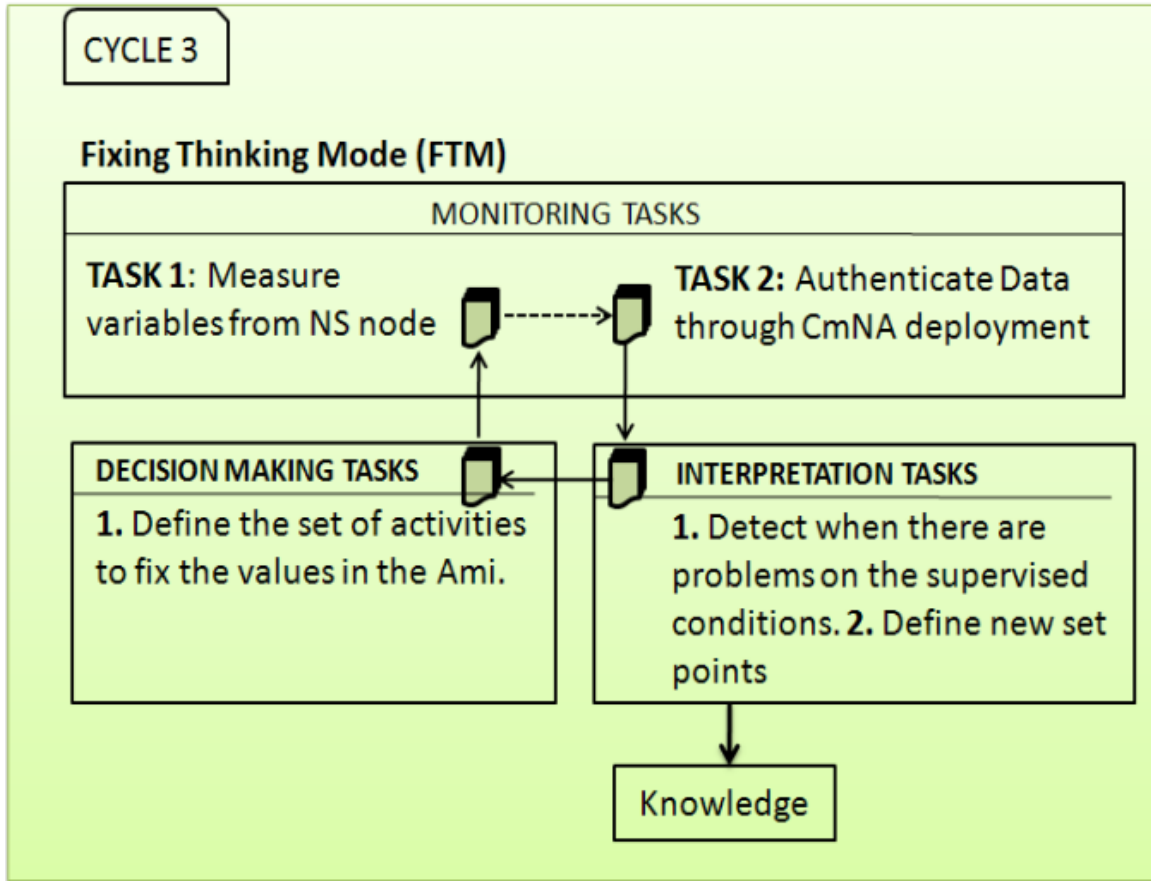


Figura 3.4. Ciclo Autónomo en modo FTM (ver ref. [3])

- i) **Tareas de Monitoreo:** Captura de variables del nodo de NSs.
- ii) **Autenticación de Datos:** en esta tarea se verifica la situación actual. Interactúa con los agentes NaS y el mecanismo de autenticación de CmNA.
- iii) **Tareas de Interpretación:** Esta tarea define las variables que requieren nuevos puntos de ajuste y sus valores. Para ello: 1. Determina la degradación de las condiciones supervisadas en el AmI. Detecta cuando hay problemas en las condiciones supervisadas. 2. Determina nuevo *setpoints*.
- iv) **Tareas de Toma de Decisiones:** Autoconfiguración, esta tarea especifica el conjunto de actividades para fijar los valores en el AmI, normalmente utilizando los nodos NA para esta tarea.

3.2. Sistema de Manejo de Nanocomunicación para Ambientes Inteligentes

Este subcapítulo define la capa CmNA. CmNA cubre todos los aspectos relacionados con la transmisión de datos, que involucran a los nanodispositivos como el NS (nanosensor), NA (nanoactuador) y Ro (nanorrutador), acoplados entre ellos, para desplegar las capacidades de la capa NSAPL en ARMNANO [1]. Además, microgateway es la unidad de procesamiento ubicada en el nanorouter, que asegura el significado lógico de los datos medidos en los nanosensores. Por lo tanto, CmNA es un elemento crucial y una capa nueva en el middleware que involucra el componente nano. La **Figura 3.5** ilustra los diferentes elementos de la capa CmNA.

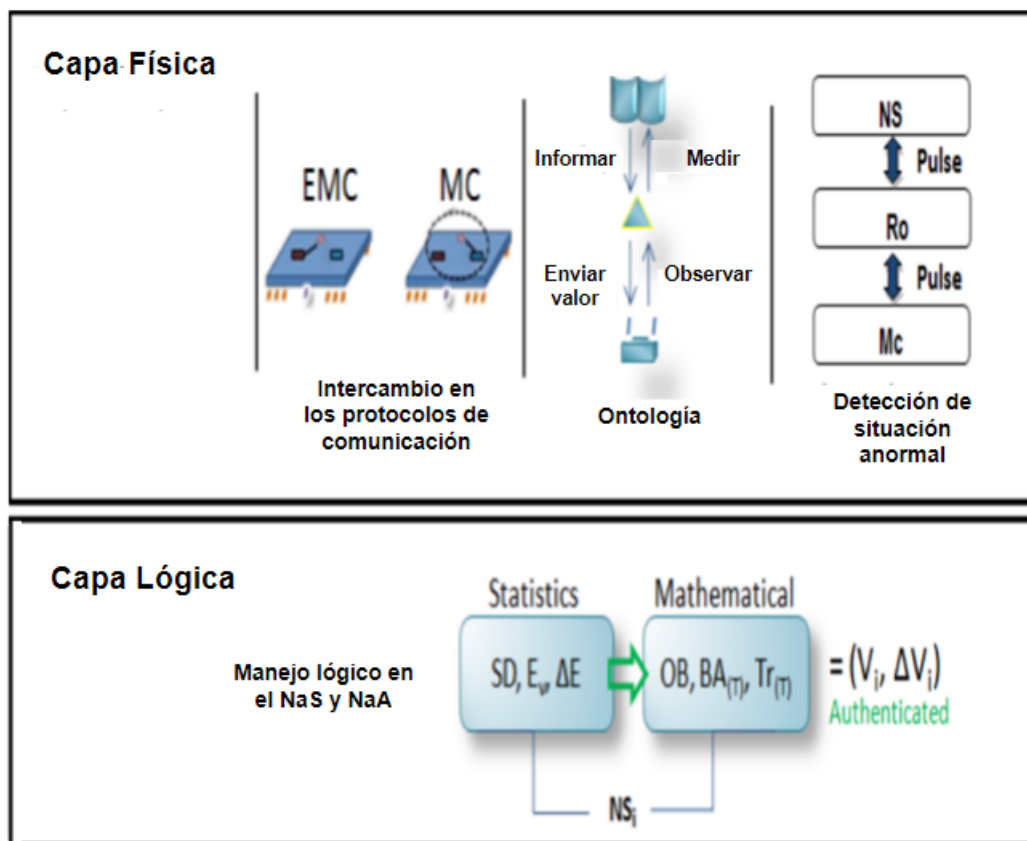


Figura 3.5. Componentes de la Capa CmNA (ver ref. [1])

La capa CmNA soportará los siguientes protocolos de comunicación:

- A. *Comunicación Electromagnética* (EMC en inglés, Electromagnetic Communication): se basa en la transmisión y recepción de radiación electromagnética de los nanodispositivos. En general, este tipo de comunicación

ocurre basado en la transmisión de una onda, la cual **tiene un conversor** en la banda de Terahertz.

- B. *Comunicación Molecular* (MC en inglés, Molecular Communication): es un paradigma comunicacional que utiliza señalización bioquímica para lograr intercambio de información entre nanosistemas sintetizados natural o artificialmente. En general, MC lleva el intercambio de información por medio de las moléculas.

Por otro lado, la capa física tiene los siguientes componentes,

- ***Mecanismo de Selección de Protocolo*** que se basa en un procedimiento de conmutación que detecta el contexto in-situ del lugar en el que se realiza la medición, para seleccionar el Protocolo de Comunicación: EMC o MC. Selecciona el protocolo usando la desviación estándar de las medidas obtenidas como sigue:
 - *Contexto Acumulado (Sistema Enfocado)*: para este caso, si la desviación estándar está por debajo del umbral de 0,1, entonces selecciona el protocolo EMC.
 - *Contexto Relajado (Sistema Disperso)*: para este caso, si la desviación estándar está por encima del umbral de 0,1; entonces selecciona el protocolo MC.
- ***Protocolo de Intercambio de comunicación***: comienza después de seleccionar el protocolo de comunicación, ya sea EMC o MC. Puede cambiar el protocolo en tiempo de ejecución utilizando una regla similar a la utilizada por el mecanismo de selección. En el caso del protocolo EMC, los nanosensores miden, pero en el caso del protocolo MC, los nanosensores son puertas de enlace. Es decir, un nodo NS típico poseerá una función híbrida.
- ***Ontología del Nivel de Gestión de Comunicación Física***: La capa CmNA requiere de una ontología, para permitir el intercambio de mensajes para tareas de comunicación entre los Ro's, NS's y NA's. Esta ontología describe los diferentes componentes de la capa CmNA (como los sensores, enrutadores, etc.), los conceptos de "actividad" y "actividad de dominio" que describen el proceso de nanocomunicación, y la "variable" y conceptos de "usuarios" que describen el nanoentorno (nanodatos, etc.).
- ***Mecanismo de Detección de Situaciones Anormales***: Debido a que el middleware es autónomo, requiere evaluar la conexión de los diferentes componentes en la capa NSAPL. Este mecanismo se inicia probando el estado operacional de los componentes del CmNA donde cada componente está equipado para enviar un pulso de una frecuencia que pueden detectar NS, NA, Ro o Mc. El middleware comienza a evaluar desde un NS o NA, los cuales envían un pulso a Ro, y luego Ro a Mc. Si estos nanodispositivos reconocen la señal del pulso, el middleware lo entiende como que NS, Ro y Mc, están conectados correctamente, cumpliéndose el estado A de la **Figura**

3.6. El estado B sucedería si Ro no recibe el pulso de NS o NA, interpretándose que no está conectado y debe migrar a otro Ro. Finalmente, el estado C es una situación de falla persistente donde la señal no se detecta en Ro y Mc, determinando el middleware autónomamente que Ro y NS, o Ro y NA, deben cambiarse.

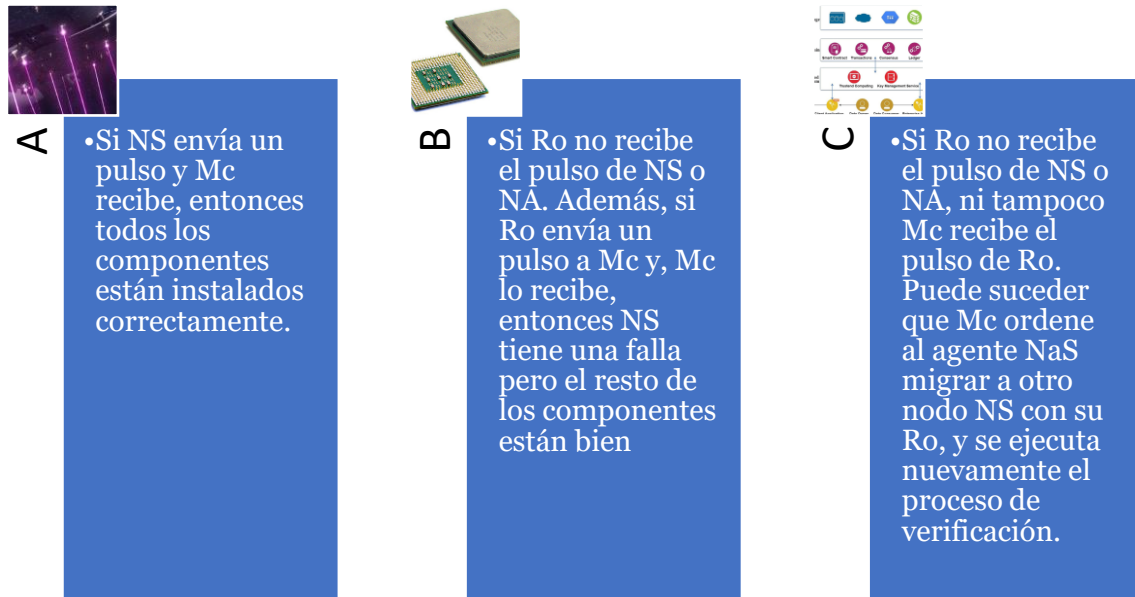


Figura 3.6. Esquema de detección de fallas en la capa CmNA reconociendo los estados A, B y C en que puede estar el middleware

La capa CmNA debe facilitar dos aspectos, una transmisión de datos descentralizada y una autenticación de datos descentralizada, de forma que las fallas se corrijan puntualmente y no afecte el sistema globalmente. Así, la capa lógica implementa los mecanismos de autenticación de datos, los cuales son:

1. **Autenticación de Datos en el Dominio Estadístico:** Como ARMNANO es un sistema autónomo, validará los datos para asegurar que corresponden a una descripción del objeto observado. El sistema reconoce los valores como estadísticamente precisos. El tratamiento estadístico incluye la Desviación Estándar (SD) y el Error Relativo (Er). En particular, estas funciones son necesarias porque un nodo NS medirá iterativamente V_i , hasta la autenticación de la misma.

