

Universidad  
Rey Juan Carlos

Máster Universitario en Ingeniería de Sistemas de  
Información  
2023-2024

*Trabajo de Fin de Máster*

“Contexto Semántico de Red para  
Robots Móviles”

---

Alejandro Calvillo Fernandez

Juan Manuel Vara Mesa  
Leganés, 10 de Junio 2024



## RESUMEN

Este trabajo presenta un modelo de orquestación semántica y políticas de descarga para redes en robots móviles con el fin para capturar las interacciones del sistema robótico. La investigación se centra en cómo integrar el contexto semántico en la gestión de recursos y la toma de decisiones, permitiendo una adaptación dinámica a las condiciones cambiantes del entorno. La validación experimental se realiza en una aplicación de vigilancia de seguridad, evaluando funciones como la detección de objetos bajo diferentes contextos operativos y factores de compresión.

Los resultados muestran que la consideración del contexto mejora significativamente la eficiencia operativa, el consumo de energía y la calidad del servicio. Este enfoque permite a los robots adaptar dinámicamente su comportamiento y la asignación de recursos, facilitando una interacción más inteligente y colaborativa entre los diferentes componentes del sistema.

**Palabras clave:** Orquestación semántica, Robots Móviles, Gestión de Recursos, Contexto Semántico, Compresión de Datos.

## ABSTRACT

This framework presents a model for semantic orchestration and offloading policies for networks in mobile robots to capture the interactions within the robotic system. The research focuses on integrating semantic context into resource management and decision-making, enabling dynamic adaptation to changing environmental conditions. Experimental validation is conducted by a security surveillance use case, evaluating functions such as object detection under different operational contexts and compression factors.

The results show that considering context significantly improves operational efficiency, energy consumption, and quality of service. This approach allows robots to dynamically adapt their behavior and resource allocation, facilitating smarter and more collaborative interaction between the various system components

**Keywords:** Semantic orchestration, Mobile Robots, Resource Management, Semantic Context, Data Compression.

# ÍNDICE GENERAL

1. INTRODUCCIÓN. . . . .	1
1.1. Motivación . . . . .	1
1.1.1. Contexto Semántico . . . . .	1
1.1.2. Orquestación Semántica . . . . .	2
1.1.3. Aplicaciones en la Industria 4.0 y otras motivaciones . . . . .	3
1.2. Plan de Proyecto . . . . .	4
1.3. Objetivo del Trabajo . . . . .	4
1.4. Estructura de la Memoria . . . . .	5
2. ESTADO DEL ARTE. . . . .	6
2.1. Sistemas sensibles al Contexto en Robótica . . . . .	6
2.1.1. Definición y Aplicaciones . . . . .	6
2.1.2. Frameworks y Modelos . . . . .	7
2.1.3. Calidad del Contexto (QoC). . . . .	8
2.1.4. Evaluación Distribuida del Contexto . . . . .	8
2.2. Aplicaciones Industriales y Energía Eficiente . . . . .	9
2.2.1. Evaluación del Entorno . . . . .	9
2.2.2. Robótica en la Industria 4.0 . . . . .	9
2.3. Enfoques en el Diseño de Sistemas Robóticos. . . . .	10
2.3.1. Diseño Basado en el Contexto . . . . .	10
2.3.2. Frameworks Semánticos. . . . .	11
3. SISTEMAS ROBÓTICOS . . . . .	12
3.1. ¿Qué es un Robot? . . . . .	12
3.2. ¿Qué es un Sistema Robótico? . . . . .	13
3.3. Capacidades de los componentes del sistema . . . . .	14
4. CONTEXTO SEMÁNTICO . . . . .	16
4.1. ¿Qué se entiende por Contexto Semántico? . . . . .	16
4.2. Tipos de Contextos. . . . .	18
4.2.1. Contexto Físico. . . . .	18