Desviación Estándar, **SD** < 0.1

Porcentaje de error, **Er** < 5%

Una SD por debajo de 0,1 significa que el nanosensor está observando el mismo valor (estadísticamente, se asegura que la precisión es alta). Además, si Er está por debajo del 5%, entonces V_i está cerca de sus valores medidos típicos en el objetivo observado, lo que indica una alta exactitud. En este contexto, el valor V_i se autentica porque la precisión y la exactitud son altas.

2. **Autenticación de datos en el dominio matemático:** En la capa CmNA se realiza también la autenticación basada en argumentos matemáticos para complementar la evaluación estadística de la sección 4. Aquí nosotros proponemos 2 herramientas matemáticas que se puede aplicar a la nanodata.

- *Comportamiento fuera de Rango (OB en inglés, Outranged Behavior):* la magnitud de cada variable tiene una oscilación normal descrita por las medidas realizadas durante un tiempo largo. Así, un comportamiento fuera de rango se entiende como una falla, y se describe como sigue.

$$OB_{(x)} = \begin{matrix} X_f \\ : \\ X_0 \end{matrix}$$

Esta ecuación establece el comportamiento normal de $V_{(x)}$ dentro de los límites X_0 y X_f . Así $OB_{(x)}$ indica autenticación cuando el valor observado V_i oscila dentro del rango X_0 y X_f y no fuera.

- *Tendencia (Tr en inglés, Trending):* la data de salida enviada para autenticación puede contener patrones, lo que indica que a nivel lógico existe una tendencia. La tendencia (Tr) puede ser definida matemáticamente como,

$$\lim_{x \rightarrow t} Tr_i(x) = V_i(t)$$

Este límite representa la tendencia de $V_i(t)$ en el tiempo. Por ejemplo, una persona alcanza la máxima estatura a los 21 años de edad, entonces V_i representa la altura de las personas donde $x \rightarrow 21$. Se cumple que Tr_i indica la autenticación cuando el valor observado V_i está por debajo del valor máximo $V_i(t)$, indicando que no es un valor atípico.

3.2.1 Sistema de Autenticación de NanoDatos en ARMNANO

En lo subsiguiente se describe los componentes del método de autenticación de los datos en ARMNANO, que dicho middleware ejecuta autónomamente. En particular, es necesario distinguir la manera como se capturan el conjunto de datos (series de tiempo), en ARMNANO, los cuales se realizan en dos modos, vista local o global (*LV* o *GV* en inglés, que significa Local View y Global View, respectivamente). Estos modos se definen como:

- ***Series de Tiempo en “Local View”***: se trata de la caracterización de un conjunto de datos monitoreado en un tiempo único t (cantidad de datos, V_i , se miden con una cantidad de NSs que miden simultáneamente la misma variable), lo que permite autenticar cada conjunto de datos en un tiempo t , individualmente. En este modo de observación la cantidad de NSs es mayor que 1.
- ***Series de Tiempo en “Global View” (GV)***: Revela que los datos son analizados globalmente en todo el rango de tiempo. Por tanto, la autenticación de datos es desde una perspectiva global, juntando el conglomerado de información que se ha registrado en cada tiempo, t_i , y no en un solo punto de datos como en el modo *LV*. Este sistema es usual para una arquitectura de un sensor único ($NS=1$). Utilizando cada dato medido en t_i en la serie de tiempo, se construye una línea temporal como $\{t_0, t_1, t_2, \dots, t_n\}$, dando cobertura a un período de tiempo p .

Según sea el modo de observación, *LV* o *GV*, para *LV* el conjunto de datos en cada punto t_i es $(V_1, V_2, V_3, \dots, V_n)$, los cuales son provistos por n nodos NSs que miden la misma variable, por lo que la autenticación se lleva a cabo en cada tiempo t_i exclusivamente. Por otro lado, en el modo global, *GV*, el *dataset* se construye sobre la base de un único NS. Así, la autenticación se hace en la data compilada durante el período de tiempo p . El conjunto de datos en p con n cantidad de observaciones es $\{(t_0, V_0); (t_1, V_1), (t_2, V_2), \dots, (t_n, V_n)\}$.

De este modo se distinguen dos situaciones claras en un sistema las cuales contempla, un único sensor para el modo *GV*, y n cantidades de NS en el modo *LV*.

3.2.1.1 Descripción de los Módulos de Autenticación

Dado que el middleware ARMNANO se ejecuta autónomamente, necesita asegurar que los datos que analiza son identificativos del contexto donde actúa. Considérese que los NS y NA están actuando libremente en el middleware y el sistema debe tomar decisiones sobre data verdadera. Así, nuestro middleware despliega en el agente NaS un software que nosotros hemos denominado *Protocolo de Autenticación de Nanodatos*.

La arquitectura para autenticar estos nanodatos en ARMNANO se basa en 2 módulos. El primer módulo consiste en la aplicación de principios estadísticos sobre el *dataset*,

mientras que el segundo módulo usa fundamentos probabilísticos basados en la técnica de Estimación de Máxima Similitud (MLE en inglés, *Maximum Likelihood Estimation*) para modelar el *dataset* [4]. Para ello este módulo usa dos etapas, la etapa 1, denominada de entrenamiento, y la etapa 2, denominada de funcionamiento.

En la **Figura 3.7.** se muestra cómo se integran los módulos para procesar cada nanodato x_i que se mide en tiempo real usando los nodos NSs desplegados en el contexto. En la **Figura 3.7.**, el contexto determina el objeto de estudio. En el contexto, NS mide y envía la información a la unidad Mc del middleware (**Figura 3.1.**) que invoca a la capa CmNA. Luego, CmNA despliega el agente NaS para que se ejecute el módulo 1 y conocer si estadísticamente x_i se admite. La salida del módulo 1 pasa al módulo 2 para saber si la distribución de MLE reconoce a x_i . Si este valor x_i pasa ambos módulos, entonces está Autenticado. Una vez el nanodato es autenticado se almacena y puede ser invocado por el DASS.

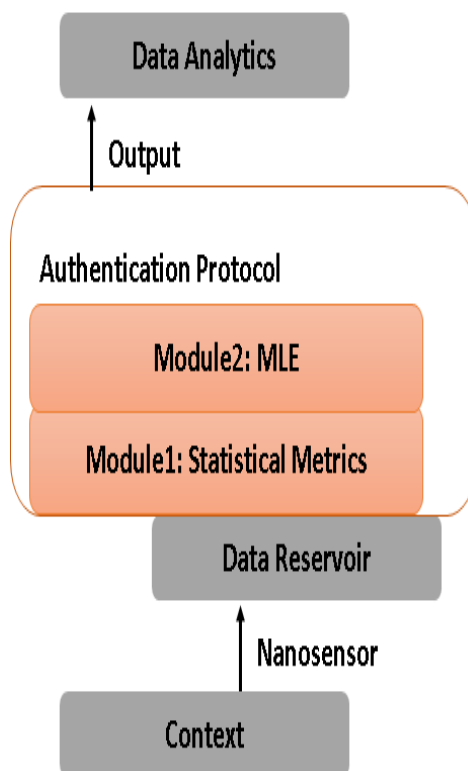


Figura 3.7. Arquitectura del sistema de autenticación de datos

A continuación, describimos las características de los módulos desplegados.

Módulo 1: Análisis Estadístico del dataset

Etapas de Entrenamiento:

Durante el modo de entrenamiento, el objetivo del primer módulo es determinar las métricas estadísticas del conjunto de datos, previendo así la dispersión de datos en modo LV o GV. El nano conjunto de datos se utilizará para establecer los valores estadísticos estándar de las variables en el contexto (Ver el macroalgoritmo de la **Tabla 1**).

Tabla 1. Macroalgoritmo para el módulo 1 en la etapa de Entrenamiento. (**Ver ref.** [4])

<u>Input:</u> crude data
<u>Procedure:</u> 1. Distinguish type of data series For <i>LV</i> : $x_i = (V_1, V_2, V_3, \dots, V_n)$ ($t = t_i, NS = n$) (it is similar for each <i>LV</i> managed with ARMNANO) For <i>GV</i> : $x_i \in GV$ from $GV = \{(V_1, t_1); (V_2, t_2), (V_3, t_3), \dots, (V_n, t_n)\}$ ($p = [1, \dots, n], NS = 1$) 2. For each <i>LV</i> , Calculate $\sigma_i, Q1_i, Q3_i, IQR_i$ in each t_i Calculate global $Q1_g, Q3_g, IQR_g \forall t_j$ where j are all the measures in the dataset for <i>LV</i> . For each x_i Discard x_i if m out of $1.5 * IQR_g$ or σ_i is very big 3. For GV , Calculate $Q1_p, Q3_p, IQR_p$ in p Discard x_i if x_i out of $1.5 * IQR_p$
<u>Output:</u> processed dataset

En el macroalgoritmo en la **Tabla 1**, en el **paso 1** se distingue el tipo de nanodata (*LV* or *GV*). En el **paso 2** se analiza x_i si el modo del dataset es *LV*. En este modo existen n NSs. De esta forma, en cada t_i se genera un set de valores $(V_1, V_2, \dots, V_n)$. Entonces, se calcula la métrica estadística $\sigma_i, Q1_i, Q3_i, IQR_i$ en t_i . Adicionalmente, se calcula la métrica estadística global **LV** usando todas las mediciones de este NS ($Q1_g, Q3_g, IQR_g$). Finalmente, asumimos que x_i es una medida atípica (outlier) si su valor está fuera de IQR_g o su σ_i es muy grande. En el **paso 3** se estandariza el análisis para *GV*, de forma que la observación se hace desde un único NS durante todo el rango de tiempo $p = [1, \dots, n]$ para registrar el conjunto $GV = \{(V_1, t_1); (V_2, t_2), (V_3, t_3), \dots, (V_n, t_n)\}$. Ahora se calcula la métrica estadística $Q1_p, Q3_p, IQR_p$, de forma que $x_i = (V_i, t_i)$ es un dato atípico si su valor está fuera de IQR_p .

Etapas de Funcionamiento:

Durante el modo de Funcionamiento, un nuevo x_i se mide y compara con las métricas calculadas en la etapa previa. En el caso de GV, x_i se compara con IQR_p , y para el LV actual con su IQR_g , con el objeto de determinar si x_i se descarta (Macroalgoritmo en la **Tabla 2**).

Tabla 2. Macroalgoritmo para el Módulo 1 en la Etapa de Funcionamiento (ver ref. [4])

<u>Input:</u> current x_i
<u>Procedure:</u> 1. Define the data series For GV $x_i = (V_i, t_i)$ For a given LV $x_i = \{V_1, V_2, \dots, V_n\}$ 2. For LV , Calculate σ_i, m_i , Discard x_i if m_i out of $1.5 * IQR_g$ or σ_i is very big 3. For GV Discard x_i if x_i out of $1.5 * IQR_p$
<u>Output:</u> x_i statistically authenticated or not.

En el Macroalgoritmo en la **Tabla 2**, en el **paso 1** se define una serie de datos. En el **paso 2** se analiza LV para el conjunto $LV = \{V_1, V_2, V_3, \dots, V_n\}$. Se usa la métrica estadística en LV de IQR_g para determinar si x_i es un dato atípico. En el **paso 3** se estudia GV, y en este caso el x_i se compara con IQR_p .

Módulo 2: Análisis Estocástico del dataset

Etapas de Entrenamiento:

Durante la etapa de entrenamiento, el macroalgoritmo del Módulo 2 determina el modelo de MLE para este dataset, el cual se define como la mejor Función de Distribución Gausiana (Ver Macroalgoritmo en la **Tabla 3**). Para esta finalidad, consideramos un dataset de entrada que es la salida del Módulo 1 durante el modo de entrenamiento.

Tabla 3. Macroalgoritmo para el Módulo 2 en la Etapa de Entrenamiento (**ver ref.** [4])

<u>Input:</u> processed dataset for Module 1
<u>Procedure:</u> Set the likelihood function $L(\theta)$ Maximize the function $Ln(L(\theta))$ Print $\theta = (m, \sigma)$
<u>Output:</u> MLE model for authentication

En la **Tabla 3**, el macroalgoritmo determina el modelo basado en MLE. En general, el método MLE se usa para construir la mejor Función de Distribución Gausiana para cada variable del dataset. Específicamente, en el paso 1 se estandariza la Función de Probabilidad de acuerdo a la Curva de Distribución de Probabilidad. Ese paso define la Función de probabilidad con el valor de (m, σ) que mejor describa cada variable del dataset.

Etapa de Funcionamiento:

Durante el modo de funcionamiento, se aplican otros criterios para aceptar un x_i : si $P(\theta|x) \geq \beta$, donde β es el umbral de aceptación. Si esta condición no se alcanza, entonces x_i es un dato atípico y se descarta. El Macroalgoritmo en la **Tabla 4** describe este comportamiento.