4.2.2. Contexto Operacional . . . . .	19
4.2.3. Contexto de Red . . . . .	20
4.2.4. Contexto de la Tarea . . . . .	20
4.2.5. Contexto Social. . . . .	20
4.2.6. Contexto Semántico Combinado . . . . .	21
4.3. Uso de Contextos para la Orquestación de Redes y Servicios . . . . .	21
5. IMPLEMENTACIÓN DEL MODELO . . . . .	24
5.1. Modelo del Sistema . . . . .	24
5.1.1. Aplicación Robótica . . . . .	25
5.1.2. Función Robótica . . . . .	25
5.1.3. Configuración Robótica . . . . .	25
5.1.4. Factor de Compresión . . . . .	25
5.1.5. Recursos de Radio y Cómputo . . . . .	26
5.1.6. Políticas de Despliegue . . . . .	26
5.1.7. Consumo de Batería . . . . .	26
5.1.8. Calidad de la Aplicación. . . . .	27
5.1.9. Selección de Configuración de Funciones . . . . .	27
5.1.10. Selección de Políticas de Despliegue . . . . .	28
5.1.11. Satisfacción de los Requisitos de la Aplicación . . . . .	28
5.1.12. Mantener el Consumo de Energía Limitado . . . . .	28
5.1.13. Limitar la Utilización de Recursos al Presupuesto Total . . . . .	28
5.1.14. Función Objetivo a resolver . . . . .	29
5.1.15. Arquitectura Física . . . . .	30
5.2. Definición del test-bed de la Experimentación . . . . .	30
5.2.1. Política 1: Descarga completa en la nube . . . . .	30
5.2.2. Política 2: Computación local en el robot . . . . .	31
6. EXPERIMENTACIÓN Y RESULTADOS . . . . .	32
6.1. Configuración del Hardware y el Software . . . . .	32
6.1.1. Hardware . . . . .	32
6.1.2. Software . . . . .	33

6.2. Modelos de Detección de Objetos . . . . .	35
6.2.1. ¿Qué son YOLOv3 y YOLOv3-tiny? . . . . .	35
6.3. Análisis del Factor de Compresión.. . . . .	37
6.3.1. YOLOv3: Descripción de las Imágenes con diferentes factores de compresión. . . . .	39
6.3.2. YOLOv3-tiny: Descripción de las Imágenes con diferentes factores de compresión. . . . .	44
6.4. Análisis de Gráficos de detecciones . . . . .	50
6.5. Análisis de los Contextos. . . . .	51
6.5.1. Contexto de Laboratorio (Laboratory) . . . . .	52
6.5.2. Contexto de Pasillo (Corridor) . . . . .	52
7. IMPACTO SOCIOECONÓMICO DEL PROYECTO . . . . .	55
7.1. Análisis Sociológico del Proyecto . . . . .	55
7.1.1. Interacción Humana y Robots. . . . .	55
7.1.2. Vida Cotidiana . . . . .	56
7.1.3. Seguridad . . . . .	57
7.1.4. Empleo . . . . .	58
7.2. Análisis Organizativo y Económico del Proyecto . . . . .	59
7.2.1. Organización Secuencial Temporal del Proyecto . . . . .	59
7.2.2. Costes Asociados al Proyecto . . . . .	63
7.2.3. Análisis Económico . . . . .	64
8. CONCLUSIONES Y TRABAJO FUTURO . . . . .	66
8.1. Desafíos y Futuras Direcciones . . . . .	66
BIBLIOGRAFÍA . . . . .	68

# 1. INTRODUCCIÓN

Este capítulo se estructura en varias secciones para proporcionar un marco completo del proyecto. Comenzamos con una motivación que destaca la importancia de la integración de robots móviles en sistemas de comunicación avanzados. En esta sección nos encontraremos con una explicación detallada del contexto semántico, donde trataremos de explicar su papel en la optimización del uso de la red y el rendimiento de los robots móviles. A continuación, se presenta el plan del proyecto, delineando los pasos y métodos utilizados en este estudio. Finalmente, se especifican los objetivos del trabajo, que guiarán la investigación y la validación experimental del modelo propuesto.

## 1.1. Motivación

La integración de robots móviles en sistemas de comunicación y redes se ha incrementado notablemente en los últimos años, impulsada por avances en las tecnologías de redes inalámbricas Beyond-5G. Diferentes proyectos de innovación europeos ya cuentan entre sus casos de uso la presencia de servicios que incluyen robots móviles [1][2]. Desde la vigilancia y la seguridad hasta la entrega de paquetes y la asistencia personal, los robots móviles desempeñan un papel importante en diversos sectores de la industria 4.0 [3]. Sin embargo, para que estos sistemas operen de manera eficiente y segura, es esencial considerar el contexto en el que se despliegan. Este trabajo se enfoca en la validación experimental de la consideración de diferentes contextos semánticos, con el objetivo de optimizar el uso de la red y mejorar el rendimiento general de los robots móviles.