Tabla 4. Macroalgoritmo para el Módulo 2 en la Etapa de Funcionamiento (**Ver ref.** [4])

<u>Input:</u> x_i statistically authenticated
<u>Procedure:</u> If $P(\theta x) \geq \beta$ x_i is accepted
<u>Output:</u> Final authentication

En el macroalgoritmo en la **Tabla 4**, el paso 1 incorpora la condición para que un nuevo x_i sea autenticado. En este caso, un nuevo x_i no es un dato atípico si la probabilidad de pertenecer a un modelo basado en el entrenado MLE es igual o más grande que β , donde β es el umbral de aceptación de la autenticación.

3.3. Un Modelo Ontológico para un Middleware que gestiona Nanodispositivos en un Ambiente Inteligente

La definición ontológica del middleware ARMNANO es basado en un estrategia de modelado denominada *Model-Driven Architecture*, que contempla la caracterización de dos modelos, el modelo independiente de la computación (CIM) y el modelo independiente de la plataforma (PIM). Mientras que CIM describe de manera general los elementos lógicos de ARMNANO definiendo el núcleo del mismo, PIM indica los aspectos de la plataforma computacional que conforman a ARMNANO. A continuación se especifica la definición de cada modelo

3.3.1 *Capa del modelo independiente de computación (CIM en inglés, Computation Independent Model) en ARMNANO*

El modelo CIM define las ontologías de dominio. Este modelo se enfoca en el dominio del sistema, así como también en los requerimientos funcionales, propiedades no-funcionales, reglas de negocio, metas estructuradas, y estrategia de procesamiento de datos que el sistema debe seguir [2].

En la Figura 3.4. se muestran las capacidades desplegadas por ARMNANO, sus requisitos funcionales y no funcionales, y una taxonomía en torno a los conceptos de nanotecnología que se dividen según diferentes aspectos en este campo. En esta figura se especifican las subáreas de ARMNANO-Nanotecnología, quien representa el área netamente funcional de nuestro middleware enraizado en los NS, NA y Ro. Estas subáreas desplegadas son, por ejemplo, material, función, estrategia, modo de comunicación, entre otras. A continuación, se definen las componentes de las subáreas o categorías de acuerdo al modelo CIM propuesto para ARMNANO, que están señaladas en la **Figura 3.8**:

- i) Material: señala la composición y morfología de los nanoobjetos;
- ii) Función: los nanodispositivos pueden ser nanosensores, nanoactuadores y nanorutadores;
- iii) Estrategia: se trata del enfoque de la nanotecnología para generar modificación de superficies, materiales especializados, patrones de sólidos, entre otras estrategias.
- iv) Modo de comunicación: aquí se especifican las opciones para la transmisión de datos a nanoescala (Ver [52] para más detalles);
- v) Contexto: ARMNANO se puede utilizar en diferentes campos, para una variedad de contextos, como salud, militar o transporte;
- vi) Morfología: los nanomateriales tienen una variedad de formas con diferentes objetivos y propiedades. En nuestro modelo ontológico, la relación principal entre los conceptos es "es-un".

En general, el conocimiento de nanotecnología a nivel CIM es clave para ARMNANO, ya que define la conciencia del contexto.

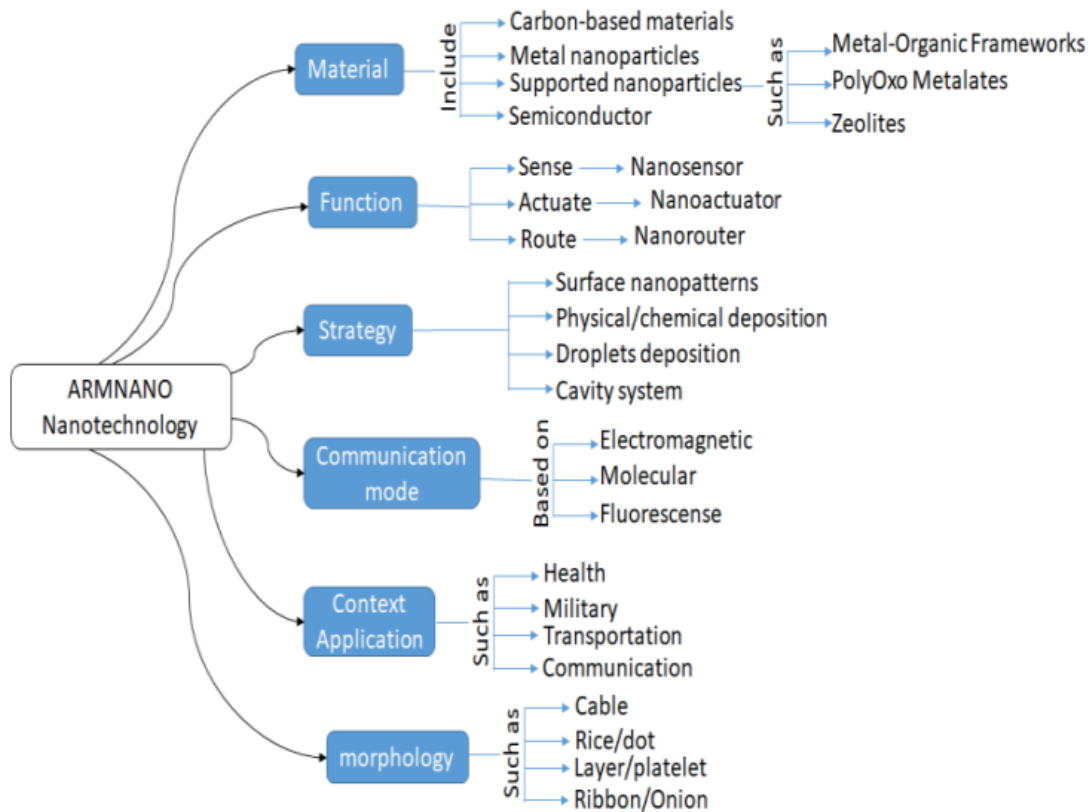


Figura 3.8. Capa CIM de la Ontología ARMNANO (ver ref. [2])

3.3.2 Capa del modelo independiente de la plataforma (PIM en inglés, *Platform Independent Model*) en ARMNANO

El modelo PIM incluye los detalles para la implementación en condiciones reales del sistema. Este modelo define los paradigmas del middleware relativos a computación autónoma, aplicaciones de la arquitectura orientada a servicios (SOA en inglés, *Services-Oriented Applications*), entre otras. En la capa PIM referida a ARMNANO, se definen estos 3 paradigmas. La relación entre estas ontologías queda establecida como indica la **Figura 3.9**:

- SOA “llama” al análisis de datos para dar servicios para el procesamiento de nanodatos,

- El análisis de datos “alimenta” la computación autónoma porque los ciclos lo utilizan,
- Un dispositivo de computación autónoma “entrega” sus tareas usando las capacidades SOA.

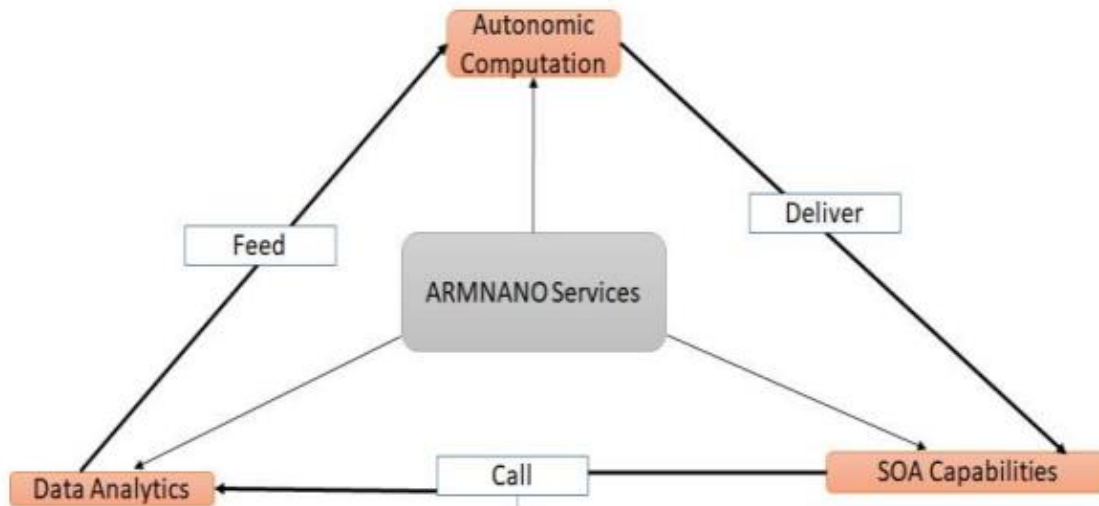


Figura 3.9. Capa PIM de la ontología ARMNANO (ver ref. [2])

Para entender cómo se establece la relación de conceptos y categorías en la capa PIM, se especifican la Ontología de Analítica de Datos y la Ontología de la Computación Autónoma como sigue,

3.3.2.1 Ontología de la Analítica de Datos: esta ontología indica las componentes en cuanto a conceptos, relaciones e instancias que se invocan en ARMNANO, para realizar las tareas de analítica de datos.

- **Conceptos:** Estos conceptos evocan las variables que se definen en las tareas de analítica de datos realizadas en ARMNANO. Tal como se describe en la Figura 3.6, los conceptos de la analítica de datos son;
 - Análisis de datos:* se relaciona con el despliegue de enfoques matemáticos o estadísticos específicos para diferentes tipos de procesamiento de nanodatos.
 - Técnicas:* se refiere a métodos teóricos específicos desplegados para analizar los nanodatos de interés.
 - Tareas:* define el género de análisis de datos a diseñar.
 - Objetivos:* señala la meta de una tarea específica de análisis de datos.
 - Tipos:* en el contexto de ARMNANO.

- **Relaciones:** describe el tipo de conexión que existe entre los conceptos, estas son;
 - i) El análisis de datos tiene una técnica
 - ii) El análisis de datos es una tarea
 - iii) El análisis de datos tiene un objetivo
 - iv) El análisis de datos es un tipo
- **Instancias:** define las posibles instancias de los conceptos.
 - i) Técnicas: redes neuronales, regresión lineal, etc.;
 - ii) Tareas: incluyen la clasificación, predicción, agrupamiento, entre otras;
 - iii) Tipos: se refieren a los diferentes tipos de servicios entregados en la ontología de análisis de datos para un dispositivo de corriente alterna.

Como es de notarse, el arreglo ontológico une los conceptos a las relaciones e instancias, queriendo indicar que estos 2 últimos se aplican sobre los conceptos.

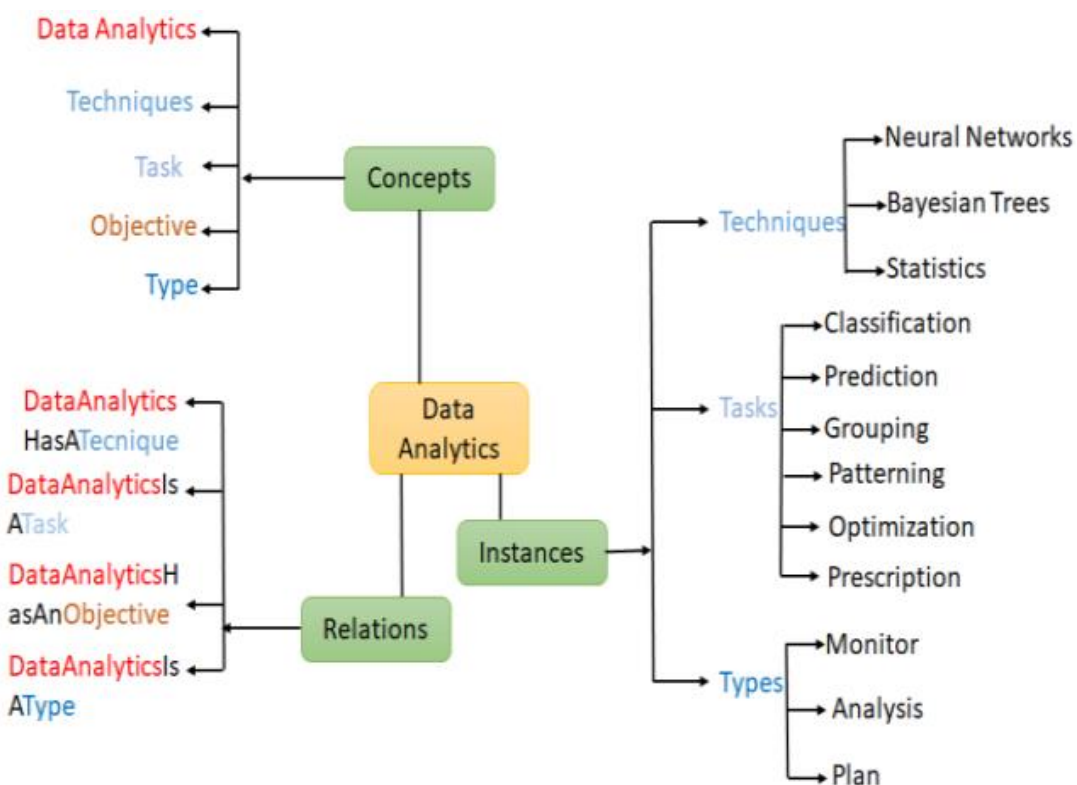


Figura 3.10. Ontología de la Analítica de datos para la Ontología ARMNANO (ver ref. [2])

Como puede verse en la **Figura 3.10**, las relaciones se establecen sobre los conceptos, utilizando, por ejemplo, conectores como “Has A” o “Is A”, dando a entender que ellos pueden combinarse dando origen a estructuras más complejas. Por otro lado, las

instancias contemplan la definición de las Técnicas, Tareas y Tipo de tareas que ejecutan durante la analítica de datos.

3.3.2.2 Ontología De La Computación Autónoma: esta ontología establece los conceptos y relaciones para controlar los aspectos relativos a la implementación de la autonomía en ARMNANO, tal como se indica en la **Figura 3.11**. A continuación, se describen los conceptos u objetos que se invocan en esta ontología:

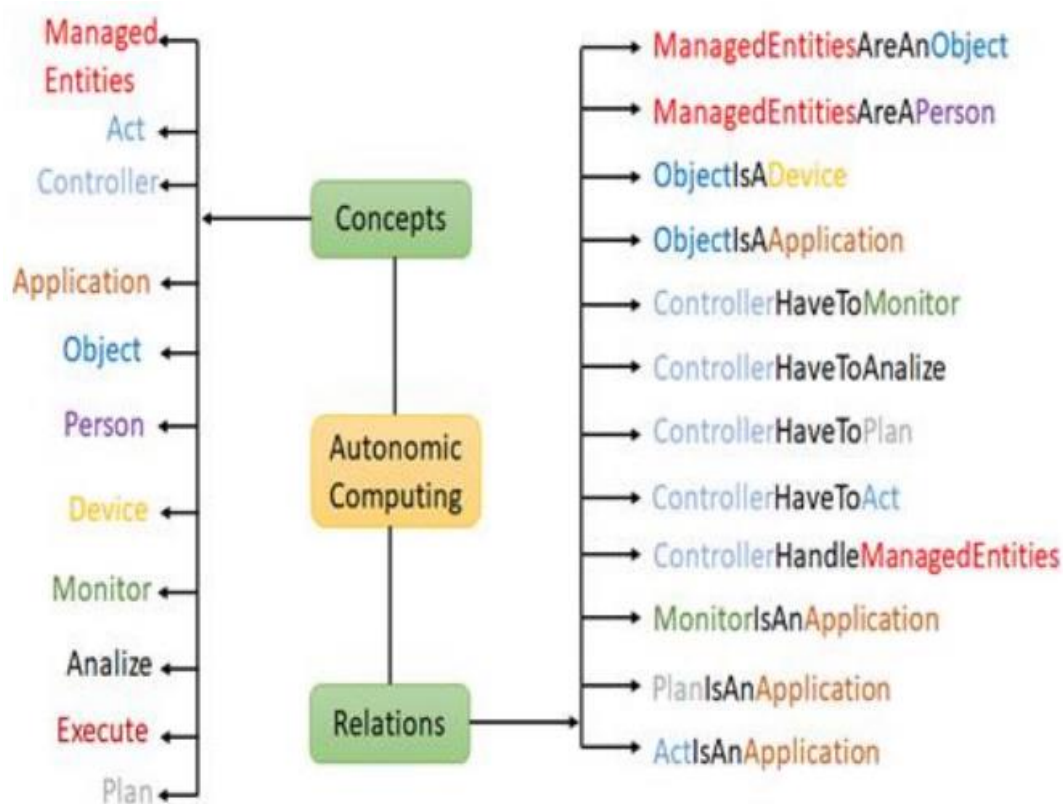


Figura 3.11. Ontología de La Computación Autónoma de la Arquitectura ARMNANO (ver ref. [2])

- i) Entidades manejadas (*managed entities*): se basa en los objetos considerados en el AmI;
- ii) Controlador (*controller*): define las diferentes tareas para el manejo de los objetos en el AmI. Se sustenta en el concepto del lazo MAPE+K [51] [12] [53];
- iii) Actuar (*act*): se refiere a la acción que es ejecutada en el AmI;
- iv) Aplicación (*application*): indica el software usado por las tareas del ciclo autónomo;
- v) Objeto (*object*): describe el hardware y software manejados en el contexto del AmI;

- vi) Persona (*person*): indica un actuador externo, ya sea un consumidor o un proveedor para ARMNANO;
- vii) Dispositivo (*device*): se refiere a uno de los hardware específicos que se emplean en el AmI;
- viii) Monitor (*monitor*): representa las tareas para rastrear las variables que definen el contexto;
- ix) Analizar (*analyze*): representa las tareas que se ejecutan después de la interpretación de los nanodatos,
- x) Ejecutar (*execute*): representa los comandos ejecutados en el AmI;
- xi) Planear (*plan*): representa las tareas que están programadas.

3.3.3. Modelo de datos multidimensional de DASS

A continuación, se describe el modelo de datos requerido por el DASS, para recopilar la información sobre el sistema donde se prestan los servicios. Para ello se utiliza un modelo multidimensional que se representa en un diagrama en la **Figura 3.12**:

- i) **Dimensión usuario**: se refiere a la persona o grupo de personas que reciben directamente los servicios de ARMNANO, donde se pueden implantar los nanosensores y nanoactuadores.
- ii) **Dimensión cosa/objeto**: describe los objetos donde se pueden implantar los nanosensores y nanoactuadores. Un objeto puede recibir servicios de ARMNANO. El objeto puede ser, en el contexto de casas inteligentes, un refrigerador o una cocina.
- iii) **Dimensión actividad**: describe los diferentes procesos en el entorno supervisado por ARMNANO, y en base a ellos, las posibles acciones y funcionalidades de los nanosensores (en general, las capacidades requeridas de los nanodispositivos).
- iv) **Dimensión temporal**: describe los parámetros como la frecuencia de medida, la granularidad de la unidad de tiempo utilizada por los nanosensores o nanoactuadores, entre otros.
- v) **Dimensión del dispositivo**: aquí se incluyen los diferentes nanodispositivos, como los nanosensores y los nanoactuadores. Describe sus funcionalidades, define sus características técnicas, entre otras cosas.
- vi) **Dimensión de servicio**: describe los servicios que brindan los nanodispositivos de ARMNANO.
- vii) **Dimensión variable**: describe las variables (nanodatos) que deben ser observadas, analizadas y modificadas, en el middleware ARMNANO. Normalmente, ARMNANO brinda servicios en base a las variables observadas en los objetos/personas. Este es un modelo genérico que puede ser utilizado en diferentes contextos

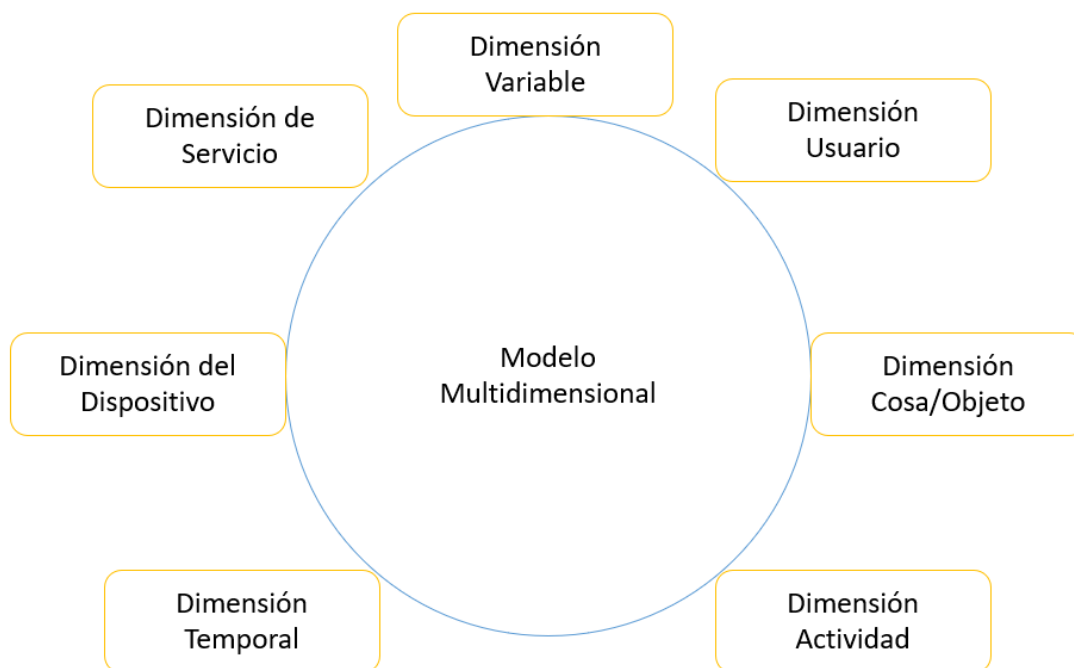


Figura 3.12. Dimensiones que compila DASS en su colección de información

Capítulo 4. Casos de Estudio en Ambientes nanotecnológicos

En este capítulo presentaremos como se despliega el DASS, como el CmNA ejecuta su modo de autenticación de nanodata, y mostrará la puesta en marcha de ARMNANO en un sistema que contempla data real.

4.1. Sistema Integral de Análisis de Datos en un Middleware Basado En Nanodispositivos (ARMNANO)

En este caso de estudio se contempla la aplicación de los 3 CAs para probar el alcance de cada uno. En este caso en particular, estos CAs se van a iniciar en una persona afectada con cáncer y que paulatinamente puede desmejorar, implicando nuevos estadios de la enfermedad, que requieren una analítica de datos diferente. El cambio de un CA a otro sugiere que la profundidad del análisis se incrementa con la finalidad de determinar nuevas tareas y disminuir el avance de la enfermedad. Un nivel más profundo de análisis datos sugieren medidas paliativas más eficaces.

En específico, el caso de estudio consiste en una mujer enferma de cáncer en un hospital que cuenta con una sala inteligente para brindar atención médica, la cual tiene desplegada a ARMNANO para monitorear constantemente variables de salud, con un arreglo de NSs distribuidos interna y externamente al paciente. Esta persona tiene el diagnóstico: “Paciente con cáncer de mama en etapa intermedia”.

ARMNANO actúa de manera descentralizada para monitorear constantemente al paciente. Como la afección de este individuo se denomina cáncer de mama, la propiedad de detección se dirige a controlar el biomarcador específico para este tipo de cáncer, proporcionado por la catepsina-D. En este sentido, la adecuación de ARMNANO a esta paciente consiste en el despliegue de los siguientes Nanodispositivos para esta paciente:

1. *Nanosensores*, es un arreglo de 12 NSs inyectados localmente en las mamas para monitorear la proteína Catepsina-D, a nivel de nmol /g (el nivel normal de catepsina-D en mujeres es de 10 nmol/g o menos)
2. *Nanoactuador*, es una malla compuesta por 12 NAs capaces de realizar principalmente 2 tareas: b.1) 6 NAs dedicados a destruir/ cortar tejido graso o tejido maligno (nódulos, tumores, etc.), b.2) 6 NA dedicados a inyectar localmente la quimioterapia.
3. *Nanorouter*, hay 2 Ro desplegadas para transmitir los valores de medida hasta el Mc en la frecuencia de observación de 1-6 min.

4. *Microgateway*, despliega el agente CmNA para reconocer la validez de la concentración de Catepsina-D medida en sangre.

Ahora se describe el despliegue de los Ciclo Autonómicos (CA).

CA de Acondicionamiento

- Tareas de Monitoreo: 1. Los 12 NSs realizan la medición regular. Los valores de concentración de catepsina-D son validados.
- Tareas de Interpretación: 1. Determinan si existe una condición anormal dada por un aumento de la concentración de catepsina-D.
- Toma de decisiones: 1. Una decisión es un tratamiento de quimioterapia suave que comienza con la aplicación de inhibidores químicos de forma local y no invasiva. Así, la paciente mujer es atendida de manera autónoma por las capacidades de este CA, ya que la arquitectura ARMNANO asegura funcionar adecuadamente en la paciente mujer con los puntos de ajuste iniciales.

CA de Optimización

En este escenario hay una distribución anormal de los valores medidos en la concentración de Catepsina-D. Así, puede desorientar una interpretación correcta en el paciente. Además, la tarea de autenticación de datos no converge rápidamente. Este escenario es una degradación de ARMNANO. Las continuas fallas en ARMNANO generan la invocación al CA de Optimización. Las tareas de CA para este caso de estudio se explican a continuación.

- Tareas de Seguimiento: 1. La observación se realiza de forma parcial. 2. Algunos valores no han sido debidamente autenticados.
- Tareas de interpretación: 1. Hay uno o más NS que fallan. Toma de decisiones: La falla es corregida y probada. los nodos NS se actualizan, sustituyendo los que fallan

por nuevos.

CA de Fijar Parámetros en Tiempo Real

La paciente mujer tiene una formación de nódulos críticos, de tal manera que la tasa de formación de células malignas siembra nuevos tumores. Es decir, en este escenario la paciente con cáncer de mama genera una formación abrupta de nódulo tumoral. Este CA permite determinar el nivel de gravedad del individuo en tiempo real, y realizar una toma de decisiones que permita disminuir los efectos adversos en el menor tiempo posible (principal ventaja de la observación continua con ARMNANO). Las tareas para este nuevo escenario son.

- Tareas de Monitoreo: Los 12 NSs miden la concentración de catepsina-D. Con estos valores se determina una formación crítica de tejido dañado.
- Tareas de interpretación: Ambas mamas muestran pérdida de consistencia y forma.
- Tareas de Toma de Decisiones: Se mandan 6 NAs para remover los nódulos que generan mala producción celular. Estos NAs actúan localmente para extirpar este tumor. Se ordena a 6 NAs que inyecten un inhibidor químico.