### 1.1.1. Contexto Semántico

El contexto semántico hace referencia a la información adicional que enriquece la interpretación de los datos recogidos, proporcionando una comprensión más profunda del entorno operativo del robot. Esta información puede incluir datos sobre la ubicación del robot, las condiciones ambientales, el estado interno del robot y las características de la red [4]. La diversidad de los entornos en los que operan requieren enfoques avanzados para la gestión y la optimización de recursos. En este contexto, la orquestación semántica de redes y servicios se presenta como una solución prometedora para mejorar la eficiencia operativa y la calidad del servicio en sistemas robóticos [5].

Aplicar dicho contexto a nuestras redes, no solo permite una interpretación más rica y precisa de los datos recogidos por los sensores del robot, sino que también facilita una serie de capacidades avanzadas para la operación efectiva en entornos dinámicos [6]. Estas capacidades incluyen la desambiguación de datos, la inferencia de nuevas relaciones, el razonamiento contextual, y la interacción natural, todas ellas críticas para la toma de

decisiones informadas y la planificación de acciones en tiempo real.

### 1.1.2. Orquestación Semántica

Teniendo claro qué se entiende como contexto semántico, podemos explicar qué es la orquestación semántica. La orquestación semántica implica el uso de información contextual adicional para tomar decisiones informadas sobre la asignación de recursos y la priorización de tareas, lo que permite una adaptación dinámica a las condiciones cambiantes del entorno y de la red [7].

Este concepto representa un avance significativo en la gestión de redes para sistemas robóticos. Al integrar información contextual en la toma de decisiones, se pueden desarrollar políticas de descarga más efectivas que determinen dónde y cómo se procesan las tareas robóticas. Estas políticas pueden variar desde la ejecución completa en el servidor de borde hasta la computación local en el robot, dependiendo de las condiciones de la red y del contexto operativo [8]. Este enfoque no solo optimiza el rendimiento del robot y la red, sino que también mejora la capacidad de los sistemas robóticos para adaptarse a entornos complejos [9].

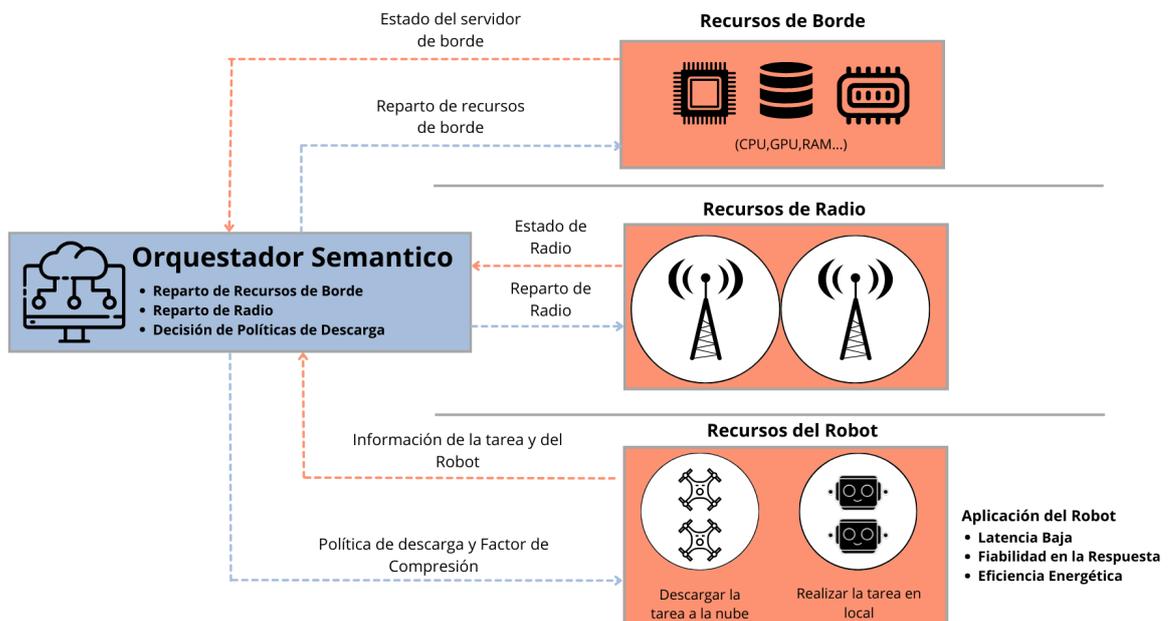


Fig. 1.1. Ejemplo de un orquestador semántico.

En la figura 1.1 podemos encontrar un ejemplo de un diseño esquemático de un orquestador semántico. En este sistema, el orquestador se encarga de decidir qué recursos radio y de computación de borde se asignan a qué tarea. Por otro lado, en el plano del robot, se encarga de decidir el factor de compresión del flujo de datos que usará para tomar las decisiones de orquestación y la política de descarga a seguir, manteniendo los

requerimientos que tiene la aplicación del robot.