Este CA actúa hasta detener la aparición de nódulos, para eliminar físicamente el tejido dañado. Los NAs poseen la ventaja de actuar de forma no invasiva. El tejido dañado se corta desde el interior y se extrae con un tubo de cánula en ambos senos.

4.2. Caso de estudio de Autenticación de nanodatos del Middleware de Manejo de Nanodispositivos

En este caso de estudio se busca evaluar las capacidades de autenticación de nanodatos utilizando el protocolo de autenticación de datos (DAP en inglés, *Data Authentication Protocol*) que se presentó en el capítulo anterior. La data cruda está compuesta del registro de la fluorescencia que se realizó en un NS experimental, *ExpNS*, utilizando 2 tampones problema (una solución tampón consiste en la mezcla de 2 o más componentes químicos que son un ácido y una base, con el objeto de generar un pH constante, y que resiste la variación de pH ante adiciones pequeñas cantidades de ácido o bases concentrada) distintos que son Buffer Salino de Fosfina (PBS en inglés, *Phosphine Buffered Saline*) y Fluido Cerebroespinal Artificial (aCSF en inglés, *Artificial Cerebrospinal Fluid*). Estos tampones se seleccionaron por la afinidad que muestran con sistemas biológicos (mirar más detalles del dataset en [54]).

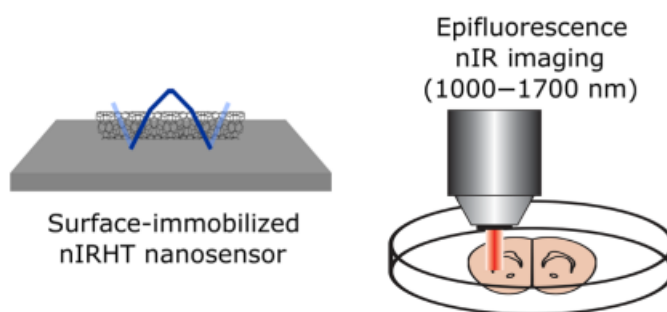


Figura 4.1. A la izquierda vemos el NS inmovilizado en la superficie. A la derecha se observa la punta del fluorómetro incidiendo la longitud de onda en el rango de 100 a 1700 nm. (ver [54])

Se conoce que *ExpNS* es un NS nanométrico a base de nanotubos de carbono, los cuales cumplen la función de conducir la corriente eléctrica, o cualquier señal que involucre flujo de electrones en la superficie del NS que está en contacto con la solución química. La data de esta experimentación fue provista por un grupo de la Universidad de Berkeley dirigido por la Prof. Markita Landry [54], quienes nos la cedieron para nuestras pruebas. En particular, nosotros pretendemos visualizar si la data de medición de *ExpNS* en las 2 soluciones tampón puede autenticarse, lo cual probaría la eficiencia de nuestro DAP en la autenticación de nanodata real. En la **Tabla 5** se muestra una fracción de la data cruda monitoreada por *ExpNS* (PBS y aCSF). Allí se destaca la medición de la fluorescencia (I/I_0) en función de la concentración de serotonina o 5-hidroxitriptamina (5-HT). En la **Figura 4.1** se muestra el esquema del *ExpNS* sobre la superficie y la punta del instrumento que incide con el rango de longitudes de onda en el infrarrojo.

Para entender dicha data completamente, asumimos que se despliega el CA más básico que posee ARMNANO, dado por el CTM. Consideramos este CA porque este ciclo busca introducir los parámetros de funcionamiento del middleware en el contexto de las muestras analizadas, y luego ejecutar sus tareas de funcionamiento para comprobar que las medidas realizadas son acordes. Una corrida de tareas del CTM determinará que el sistema es estable, y no necesita invocar ciclos posteriores como el ciclo de optimización o el de fijación de parámetros en el middleware.

Tabla 5. Primeros 20 datos del dataset de la fluorescencia I/I₀ medida en los tampones de PBS y aCSF en el rango de concentraciones de 0.1 a 2 μ M

Concentración de Serotonina (μ M)	Fluorescencia en el tampón de PBS	Fluorescencia en el tampón aCSF
0.1	0.02489	0.00803
0.2	0.05148	0.01915
0.3	0.07831	0.03177
0.4	0.10504	0.04541
0.5	0.13151	0.05892
0.6	0.15762	0.07483
0.7	0.18334	0.09034
0.8	0.20862	0.10624
0.9	0.23344	0.12246
1.0	0.25779	0.13895
1.1	0.28168	0.15565
1.2	0.30508	0.17254
1.3	0.32802	0.18956
1.4	0.35049	0.2067
1.5	0.37250	0.22391
1.6	0.39405	0.24119
1.7	0.41515	0.25851
1.8	0.43582	0.27584
1.9	0.45606	0.29318
2.0	0.47588	0.31051

Dado que el estudio se realiza en diferentes tampones, tal como muestra la data cruda de la **Tabla 5**, encontramos que la respuesta es mayor en la solución aCSF, alcanzando 2.5 unidades, que en la solución PBS, donde la fluorescencia alcanzó 1.9 (ver **Figura 4.2**). La estrategia para la autenticación de datos debe contemplar algo más que tomar la data en todo el rango de concentraciones de 5-HT, y más que nada considerando la no planaridad del gráfico. De esta forma se optó por dividir la curva en 3 zonas con el objetivo de encontrar modelos más distintivos de la realidad, más que un modelo global que dictaminaría una idea dispersa y poco representativa. Como dijimos antes, el DAP

considera 2 módulos para autenticar, donde el primero genera una perspectiva estadística, mientras que el segundo establece un modelo probabilístico basado en una distribución gaussiana. Ambos se aplican a las diferentes secciones (zonas) de todo el registro, que están numeradas en la **Figura 4.2**.

Nuestro análisis indicó que seccionar en más de 3 implicaría mucho trabajo para un simple conjunto de datos, y seccionarlo en menos de 3 sería poco representativo. En la **Figura 4.2** está representado el comportamiento de la fluorescencia medidos con el NS en las soluciones tampón de *PBS* y *aCSF*, donde visualizamos que la respuesta es mayor en la solución de *aCSF*. La señal en el eje de las ordenadas tiende a ser constante cuando la [5-HT] es muy grande, además, la pendiente es positiva en todo el rango de mediciones.

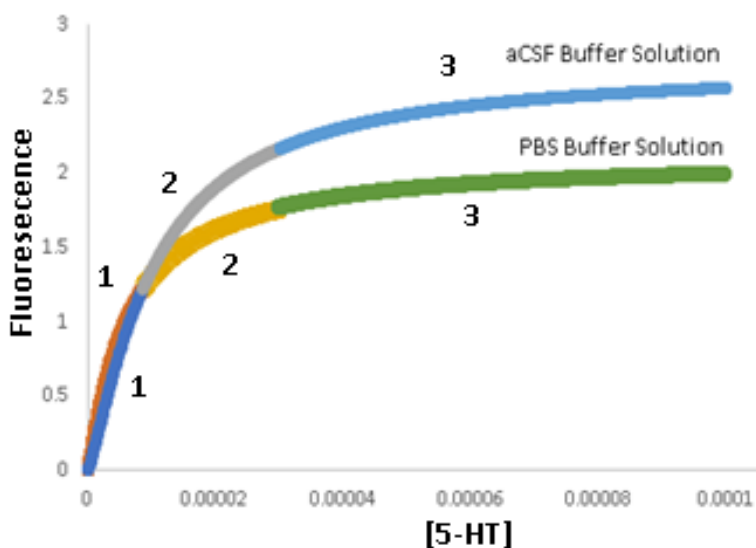


Figura 4.2. Rendimiento del nanosensor experimental (*ExpNS*) en las soluciones tampón de PBS y aCSF (**Ver ref. [4]**)

Aplicando el DAP, se observa que la pendiente aumenta más rápido en la zona 1 que en la zona 2, y en esta última es mayor que en la zona 3 (ver **Tabla 5** y **Figura 4.2**). El aplanamiento de las curvas a diferentes alturas indica que el NS se satura a distintas velocidades, encontrando un $\Delta I/I_0 = 0.6$ (este valor se encuentra sustrayendo el máximo de fluorescencia en *aCSF* menos el máximo en *PBS*) cuando se comparan los puntos superiores de las curvas. Una saturación del NS sugiere que el sensor ya no detecta cambios ante adiciones sucesivas de 5-HT dado que todos los sitios activos en la matriz que mide están ocupados. Incluso si se adiciona más 5-HT es indetectable, enviando en consecuencia la misma magnitud de la señal.

Tabla 6. Métrica estadística de las 3 zonas para los buffers de *aCSF* y *PBS*

PBS	3	m	σ	Q1	Q3	IQR
		0.76015	0.349721	0.504791	1.059688	0.554897
	2	m	σ	Q1	Q3	IQR
		1.570572	0.147174	1.467194	1.695923	0.22873
	1	m	σ	Q1	Q3	IQR
		1.91324	0.060556	1.87375	1.963918	0.090168
aCSF	3	m	σ	Q1	Q3	IQR
		0.652538	0.371282	0.336458	0.979616	0.643158
	2	m	σ	Q1	Q3	IQR
		1.821236	0.262526	1.630112	2.047003	0.41689
	1	m	σ	Q1	Q3	IQR
		2.445552	0.108284	2.37672	2.535587	0.158867

En la **Tabla 6** se representan las métricas estadísticas del módulo 1 calculadas en ambos tampones. Estos datos revelan que el valor de m disminuye al pasar de la zona 1 a la zona 2, y luego hasta la zona 3, sin embargo, σ aumenta. Para *PBS*, σ es 0.060 en la zona 1, 0.147 en la zona 2 y 0.349 en la zona 3, mientras que para el tampón *aCSF*, σ es 0.108 en la zona 1, 0.262 en la zona 2 y 0.371 en la zona 3, lo cual demuestra que la desviación estándar solo incrementa ligeramente entre las zonas, y aumenta en la zona de planaridad por que el rango de concentraciones considerado es más extenso.

El módulo 1 no detecta valores atípicos, lo que indica que el criterio de $1.5 \cdot \text{IQR}$ incluye todos los datos. Este resultado habla de la robustez de este módulo dentro de todo el DAP, considerando que el análisis se está haciendo en una data que varía con pendiente positiva en todo su rango y no es plana.

Tabla 7. Métrica de calidad en el caso de estudio 2

	Clase	P	R	F-m	A
E6#9 en PBS. Zona1	Falla	100	56.2	0.72	89.6
D6#8 en aCSF. Zona1	Falla	100	52.9	0.692	91.9
E6#9 en PBS. Zone2	Pasa	100	77.4	0.872	95.7
D6#8 en aCSF. Zona2	Pasa	100	85.1	0.92	98.1
E6#9 en PBS. Zona3	Pasa	100	94.1	0.969	99
D6#8 en aCSF. Zona3	Pasa	100	97.3	0.967	98

Pasa si $F-m > 80$, si no entonces **Falla**. **P**: precision, **R**: recall, **F-m**: f-measure, **A**: exactitud

Con la finalidad de avalar los resultados derivados con el DAP se calcularon las siguientes métricas de calidad: precisión (p), recall (R), el F-measure (F-m), y exactitud (A), para determinar si el dato fallo o pasa. En la **Tabla 7** se exponen los resultados con la finalidad de determinar la calidad del análisis en cada fracción de datos. La clase determina la evaluación global de la zona (zona 1 PBS, zona 1 aCSF, etc), bien sea que pertenezca al tampón PBS o el aCSF. F-m se calcula a partir de la precisión y el recall utilizando la siguiente fórmula:

$$F - m = \frac{2}{recall^{-1} + precision^{-1}}$$

De esta forma, F-m es una medida de calidad general del resultado obtenido, indicando que si $F-m > .75$, entonces *pasa* (el valor es aceptado), y si $f-m < .75$ entonces *falla* (el valor no es aceptado). F-m es basado en la precisión ($p = tp/(tp + fp)$) y el recall ($p = tp/(tp + fn)$), donde *tp* es verdadero positivo, *fp* es falso positivo, y *fn* es falso negativo.

La exactitud es $A = tp + fp / (tp + fp + tn + fn)$. El tp se refiere a un valor típico de la data evaluada aceptado apropiadamente; el fn se refiere a los valores indicados como atípicos pero que no eran; y fp se refiere a los valores aceptados que deberían haber sido rechazados. De las 6 zonas destacadas en la **Tabla 7**, la zona 1 es la que presenta peor métrica, indicando en la clase que ambas *Fallan*, dado que aplicando el criterio del F-m la zona 1 PBS tenía 0.72 y la zona 1 de aCSF tenía 0.69, siendo ambos valores menores a 0.75 como criterio de referencia. Esto concuerda con el hecho de que la zona 1 en ambos tampones fueron las zonas con mayor cantidad de valores atípicos presentes y a que la pendiente era la más pronunciada, siendo la zona de menos planaridad. Ya en la zona 2 en *PBS* y *aCSF*, los valores de F-m incrementan a 0.87 y 0.92, respectivamente, estableciendo que ambas zonas *pasan*. Y en la zona 3, siendo la zona de más planaridad la data autenticada *pasa*, lo que implica que la métrica corrobora efectivamente el DAP que se estableció y que determinaba cuales valores pasan y cuáles no. Adicionalmente, estas métricas corroboran la estrategia de seccionar todo el comportamiento en zonas, considerando que una perspectiva global intentaría linealizar sin sentido el comportamiento creciente de la curva, y probablemente indicando un F-m global muy bajo.