La diversidad de entornos y las complejidades inherentes a las operaciones robóticas plantean desafíos significativos en la gestión y optimización de recursos. Este enfoque permite una adaptación dinámica a las condiciones cambiantes del entorno y de la red, mejorando así la eficiencia operativa y la calidad del servicio.

### **1.1.3. Aplicaciones en la Industria 4.0 y otras motivaciones**

En la Industria 4.0, los robots móviles desempeñan diversos roles en una variedad de sectores. En aplicaciones de vigilancia y seguridad, los robots utilizan tecnologías avanzadas de visión y detección para identificar amenazas potenciales y responder de manera autónoma a eventos inusuales [10]. En la logística y la entrega de paquetes, los robots optimizan las rutas de entrega y gestionan dinámicamente las tareas en función de las condiciones del entorno y la red [11]. En la asistencia personal, los robots mejoran la calidad de vida de las personas al proporcionar servicios adaptados a sus necesidades específicas.

Este trabajo se centra en la aplicación de vigilancia de seguridad, evaluando funciones como la visión por computadora bajo diferentes contextos operativos y factores de compresión. Utilizando datos reales y en tiempo real, se proporciona una caracterización detallada del comportamiento y rendimiento de las funciones robóticas, lo que permite obtener conclusiones aplicables a escenarios prácticos.

De forma adicional, en la memoria se destaca la importancia de la robustez y la fiabilidad en la comunicación entre los robots y el servidor de borde. La estabilidad de la red y la calidad de servicio (QoS) son factores críticos para el desempeño del sistema, especialmente en aplicaciones donde la precisión y la rapidez en la toma de decisiones son esenciales [12]. Al integrar la información contextual en la gestión de la red, es posible priorizar el tráfico de datos crítico y ajustar las rutas de datos para minimizar la latencia, mejorando así la eficiencia operativa y la capacidad de respuesta del sistema.

El uso de datos en tiempo real garantiza que las decisiones tomadas por el sistema sean precisas en el contexto actual. Tanto la recopilación, como el procesamiento de datos en tiempo real, permiten al sistema adaptarse dinámicamente a las condiciones cambiantes del entorno y de la red, optimizando el uso de los recursos y mejorando la calidad del servicio. En este trabajo, se ha empleado una combinación de sensores y tecnologías de comunicación avanzadas para capturar y procesar datos en tiempo real, lo que permite una evaluación precisa y detallada de las políticas de orquestación y descarga propuestas.

Por otro lado, este trabajo aborda la colaboración entre robots y otros dispositivos de red para mejorar la eficiencia operativa. Al compartir información contextual, los robots pueden coordinarse mejor entre sí y con otros componentes del sistema, lo que reduce el riesgo de colisiones y optimiza el flujo de trabajo.

## 1.2. Plan de Proyecto

El plan de este proyecto se basa en varios componentes clave:

- **Revisión del Estado del Arte:** Analizar los trabajos previos relacionados con sistemas robóticos y la gestión contextual.
- **Desarrollo del Modelo:** Proponer un modelo de colaboración robot-sistema que integre el contexto semántico.
- **Validación Experimental:** Diseñar un test-bed para validar el modelo propuesto en escenarios prácticos.
- **Evaluación y Análisis de Resultados:** Realizar experimentos para evaluar el impacto de los contextos semánticos en el rendimiento de los robots móviles y optimizar el modelo basado en los resultados obtenidos.

## 1.3. Objetivo del Trabajo

En este trabajo, se tiene como objetivo dos puntos clave con el fin de optimizar el uso de la red en sistemas robóticos:

- Proponer un modelo de colaboración robot-sistema, que tenga en cuenta el contexto en el que se encuentra el robot.
- Validar experimentalmente los fundamentos del modelo. Esto incluye el uso de diferentes umbrales de compresión de los datos recogidos por el robot, y el análisis del impacto de dos contextos diferentes.

En el trabajo, presentamos un modelo de orquestación semántica y políticas de descarga en redes para robots móviles. El modelo propuesto trata representar las interacciones complejas entre los diferentes componentes del sistema robótico. Esta representación permite una gestión eficiente de los recursos y una toma de decisiones informada, considerando tanto el contexto físico como el contexto de red y el contexto operacional del robot. Destacamos como innovación clave la definición de políticas de descarga, que determinan dónde y cómo se procesan las tareas robóticas. Estas políticas pueden variar desde la ejecución completa en el servidor de borde hasta la computación local en el robot, dependiendo de las condiciones de la red y del contexto operativo.

La validación experimental se centra en una aplicación de vigilancia de seguridad, que incluye funciones como la navegación autónoma, la identificación de personas y el control del robot. Estas funciones se han evaluado bajo diferentes contextos operativos y factores de compresión, proporcionando una caracterización detallada de su comportamiento y rendimiento. La utilización de modelos de detección de objetos avanzados, como Yolox\_s

y YoloX\_tiny, permite analizar cómo diferentes niveles de compresión afectan la precisión y la eficiencia de las tareas de visión por computadora.

Dicha validación se ha llevado a cabo utilizando datos reales y en tiempo real, lo que permite obtener conclusiones aplicables a escenarios prácticos. Nuestra metodología incluye la implementación de un test-bed experimental que evalúa la calidad y el consumo de energía de funciones robóticas bajo diferentes configuraciones y políticas de descarga, utilizando un robot Turtlebot equipado con sensores y un servidor de borde con capacidades de procesamiento de alto rendimiento. Este enfoque permite una caracterización detallada del comportamiento del sistema en diversos contextos operativos, proporcionando una perspectiva valiosa para la optimización del rendimiento del robot y la red.

La integración de las políticas de orquestación y descarga en el modelo se ha realizado mediante un conjunto de métricas que incluyen la precisión de la detección de objetos, el consumo de energía y la calidad del servicio. Estas métricas permiten evaluar de manera objetiva el impacto de las políticas propuestas y proporcionar recomendaciones para su optimización. Los resultados experimentales muestran que es posible lograr un equilibrio entre la eficiencia del sistema y la calidad del servicio mediante la consideración adecuada del contexto semántico y la adaptación dinámica de las políticas de descarga.

#### **1.4. Estructura de la Memoria**

Este trabajo se estructura en varios capítulos. El capítulo 1 presenta el objetivo del estudio. En el capítulo 2, revisamos los trabajos previos y relacionados con el mismo. El siguiente capítulo 3 define los conceptos fundamentales de la arquitectura de un robot y su participación en un sistema telemático. En el capítulo 5, se encuentra una definición formal del modelo. El capítulo 6 presenta la test-bed experimental y los resultados obtenidos, validando las hipótesis propuestas. En el capítulo 7, encontramos un desglose en términos de organización y costes del proyecto. Finalmente, la conclusión y el trabajo futuro, que se puede encontrar en el capítulo 8, resume los hallazgos y sugiere direcciones para futuras investigaciones.

## 2. ESTADO DEL ARTE

La investigación presentada en este trabajo destaca la relevancia de integrar contextos semánticos en la orquestación y gestión de redes para sistemas robóticos. La validación experimental mediante datos reales y en tiempo real demuestra que considerar la información contextual permite mejorar significativamente la eficiencia operativa, el consumo de energía y la calidad del servicio. La capacidad de los robots para adaptar dinámicamente su comportamiento y la asignación de recursos en función de las condiciones cambiantes del entorno y de la red se traduce en sistemas más robustos y efectivos. Este enfoque no solo optimiza el rendimiento de los robots, sino que también facilita una interacción más inteligente y colaborativa entre los diferentes componentes del sistema, abriendo nuevas vías para el desarrollo de aplicaciones robóticas.

La integración de la sensibilidad contextual en sistemas robóticos ha sido un tema de creciente interés en la investigación y el desarrollo de tecnologías avanzadas. La capacidad de los robots para interpretar y responder a su entorno de manera adecuada es crucial para mejorar su eficiencia y funcionalidad en diversas aplicaciones, desde la vigilancia hasta la asistencia personal y la manufactura industrial. En este capítulo, exploraremos otros modelos presentes en el estado del arte, al igual que trataremos de exponer diversas definiciones importantes para la comprensión de esta memoria.

### 2.1. Sistemas sensibles al Contexto en Robótica

En esta sección, exploraremos los sistemas sensibles al contexto en el ámbito de la robótica presentes en el estado del arte. Analizaremos cómo estos sistemas permiten a los robots percibir los cambios en su entorno, mejorando su interacción con el entorno y su eficiencia operativa en diversas aplicaciones. Este conocimiento sentará las bases para entender la importancia de la sensibilidad contextual en la robótica.