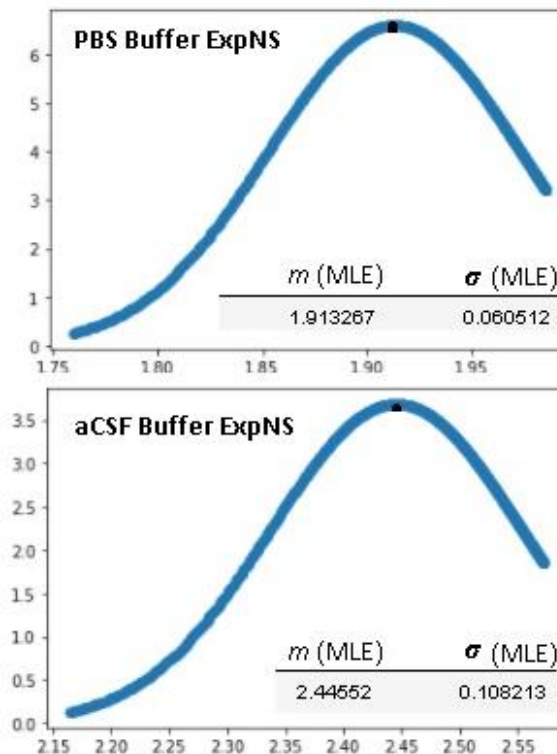


Figura 4.3. Cálculo en el modo de funcionamiento (modelado con MLE) para el 2^{do} módulo en la zona 1. (ver ref. [4])

Luego de aplicar el módulo 2 se genera la **Figura 4.3**, que representa la curva gaussiana que resulta con la técnica MLE en la zona 1. Se muestra esta zona solamente como representación del método, pero se procede de igual forma en cada fracción de datos señalada en la **Figura 4.2**. Las coordenadas de la curva gaussiana en la zona 1 fueron $(m, \sigma) = (1.913, 0.060)$ en el buffer *PBS* y $(2.445, 0.108)$ en el buffer *aCSF*. Estas coordenadas revelan las características del modelo MLE, y cuyo análisis puede extrapolarse al resto de las zonas. Dada la mayor pendiente en la zona 1 obtenida por MLE, la cantidad de valores atípicos es mayor aquí comparado a las otras zonas. El número de valores atípicos disminuyó siguiendo el orden Zona1 > Zona2 > Zona3, lo que concuerda con el hecho de que la curva se va aplanando hasta que llega a la zona 3.

4.3. Un Sistema Inteligente de Análisis De Datos para el Despliegue Óptimo de Nanosensores en el Contexto de una Aplicación de eSalud

A continuación, detallamos el despliegue de ARMNANO invocando el CA más básico, el CTM. Consideramos este CA porque está encargado de acondicionar los parámetros del middleware, indicando inicialmente si la determinación del contexto invocando las propiedades de ARMNANO está correcta. En particular, este CA es responsable de determinar la configuración del entorno [5] y de instanciar ARMNANO en función del contexto.

4.3.1. *Descripción del Caso de Estudio*

En el campo de eHealth, uno de los más interesantes está relacionado con las enfermedades hepáticas, es decir, aquellas enfermedades relacionadas con la insuficiencia hepática. Las estadísticas actuales demuestran que los individuos aumentaron su consumo de alcohol en los años 2020 y 2021, a nivel mundial [55] [56], lo que está generando varias enfermedades vinculadas al hígado [57] [58]. Vamos a probar ARMNANO en un conjunto de datos que define a personas de diferentes edades y géneros para obtener un "panorama general" y una "vista final" sobre los estados hepáticos de los pacientes. Nuestro conjunto de datos incluye 347 individuos con respecto a cinco indicadores del estado hepático. Cuatro de ellos son enzimas y uno es el Volumen Corpuscular Medio (MCV). Estas variables se representan a continuación con su rango normal.

- **Enzima SGOT:** es una enzima que se libera cuando hay daño tisular. Su concentración aumenta drásticamente en casos de herida o daño. Una magnitud en 50 unidades/L para hombres o 45 para mujeres indica daño.
- **Enzima SGPT:** es una enzima que se produce en varios órganos, pero su mayor concentración se encuentra en el hígado. Cuanto mayor sea la concentración de esta enzima, mayor será la tendencia al daño hepático.
- **Enzima ALKPHOS:** su función es catalizar la hidrólisis del fosfato en el suero a pH alcalino. ALKPHOS es una variable específica para el hígado.
- **Enzima Gammagt:** monitorea el bloqueo de los conductos biliares. Las mediciones superiores a 30 UI/L están directamente relacionadas con un daño hepático.

- **MCV:** es una medida del volumen de las células en el hígado (medido en femtolitros, fL). Por encima del nivel de 90 fl se considera anemia macrocítica.

Para probar las capacidades de la 3ra tarea en el CTM, consideramos solo las variables ALKPHOS y MCV, por varias razones, entre ellas, que el criterio de selección para identificar daño hepático es cuando la concentración de enzimas (cualquiera de ellas) es anormalmente alta, lo que puede determinarse usando ALKPHOS (compuesto químico) y MCV (característica geométrica). Representan dos cosas diferentes medidas en un mismo sistema, dando heterogeneidad a nuestro enfoque, además de manejar menos información y descartar posibles fuentes de errores. Basado en el diseño del ARMNANO, un conjunto de NSs se inyecta localmente en el hígado del paciente para realizar las mediciones. Los nodos NSs seguirán las dos variables previamente definidas.

4.3.2. Descripción del Ciclo Autonómico de Condicionamiento (CTM)

Este CA debe determinar los NSs a implementar para las tareas específicas de análisis de datos que se realizan en el contexto (Ver **Figura 4.4**) para la instanciación del caso de estudio. Se compone de 4 tareas, tal y como se define en [3].

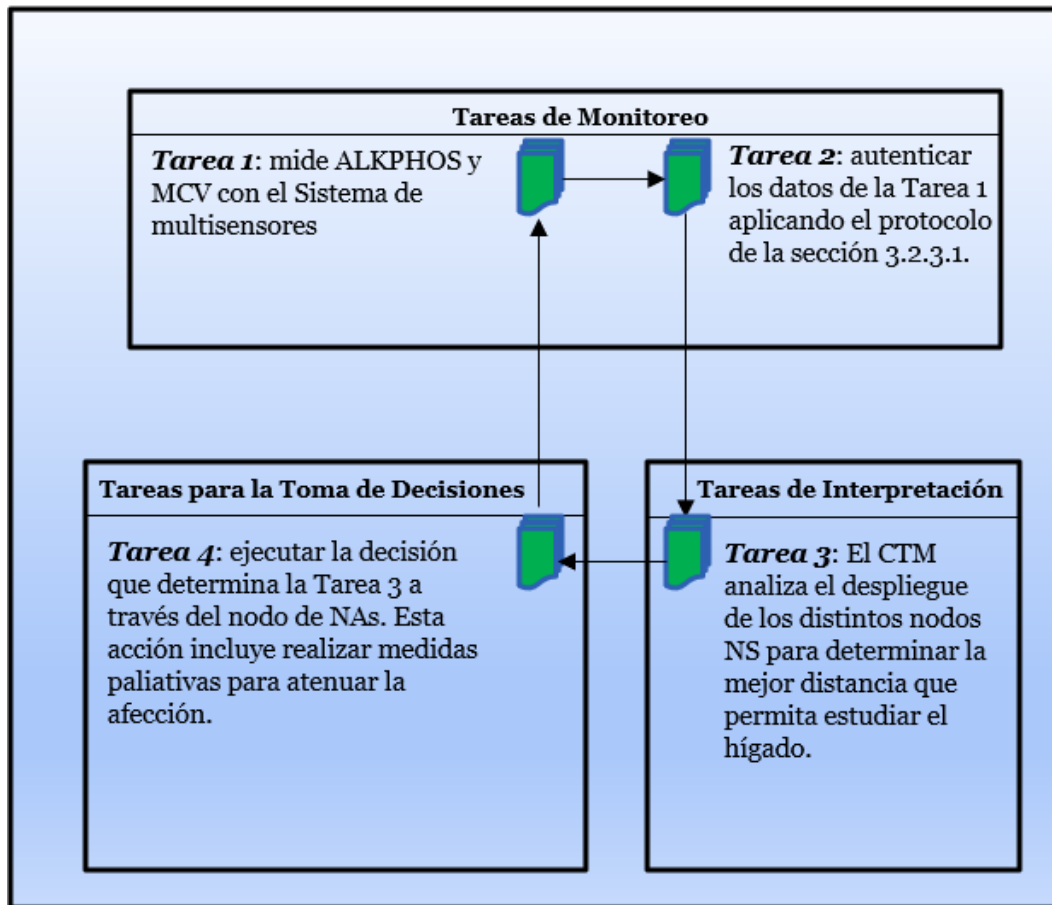


Figura 4.4. Tareas para el ciclo autónomo CTM

ARMNANO tiene tareas específicas de análisis de datos en cada CA. La descripción en las tareas basadas en CTM es la siguiente:

- **Tarea 1**, usar los sistemas multisensores NSs en el hígado para medir ALKPHOS y MCV: medimos las variables objetivo usando los nodos NSs.
- **Tarea 2**, Autenticar los datos de la Tarea 1 usando el protocolo de autenticación de ARMNANO, el cual se ejecuta para determinar si los valores de los datos son apropiados en las variables medidas.
- **Tarea 3**, El CA de CTM considerará el despliegue de tres NSs, cinco NSs, diez NSs, veinte NSs o treinta NSs en el hígado del paciente. Por lo tanto, nuestro objetivo es determinar la distancia en el hígado (Distancia desde el final del hígado hasta la periferia, DCH) para aplicar un tratamiento localizado.
- **Tarea 4**, Ejercer la decisión en el hígado mediante el uso de NAs actuando localmente: comprende la acción paliativa de los NAs para disminuir el problema hepático, por ejemplo, inyectar inhibidores de la enzima ALKPHOS.

4.3.3.1. Especificación de las tareas de análisis de datos

De las tareas especificadas arriba, a continuación, se va a desarrollar la número 3, la cual comprende aplicar la analítica de datos en la información ya autenticada y concluir sobre dicha data cual es la distancia más óptima para colocar los indicadores enzimáticos en el hígado. Esta tarea de analítica de datos comprende la utilización de Algoritmos Genéticos (AG).

Un AG es un proceso metaheurístico basado en la selección natural, el cual se usa comúnmente para resolver problemas de optimización [54]. Los AGs usan operadores inspirados en la forma en que combina la información genética, tales como mutación y cruce, para generar soluciones de alta calidad. Un AG está compuesto de una población de individuos (soluciones candidatas a un problema de optimización). Cada individuo tiene un conjunto de propiedades (componentes del cromosoma), que determinan su calidad mediante una función de aptitud. El AG se ejecuta en un proceso de evolución en que participa los individuos de la población iterando (cada iteración se llama generación o época). En cada generación, la calidad de cada individuo de la población se evalúa mediante la función de aptitud (normalmente, es la función objetivo del problema de optimización a resolver). Se seleccionan los mejores individuos y cada cromosoma se modifica utilizando los operadores genéticos. De esta manera, se construye una nueva generación para ser utilizada en la siguiente iteración. Frecuentemente, el algoritmo finaliza según un número máximo de generaciones, o cuando se ha alcanzado un buen valor de aptitud (fitness).

En nuestro caso, un cromosoma se define como un vector de números reales, de tamaño igual a 2, cuyos elementos serán: ALKPHOS, MCV y DCH.

$$Chromosome = [ALKPHOS, MCV]$$

Donde los rangos clínicos generales para un hígado sano están entre $44 \leq ALKPHOS \leq 200$ y $80 \leq MCV \leq 140$. Por otro lado, los valores ideales de DCH para determinar con precisión el estado del hígado deben ser $0 \leq DCH \leq 160.000$ (existe un equivalente entre este valor extremo de DCH con el valor longitudinal promedio de 14,5 cm). Aplicamos la capacidad evolutiva de los AGs para determinar la cantidad de NSs que se deben utilizar en personas con enfermedades en el hígado, y la distancia DCH más óptima donde los mismos deben insertarse. Las variables independientes que se incorporan en el AG son ALKPHOS y MCV, y así poder estudiar el valor de DCH para monitorear posibles daños en el hígado.

La función objetivo en nuestro caso inicial determina el DCH en base a las variables ALKPHOS y MCV. Según la literatura [57] [58] [6], esta función es:

$$FF = DCH = 0.05 * ALKPHOS^2 * MCV$$

Donde se usa la constante 0.05 para eliminar las unidades de las variables. La optimización ocurre cuando se minimiza la función objetivo FF ($\min(ff)$), lo que significa que se obtienen los valores ideales de ALKPHOS y MCV para calcular DCH.

Así, el cromosoma utilizado depende de 2 variables. ALKPHOS que representa una enzima que existe principalmente en el hígado, y MCV que es una variable física que mide el volumen celular. En la mutación directa se sustituye aleatoriamente los valores de los elementos del cromosoma tomados de las bases de datos en [59], [58] y [6] con aquellos existentes en las medidas que se hacen al paciente. Para la optimización de los hiperparámetros del AG se ha utilizado el siguiente conjunto de valores.