#### 2.1.1. Definición y Aplicaciones

Los sistemas sensibles al contexto son aquellos que pueden percibir y reaccionar a los cambios en su entorno. Estos sistemas utilizan datos de sensores para inferir el estado del entorno y tomar decisiones informadas sobre las acciones a realizar [13]. La implementación de estos sistemas en robótica ha permitido el desarrollo de aplicaciones avanzadas, como robots de servicios, que pueden adaptar su comportamiento en función de la situación específica [14].

En la robótica de servicios, los sistemas sensibles al contexto se utilizan para mejorar la interacción del robot con su entorno y con los humanos. Por ejemplo, los robots de vigilancia pueden utilizar tecnologías de visión para detectar y responder a eventos inusuales

en tiempo real [15]. Otro ejemplo es el uso de sistemas móviles contextuales que pueden gestionar servicios y realizar tareas en nombre del usuario, como la reconfiguración automática de un robot en función del ruido y la luminosidad del entorno [16].

Los sistemas robóticos industriales también se benefician de la sensibilidad contextual, ya que pueden adaptarse dinámicamente a las condiciones cambiantes del entorno de trabajo. Esto es especialmente importante en entornos de ensamblaje industrial, donde los modelos cognitivos y de computación ubicua permiten el control preciso de los comportamientos del robot [17]. Además, los sistemas basados en middleware (software con el que diferentes aplicaciones se comunican entre sí) sensibles al contexto proporcionan una plataforma para integrar y gestionar datos contextuales, mejorando la capacidad de los robots para aprender y adaptarse a nuevas tareas [18]. CAMUS (Context-Aware Middleware for URC Systems) es un ejemplo de un middleware desarrollado específicamente para sistemas robóticos inteligentes basados en red. Este middleware ayuda a los robots a ser conscientes del contexto del usuario y proporcionar servicios proactivos y personalizados. La arquitectura de CAMUS permite la integración y gestión de datos contextuales de manera eficiente, facilitando la adaptación dinámica del comportamiento del robot en función del contexto [14].

### 2.1.2. Frameworks y Modelos

Diversos frameworks y modelos han sido propuestos para integrar la sensibilidad contextual en los sistemas robóticos. Un enfoque común es el uso de modelos cognitivos y de inferencia basados en ontologías para gestionar y transformar información contextual de alto nivel en datos utilizables por sistemas técnicos.

La implementación de frameworks semánticos, como SmartRules, facilita el desarrollo de aplicaciones de Internet de las Cosas Robóticas (IoRT) sensibles al contexto. Estos frameworks permiten la creación de reglas de gestión del contexto y la integración de dispositivos físicos y virtuales, lo que resulta en una mayor adaptabilidad y eficiencia de los sistemas robóticos [19]. SmartRules, por ejemplo, utiliza un lenguaje de reglas de producción para la inferencia reactiva basada en información contextual representada en un lenguaje ontológico dedicado, llamado  $\mu$ -Concept.

Por otro lado, la evaluación continua de la calidad del contexto es fundamental para garantizar que las decisiones de los sistemas robóticos se basen en datos fiables. Esto implica la implementación de métricas y modelos que permitan evaluar constantemente la precisión de los datos contextuales, adaptando así el comportamiento del robot según sea necesario. Los frameworks que incluyen estas capacidades pueden mejorar significativamente la adaptabilidad de los sistemas robóticos en entornos dinámicos [20].

En entornos colaborativos, donde múltiples robots interactúan y comparten información, los modelos de inferencia distribuida son esenciales. Estos modelos permiten que cada robot realice inferencias basadas en datos compartidos y contextuales, mejorando la

coordinación y eficiencia en la realización de tareas complejas. La integración de enfoques de inferencia distribuida con modelos de conocimiento compartido puede facilitar la cooperación entre robots en entornos industriales y de servicios [21].

Los modelos formales y las técnicas de simulación son utilizados para validar y verificar el comportamiento de los sistemas sensibles al contexto. Estos modelos permiten realizar pruebas exhaustivas y análisis de escenarios antes de la implementación en el mundo real, asegurando que el sistema funcione según lo esperado bajo diversas condiciones. Por ejemplo, el uso de redes de control booleanas para verificar la estabilidad y otras propiedades útiles en sistemas de alerta ante avalanchas y deslizamientos de tierra ha demostrado ser efectivo para implementar sistemas de alarma precisos y oportunos [22].

La computación en la nube proporciona una solución eficiente para manejar grandes volúmenes de datos y realizar cálculos intensivos necesarios para la sensibilidad contextual. Los frameworks basados en la nube permiten que los robots descarguen tareas de procesamiento y almacenamiento, mejorando así su capacidad para analizar y responder a datos contextuales en tiempo real. Este enfoque es especialmente útil en aplicaciones de educación móvil, donde los robots pueden optimizar sus respuestas basadas en el contexto del usuario [23].

### **2.1.3. Calidad del Contexto (QoC)**

La calidad de la información contextual es un factor crítico para el rendimiento de los sistemas sensibles al contexto. Los datos contextuales pueden ser imprecisos o incompletos debido a las limitaciones técnicas de los sensores, su disponibilidad, disfunción y la naturaleza altamente dinámica del entorno. Para abordar estas imperfecciones, se utiliza el concepto de Calidad del Contexto (QoC), que mide la calidad de cualquier información utilizada como contexto [20]. La QoC incluye indicadores como la precisión, la oportunidad, la completitud y la relevancia de los datos contextuales [24].

Además, la QoC se utiliza para garantizar que las decisiones de adaptación se basen en datos contextuales de alta calidad. Esto es esencial en aplicaciones críticas donde la precisión y la confiabilidad son fundamentales para la seguridad y el rendimiento [25]. Por ejemplo, en sistemas de salud inteligentes, la QoC puede mejorar significativamente la eficiencia de los sistemas al proporcionar métricas de calidad junto con la información contextual [26].

### **2.1.4. Evaluación Distribuida del Contexto**

La evaluación distribuida del contexto permite que múltiples robots en un entorno inteligente colaboren y compartan información contextual. Este enfoque mejora la adaptabilidad y la robustez de los sistemas robóticos en entornos dinámicos al permitir que los robots intercambien datos y realicen inferencias contextuales basadas en un modelo

de conocimiento compartido [27]. Este tipo de sistemas se beneficia de la capacidad de realizar evaluaciones contextuales en tiempo real para aplicaciones donde las condiciones pueden cambiar rápidamente, como en la robótica de asistencia en hospitales [28].

Los sistemas de evaluación distribuida del contexto también emplean técnicas avanzadas de razonamiento contextual para manejar la complejidad y el volumen de datos contextuales. Al distribuir el proceso de razonamiento, estos sistemas pueden escalar y mantener un alto rendimiento en escenarios a gran escala [29]. La evaluación distribuida es un campo a estudiar para aplicaciones que requieren la cooperación de múltiples robots, permitiendo una toma de decisiones coordinada [30].

## **2.2. Aplicaciones Industriales y Energía Eficiente**

En esta sección, se discutirá cómo los sistemas robóticos sensibles al contexto se aplican en entornos industriales y cómo contribuyen a una mayor eficiencia energética. Se explorarán las ventajas de estos sistemas en mejorar la operatividad y seguridad en la industria.

### **2.2.1. Evaluación del Entorno**

Los sistemas robóticos sensibles al contexto son particularmente útiles en entornos industriales, donde pueden evaluar y adaptarse a las condiciones cambiantes para mejorar la eficiencia operativa y la seguridad [31]. La capacidad de los robots para adaptarse a las condiciones cambiantes del entorno es crucial para mejorar la operatividad y la seguridad [32]. Los sistemas robóticos que incorporan evaluaciones de seguridad pueden reaccionar de manera proactiva para evitar colisiones con obstáculos potenciales, incluyendo seres humanos, mediante la implementación de estrategias de control orientadas a la seguridad.

En entornos inteligentes, como los hogares inteligentes, los robots se integran con sistemas de monitoreo del entorno para proporcionar apoyo interactivo a personas con limitaciones funcionales. Estos sistemas combinan la detección y mapeo de objetos con la interacción robótica para detectar y asistir en la corrección de errores durante actividades diarias [33]. Asimismo, en situaciones extremas, como las zonas de emergencia, los robots equipados con sistemas de monitoreo ambiental pueden evaluar la contaminación ecológica y su impacto en la salud humana, transmitiendo datos a sistemas de soporte de decisiones inteligentes [34].

### **2.2.2. Robótica en la Industria 4.0**

En el contexto de la Industria 4.0, la robótica industrial se beneficia enormemente de las tecnologías cognitivas y del Internet de las Cosas (IoT). Los sistemas robóticos que utilizan servicios en la nube sensibles al contexto pueden realizar tareas de manipulación

de materiales de manera más eficiente y con un menor consumo de energía [35]. Estos avances permiten la creación de fábricas inteligentes donde los robots juegan su papel en la mejora de la productividad, la calidad del producto y la eficiencia operativa [36], permitiendo la creación de entornos de trabajo más seguros mediante la colaboración entre humanos y robots [37].