- Número de Generaciones: [10, 50, 100, 150, 200]
- Tamaño de la Población: [10, 30, 50, 70, 100]

Los operadores genéticos usados fueron el operador de cruce y el de mutación. La probabilidad de usar el de cruce fue de 0.8 y de usar la mutación de 0.3. El operador de cruce era un operador clásico en el que se seleccionó aleatoriamente 1 punto en los 2 padres y se intercambiaban sus fragmentos. Por otro lado, el de mutación consistió en modificar aleatoriamente los valores en un fragmento del individuo cuyos 2 puntos extremos del fragmento en el cromosoma fueron también escogidos aleatoriamente.

4.3.3. Resultados Numéricos y Análisis

Para analizar el comportamiento de este ciclo autónomo se usaron 3 bases de datos que están en [6], [58], [59]. Las bases de datos [6] y [58] son de pacientes con enfermedades de hígado, el conjunto de datos de [6] con 6000 pacientes, y el conjunto de datos de [58] con 32000 personas. La base de datos de [59] es de pacientes sin afección detectada en el hígado, con alrededor 3000 pacientes.

4.3.3.2. Descripción del caso 1

En el primer caso de prueba se utiliza un solo sensor por variable (1 NS). Ejecutamos la optimización de hiperparámetros cada vez que se ejecuta el AG [5]. Por ejemplo, los mejores valores para cada parámetro con el conjunto de datos de [59] fueron de 10 ciclos (generaciones) y un tamaño de la población de 100.

En la **Tabla 8** se muestra que la mejor distancia desde un extremo del hígado hasta la periferia, usando los valores de ALKPHOS y MCV para uno de los pacientes en el conjunto de datos [59], es 24.225. Esto significa que con un NS se puede monitorizar el hígado de ese paciente sin problema. La **Tabla 8** muestra el mejor individuo encontrado en el caso 1.

Tabla 8. Valores de MCV y ALKPHOS en el mejor individuo del caso 1

Dataset	ALKPHOS	MCV	DCH (FF)
[59]	76	85	24225,00

La **Tabla 9** revela además 2 cosas. Para el primer conjunto de datos [58] se obtiene un DCH muy grande en el paciente monitoreado; es decir, el valor está localizado en la periferia del hígado. Esto indica que se puede inyectar un solo sensor en dicho paciente. Sin embargo, lo más relevante es que, en el segundo conjunto de datos extraídos de [6] nuestro AG no puede lograr un DCH, ni siquiera en su periferia, dado que el valor detectado en el mejor individuo fue de 168750, y ello está fuera del rango de 0 a 160000 para monitorear a este paciente. Esto sugiere que debemos intentar con más sensores, y establece por tanto el objeto de estudio del caso 2, donde se valorará el número de NS que se requieren y los valores óptimos de DCH donde se inyectan los mismos para hacer un mejor diagnóstico del hígado.

Tabla 9. Mejores individuos encontrados utilizando las bases de datos de personas afectadas en el hígado

Dataset	ALKPHOS	MCV	DCH (FF)
[58]	146	99.45	106092,42
[6]	150	120	168750,00

4.3.3.2. Descripción del caso 2

En este caso de estudio la idea es determinar la cantidad adecuada de sensores para poder monitorear a un paciente enfermo. En particular, vamos a utilizar varios pacientes del conjunto de datos [6], que no se pudo monitorear con un NS. Por otro lado, en este caso, la longitud del cromosoma es variable, por lo que su longitud n depende del número de NS utilizados en esa ejecución de AG. Así, el cromosoma t se representa como un vector de números reales, con un tamaño igual a n .

Cromosoma = [ALKPHOS1, MCV1, DCH1, ALKPHOS2, MCV2, DCH2, . . . , ALKPHOSn, MCVn, DCHn]

La función objetivo se establece en función del número de NS, por tanto, su planteamiento es como una sumatoria que va de 0 a n donde los sumandos son la contribución de cada NS, dada por $ALKPHOS_i^2 * MCV_i$. En consecuencia, el criterio de parada es equivalente a el caso de 1 NS, donde se busca minimizar el total de contribuciones NS que describe la función FF abajo. Formalmente esta función es:

$$FF = 0.05 * \sum_{i=1}^n ALKPHOS_i^2 * MCV_i$$

Tabla 10. Los mejores individuos encontrados en el caso 2 para el dataset [6]

Number of NSs	DCH (FF)
3	162500
4	156500
5	132000
10	132000

La **Tabla 10** indica que se requieren al menos cuatro NSs para monitorear correctamente el hígado. Además, indica que tener más de cinco NSs no mejora la precisión de los resultados del monitoreo de la afección hepática.

Algo interesante de determinar es cómo implementar el número de NS óptimo en el estudio de la mejor distancia DCH para realizar las mediciones de ALKPHOS y MCV. Para ello se ha seccionado el hígado en 4 zonas de relevancia, tal como se indica en la **Figura 4.5**, y así poder asignar cuál de ellas toma importancia en el diagnostico a partir de la salida del AG. Por ejemplo, en el caso de cuatro NS, de acuerdo al mejor individuo de la **Tabla 10**, los 4 NS deberían inyectarse en una de las 4 zonas: A, de 0 a 40000; B, de

40000 a 80000 ua; C, de 80000 a 120000 ua; o D, de 120000 a 160000. Por tanto, se espera que el diagnóstico aumente la certeza, que es lo que quiere la Tarea 3 del CA CTM, y el tratamiento sea más adecuado, que es lo que busca la Tarea 4.

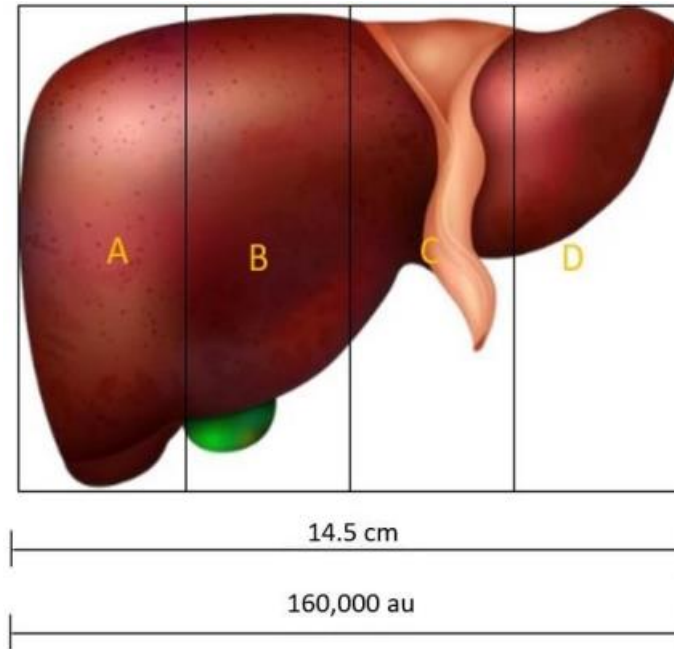


Figura 4.5. Hígado seccionado en 4 regiones para inyectar los NSs en cada una (ver ref. [5])

El comportamiento de un sistema multisensor determina las zonas donde se desplegarán los NSs. Por otro lado, un número pequeño o grande de NSs no es indicativo de un DCH óptimo, sino que es un número que debe determinarse (en nuestro caso, 4 o 5 NSs). En nuestro ejemplo de la **Figura 4.5**, escalando al tamaño del hígado de 14,5 cm, el AG determina las zonas en las que se debe inyectar los sensores en un paciente enfermo específico, tal como se muestra en la data de la **Tabla 11**. Utilizando la misma función objetivo de un sistema multisensor, pero evaluando el AG con diferentes pacientes, se tiene que el número de NS **apropiado** que persiste es de 5. Este resultado es bastante lógico, dado que la distancia DCH es muy similar entre los sistemas que contienen 5 NSs (ver **Tabla 11**), y los valores de NS son próximos entre si indicando que la cantidad óptima de NS está oscilando en ese número de 5. Un mayor volumen de datos revelaría una zona más precisa, pero se requiere una mayor cantidad de NS y posiblemente seccionar el hígado en una mayor cantidad de zonas que aumenten la precisión del resultado. Es claro con este resultado que el CA invocando las propiedades del AG es robusto con un sistema multisensor, y que en este caso específico la región óptima para realizar el diagnóstico es en la D (de 120000 a 160000).

Tabla 11. El mejor individuo encontrado en el caso 2 para el conjunto de datos [6] y diferentes pacientes.

Patient	Number of NSs	DCH (FF)
1	4	156500
2	5	152500
3	3	158000
4	5	148000
N	5	155500

La **Tabla 11** muestra valores muy próximos de DCH entre sí, a pesar de que se consideran distintos pacientes que van desde 1 hasta N . La salida en cada caso es un número de NS que varía entre 3 y 5, lo cual tampoco representa una diferencia remarcable, y que se justifica considerando que los valores de ALKPHOS y MCV son similares entre sí en pacientes enfermos del hígado, y que tienen dichos indicadores fuera de rango. De esta forma el CA establece las Tareas 3 y 4 incluso en pacientes que son diferentes y que hacen uso de las propiedades del CA CTM instanciado en afectados del hígado e invocando las propiedades de ARMNANO.

Capítulo 5. Conclusiones y trabajos futuros

5.1. Conclusiones

En la realización de este trabajo se han realizado varias investigaciones que están concatenadas entre sí y tienen que ver con el ensamblaje de sistemas nanodistribuidos para dar soluciones a problemas de índole variado utilizando CAs de tareas de análisis de datos. De esta forma, las conclusiones más resaltantes de este trabajo son las siguientes:

- Los middlewares pueden gestionar un sistema nanoensamblado que contemple NSs y NAs, siempre que se incorporen otras capas adicionales que den soporte a la integración de estos nanodispositivos en las tareas que el middleware va a ejecutar.
- La Capa CmNA es una capa fundamental de ARMNANO responsable de la comunicación de nanodatos. Entre otras cosas, participa en la evaluación de la calidad de los nanodatos y en el establecimiento de mecanismos de autocorrección del middleware. Particularmente, el CmNA ejecuta un agente denominado NaS que posee la capacidad de autenticar la nanodata, siendo un filtro de la información que se mide en el nodo NS para posteriormente transferirlo al DASS.
- La autenticación de los nanodatos se pudo realizar efectivamente en sistemas que se consideraban muy sensibles dado que existen en la escala nanométrica. En la nanoescala se pudo medir los niveles de la serotonina en NSs cuya matriz era sensible a dicha sustancia y que detectaba los cambios de concentración de la misma expresados en la magnitud de fluorescencia. La autenticación se realizó efectivamente en los valores de fluorescencia obtenidos en la nanoescala aplicando un protocolo que contenía dos módulos cuyos argumentos se complementaban entre sí. Una primera perspectiva evaluaba la certeza estadística de los valores de fluorescencia determinando su precisión y exactitud, y el segundo módulo consideraba el ajuste del dataset a un modelo probabilístico denominado Estimación de Máxima Similitud. La aplicación de ambos módulos demostró ser eficiente en la autenticación de data que no era totalmente plana o lineal. Esta estrategia fue muy efectiva y determinó una mayor cantidad de data atípica en la zona menos lineal de la curva.
- Se proponen 3 CAs que se ejecutan en el DASS de manera sucesiva que son, el Acondicionante (CTM), el Optimizante (OTM), y el de Fijar Parámetros en Tiempo Real (FTM). Estos ciclos constituyen la parte central de ARMNANO dado que las tareas que realiza nuestro middleware están contenidas como tareas de dichos ciclos, y simplemente cada tarea es invocada por los agentes específicos que requieren las mismas. Los ciclos se ejecutan desde el principio para insertar los parámetros iniciales con el CTM, e intentar dar soluciones si los valores esperados en la variable estudiada están fuera de rango. El OTM intenta optimizar si la falla

en la o las variables persiste, para lo cual ejecuta técnicas de analítica de datos más profundas a modo de entender la falla. Luego, el FTM fija definitivamente los parámetros para forzar al ciclo a actuar de la forma adecuada. Así, los CAs definen el comportamiento autónomo de ARMNANO, que pueden implementarse a nivel nanométrico en contextos genéricos.

Entre las recomendaciones de trabajos futuros más relevantes se pueden mencionar:

- Realizar pruebas de desempeño de ARMNANO utilizando nanodispositivos reales.
- Incorporar un sistema dentro de la capa DASS que permita optimizar las tareas de análisis de datos (meta-aprendizaje).
- Desarrollar un sistema que permita optimizar la cantidad de nanodispositivos que se despliegan en el entorno, para minimizar la cantidad de los mismos desplegados sin degradar la capacidad de observabilidad del entorno.
- Diseñar un sistema de supervisión del middleware que permita autónomamente auto-repararse, auto-optimizarse, auto-configurarse, usando los mismos principios de ciclo autonómico desarrollado para las aplicaciones basadas en nanodatos a las cuales le brinda servicio el middleware.
- Probar el middleware en otros contextos que no sean necesariamente el de salud, como el de los salones inteligentes, casas inteligentes, entre otros.

Referencias

- [1] A. Lopez Pacheco y J. Aguilar Castro, «NANO-Communication Management System for Smart Environments,» *Revista Venezolana de Computación*, vol. 5, pp. 12-22, 2018.
- [2] A. Lopez Pacheco, J. Aguilar, E. Puerto y R. Garcia, «An ontological model based on the ontology driven architecture paradigm for a middleware in the management of nano-devices in a smart environment,» *Journal of Physics: Conference Series. 5th IMRMPT*, vol. 1386, pp. 1-7, 2019.
- [3] A. Lopez pacheco, J. Aguilar, A. Pinto y J. Torres, «Data Analysis Smart Systems in a Nanodevices - Based Middleware,» *Contemporary Engineering Sciences*, vol. 11, pp. 4665-4679, 2018.
- [4] A. Lopez y J. Aguilar, «A Nanodata Authentication Approach for a Management Middleware of NANODEVICES,» *International Journal of Nanomanufacturing*, p. 1, 2023.
- [5] A. Lopez y J. Aguilar, «A Data Analysis Smart System for the Optimal Deployment of Nanosensors in the Context of an eHealth Application,» *Algorithms*, vol. 16, pp. 1-13, 2023.
- [6] S. Bellantini, C. Tiribelli, G. Saccoccio, M. Sodde, N. Fratti, C. De Martin y G. Cristianini, «Prevalence of Chronic Liver Disease in the General Population of Northern Italy. The Dionysos Study,» *Hepatology*, vol. 20, pp. 1442-1450, 1994.
- [7] J. Aguilar, M. Cerrada, G. Mousalli, F. Rivas y F. Hidrobo, *A Multiagent model for intelligent distributed Control Systems*, Lecture Notes in Artificial Intelligence, Springer-Verlag, 2005, pp. 191-197.
- [8] J. Martinez, M. Faminiliar, I. Corredor, A. Garcia, S. Bravo y L. Lopez, «Composition and Deployment for eHealth Services over Wireless Sensor Networks,» *Mathematical and Computer Modelling*, vol. 53, pp. 485-503, 2011.
- [9] M. Shamsipur, A. Barati y Z. Nematifar, «Fluorescent pH nanosensors: Design strategies and applications,» *Journal of Photochemistry and Photobiology C: Photochemistry Reviews*, vol. 39, pp. 76-141, 2019.
- [10] F. Dressler y S. Fischer, «Connecting in body nanocommunication with body area networks: challenges and opportunities of the Internet of NanoThings,» *Nano Communication Networks*, vol. 6, pp. 29-38, 2015.

- [11] V. Sokolova, G. Mekky, S. Van der Meer, M. Seeds, A. Atala y M. Epple, «Transport of ultrasmall gold nanoparticles (2 nm) across the blood–brain barrier in a six-cell brain spheroid model,» *Scientific Reports*, vol. 10, pp. 18033-18036, 2020.
- [12] J. Aguilar, M. Sanchez, C. J., P. Valdiviezo-Diaz, L. Barba-Guaman y L. Chamba-Eras, «Learning analytics tasks as services in smart classroom,» *Universal Access in the Information Society*, vol. 17, p. 693, 2018.
- [13] J. Vizcarrondo, J. Aguilar, E. Exposito y A. Subias, «ARMISCOM: Autonomic Reflective Middleware for Management Service Composition,» *Proceedings of the IEEE Global Information Infrastructure and Networking Symposium*, pp. 1-8, 2012.
- [14] J. Aguilar, M. Jerez, E. Exposito y T. Villemur, «CARMiCLOC: Context Awareness Middleware in CLOud Computing,» *Proceedings of the Latin American Computing Conference*, pp. 9-23, 2015.
- [15] X. Hesselbach y A. Galal, «Nanonetworks Communication Architecture: Modeling and Functions,» *Nano Communication Networks*, vol. 1, pp. 1-21, 2018.
- [16] H. Chong, I. Akyildiz y W. Gerstacker, «Timing Adquisition for Pulse-based Wireless Systems in the Terahertz Band,» *Proceedings of the 2nd Annual International Conference on Nanoscale Computing and Communication*, vol. 3, pp. 1-6, 2015.
- [17] N. Akkari, J. Jornet, P. Wang, E. Fadel, L. Elrefaei, M. Malik, S. Almasri y S. Akyildiz, «Joint Physical and Link Layer Error Control Analysis for Nanonetworks in the Terahertz Band,» *Wireless Networks*, vol. 22, pp. 1221-1233, 2016.
- [18] Q. Abbasi, K. Yang, N. Chopra, J. Jornet, N. Abuali, K. Qaraque y A. Alomainy, «Nanocommunication for Biomedical Applications: A Review of the State of the Art from Physical Layers to Novel Networking Concepts,» *IEEE Access*, vol. 4, pp. 3920-3935, 2016.
- [19] J. kokkonien, J. Lehtomaki, K. Umebayashi y M. Juntti, «Frequency and Time Domain Channel Models for nanonetworks in Terahertz Band,» *IEEE Transactions on Antennas and Propagation*, vol. 3, pp. 1-14, 2015.
- [20] F. Granelli, C. Costa y R. Bassolli, *Multiple Access Control Strategies for Nanoscale Communications and Networking*, Roma, 2019.
- [21] S. Karcher, E. Willighagen, J. Rumble, F. Ehrhart, C. Evelo, M. Fritts, S. Gaheen, S. Harper, M. Hoover, N. Jeliaskova, N. Lewinski, R. Robinson, K. Mills, A. Mustad, D. Thomas, G. Tsiliki y C. Hendren, «Integration Among databases and Datasets to

- Support Productive Nanotechnology: Challenges and Recommendations,» *NanoImpact*, vol. 9, pp. 85-101, 2018.
- [22] B. Bunakov y B. Matthews, «Metadata for Nanotechnology: Interoperability Aspects,» *Communication in Computer and Information Science. 11th International Conference, MTSR 2017*, vol. 75, pp. 247-252, 2017.
 - [23] V. Bunakov, T. Griffin, B. Matthews y S. Cozzini, «Metadata for Experiments in Nanoscience Foundries,» *International Conference on Data Analytics and Management in Data Intensive Domains. Data Analytics and Management in Data Intensive Domains*, vol. 706, pp. 248-262, 2017.
 - [24] M. Kornilina, V. Podryga, S. Polyakov, D. Puzyrkov y M. Yakovoskyi, «Cloud Service for Solution of Promising Problems of Nanotechnology,» *Supercomputing Frontiers and Innovation*, vol. 4, pp. 257-271, 2017.
 - [25] C. Liaskos y A. Tsioliaridou, «A promise of Realizable, Ultra-Scalable Communications at NanoScale: A MultiModal NanoMachine Architecture,» *IEEE Transactions on Computers*, vol. 64, pp. 1282-1295, 2015.
 - [26] I. Akyildiz, F. Brunneti y C. Blazquez, «Nanonetworks: A New Communication Paradigm,» *Computer networks Journal*, vol. 52, pp. 2260-2279, 2008.
 - [27] I. Akyildiz y J. Jornet, «The Internet of Nanothings,» *IEEE Wireless Communications*, vol. 1, pp. 58-63, 2010.
 - [28] P. Wang, J. Jornet, M. Malik, N. Akkari y I. Akyildiz, «Energy and Spectrum Aware MAC Protocol for perpetual Wireless Nanosensor Networks in the Terahertz Band,» *Ad Hoc networks*, vol. 8, pp. 2541-2555, 2013.
 - [29] D. Cook, J. Augusto y V. Jakkula, «Ambient Intelligence: Technologies, Applications and Opportunities,» *Pervasive and Mobile Computing*, vol. 5, pp. 277-298, 2009.
 - [30] I. Group, Scenarios for Ambient Intelligence in 2010, 2001.
 - [31] J. Rech y K. Althoff, «Artificial Intelligence and Software Engineering. Status and Future Trends,» *Themenschwerpunkt KI and SE*, vol. 3, pp. 5-11, 2004.
 - [32] F. Sadri, «Ambient Intelligence: A Survey,» *ACM Computing Surveys*, vol. 43, pp. 1-66, 2011.
 - [33] C. B. O. Bettini, K. Henriksen, J. Indulska, D. Nicklas, A. Ranganathan y D. Riboni, «A survey of Context Modelling and Reasoning Techniques,» *Pervasive and Mobile Computing*, vol. 6, pp. 161-180, 2009.

- [34] T. Patkos y D. Plexousakis, «Epistemic Reasoning for Ambient Intelligence,» *ICAART 2011 - Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* , vol. 1, pp. 242-248, 2011.
- [35] D. Ben-Zvi, «The Challenge of Developing Statistical, Literacy, Reasoning and Thinking,» de *Reasoning About Data Analysis*, 2004, pp. 121-145.
- [36] G. Riva, «Ambient Intelligence in health Care,» *Cyberpsychology, Behavior and Solcial Networking*, vol. 1, pp. 295-300, 2003.
- [37] G. a. M. D. Alfano, «On information transmission about nanomachines,» *1st International Conference on Nanonetworks and Workshops*, pp. 1-5, 2006.
- [38] M. Goebel y L. Gruenwald, «A Survey of Data Mining and Knowledge Discovery Software Tools,» *ACM SIGKDD Explorations Newsletter*, vol. 1, pp. 20-33, 1999.
- [39] j. Jung, C.-C. Liu y M. Hong, «Multiple Hypotheses and Their Credibility in On Line Fault Diagnosis,» *IEEE Transactions on Power Delivery*, vol. 2, pp. 225-230, 2001.
- [40] P. Balaji, G. Sachdeva, D. Srinivasan y C. Tham, «Multiagente Based System Based Urban Traffic Management,» *Proceedings of IEEE Congress on Evolutionary Computation* , pp. 1740-1747, 2007.
- [41] F. Bellifemine, G. Caire y D. Greenwood, *Developing Multi-agent Systems with JADE*, England: John Wiley: Chichester, 2007.
- [42] J. Anitha, K. Raja y C. Selvaraju, «Autonomic Computing,» *International Journal of Research and Reviews in Software Engineering*, vol. 1, pp. 1-15, 2011.
- [43] M. Parashar y S. Hariri, «Autonomic Computing: An Overview,» *Computatins in Space and Space in Computations*, vol. 3, pp. 257-269, 2004.
- [44] M. Huebscher y J. McCann, «A survey of Autonomic Computing. Degrees, Models and Applications,» *ACM Computing Services* , vol. 40, pp. 1-31, 2008.
- [45] A. Lopez-Pacheco y J. Aguilar, «Autonomic Reflective Middleware for the Management of NANOdevices in a Smart Environment (ARMNANO),» *Current Analysis on Communication Engineering*, vol. 2, pp. 56-63, 2019.
- [46] M. Zamani y M. Tahoori, «Reliable Design for Crossbar Nano-architectures,» *Modeling and Optimization in Science and Technologies* , vol. 9, pp. 421-444, 2017.
- [47] J. Aguilar, P. Valdiviezo, J. Cordero y M. Sanchez, «Conceptual Design of a Smart Classroom Based on Multiagent Systems,» *Proceeding of the International Conference on Artificial Intelligence*, pp. 471-477, 2015.

- [48] M. Sanchez, J. Aguilar, J. Cordero y J. Valdiviezo, «Basic features of a Reflective Middleware for Intelligent Learning Environment in the Cloud (IECL),» *Proceeding of the Asia-Pacific Conference on Computer Aided System Engineering (APCASE)*, pp. 5-17, 2015.
- [49] M. Mendoca, J. Aguilar y N. Perozo, «MIR-EO: Middleware Reflexivo para la Emergencia Ontologica en Ambientes Inteligentes,» *Latin American Journal of Computing*, vol. 3, pp. 16-21, 2016.
- [50] J. Aguilar, J. Cordero, L. barba, M. Sanchez, P. Valdiviezo y L. Chamba, «Learning Analytics Tasks as Services in Smart Classroom,» *Universal Access in the Information Society Journal*, vol. 17, pp. 693-709, 2018.
- [51] J. Aguilar, J. Cordero y O. Buendia, «Specification of the Autonomic Cycles of Learning Analytic Tasks for a Smart Classroom,» *Journal of Educational Computing Research*, vol. 56, pp. 866-891, 2017.
- [52] A. Lopez Pacheco y J. Aguilar Castro, «NANO-Communication Management System for Smart Environments,» *Revista Venezolana de Computacion*, vol. 5, pp. 12-22, 2018.
- [53] J. Vizcarrondo, J. Aguilar, E. Exposito y A. Subias, «ARMISCOM: Autonomic reflective middleware for management service composition,» *Proc. 4th IEEE Global Information Infrastructure and Networking Symposium, IEEE Communication Society*, vol. 1, p. 1, 2012.
- [54] J. Aguilar, «Definition of an Energy Function for the Random Neural to Solve optimization Problems,» *Neural Networks*, vol. 11, pp. 731-737, 1998.
- [55] I.S., *EEE Recommended Practice for Nanoscale and Molecular Communication Framework*, New York: IEEE Communications Society, 2015.
- [56] A. Russell, «Rough Consensus and Running Code' and the Internet-OSI Standards War,» *IEEE Annals of the History of Computing*, vol. 28, pp. 48-61, 2006.
- [57] M. Scorza, A. Elce, F. Zarrilli, R. Liguori, F. Amato y G. Castaldo, «Genetic Diseases That Predispose to Early Liver Cirrhosis,» *International Journal of Hepatology*, p. 713754, 2014.
- [58] A. Moon, A. Singal y E. Tapper, «Contemporary Epidemiology of Chronic Liver Disease and Cirrhosis,» *Clinical Gastroenterology and Hepatology*, vol. 18, pp. 2650-2666, 2020.
- [59] S. Asrani, H. Devarbhavi, J. Eaton y P. Kamath, «Burden of liver diseases in the world,» *Journal of Hepatology*, vol. 70, pp. 151-171, 2019.