Los robots industriales han evolucionado significativamente con el devenir de la Industria 4.0. Las fábricas inteligentes emplean robots que pueden realizar tareas complejas con una alta precisión y adaptarse rápidamente a cambios en los procesos de manufactura. Esta capacidad se debe en gran parte a la utilización de tecnologías como la computación en la nube, el Big Data y los sensores inteligentes, que permiten a los robots recopilar y analizar datos en tiempo real para optimizar sus operaciones [38]. Además, la implementación de robots colaborativos, o “Cobots”, está transformando la manera en que los humanos interactúan con las máquinas, haciendo posible trabajar “codo con codo” de manera segura y eficiente [39].

La migración de sistemas robóticos tradicionales hacia componentes de la Industria 4.0 implica la adopción de nuevos estándares y la integración de robots en redes inteligentes. Este enfoque permite a los robots comunicarse de manera autónoma, mejorando así la flexibilidad de las líneas de producción [40]. Los robots en la Industria 4.0 también se benefician de avances en la inteligencia artificial, lo que les permite aprender nuevas tareas sin necesidad de una programación formal, colaborar tanto con otros dispositivos autónomos, como con operadores humanos de manera más efectiva [41].

### **2.3. Enfoques en el Diseño de Sistemas Robóticos**

El diseño de sistemas robóticos basado en el contexto permite una mayor adaptabilidad y robustez en diversas condiciones operativas. En esta sección, se analizarán los enfoques actuales en el diseño de robots que utilizan conocimiento contextual para optimizar su rendimiento.

#### **2.3.1. Diseño Basado en el Contexto**

El diseño basado en el contexto en robótica implica la representación explícita del conocimiento contextual para mejorar la robustez y la capacidad de adaptación de los sistemas robóticos a diferentes condiciones operativas. Este enfoque permite que los robots personalicen sus funcionalidades según las características específicas de la situación en la que se encuentran. Por ejemplo, el uso de secuencias contextuales como estructura de control en robots autónomos permite una mayor flexibilidad y adaptabilidad al cambiar dinámicamente sus comportamientos en respuesta a cambios en el entorno [42]. Esta metodología trata de facilitar la implementación de pequeños componentes especializados en lugar de subsistemas complejos [43].

Además, la integración del conocimiento contextual en el diseño de sistemas robóticos permite una mejor gestión de la información en entornos dinámicos. Un marco de trabajo basado en el contexto puede proporcionar modelos para la adaptación en tiempo real de los sistemas robóticos, mejorando así su capacidad para enfrentar situaciones imprevistas. Por ejemplo, el uso de modelos probabilísticos y enfoques de búsqueda de políticas basadas en modelos para la generalización de habilidades robóticas demuestra que los robots pueden adaptarse eficientemente a nuevos contextos, reduciendo significativamente la cantidad de interacciones necesarias para el aprendizaje [44].

### **2.3.2. Frameworks Semánticos**

Los frameworks semánticos en robótica proporcionan una base estructurada para la representación y gestión del conocimiento contextual, facilitando la integración de dispositivos físicos y virtuales y mejorando la adaptabilidad y eficiencia de los sistemas robóticos. Un enfoque destacado es el uso de mapas semánticos, que combinan información métrica con propiedades de alto nivel del entorno, como tipos de lugares y ubicaciones de objetos. Estos mapas permiten a los robots interpretar y razonar sobre su entorno utilizando descripciones en lenguaje natural y observaciones de sensores de bajo nivel [45]. Este tipo de representación híbrida mejora tanto la precisión métrica como la capacidad de los robots para comprender y actuar en diferentes escenarios.

Otro ejemplo es el uso de ontologías y modelos formales para la navegación y la interacción humano-robot. Los frameworks semánticos basados en ontologías permiten a los robots razonar sobre las relaciones y propiedades de los objetos en su entorno, facilitando tareas complejas como la categorización de lugares y la planificación de rutas [5]. Estos sistemas pueden aprender nuevas categorías de objetos y lugares a través de interacciones con usuarios, mejorando su capacidad de adaptación en tiempo real [46].

La integración de la sensibilidad contextual en los sistemas robóticos promete mejorar significativamente su rendimiento y adaptabilidad en diversas aplicaciones. Los avances en middleware sensible al contexto, frameworks semánticos y modelos de evaluación distribuida del contexto son pasos cruciales hacia el desarrollo de robots más inteligentes y eficientes.

## 3. SISTEMAS ROBÓTICOS

En este capítulo, se profundizará en la definición y componentes esenciales de un sistema robótico. Primero, se clarificará qué es un robot, sus capacidades y componentes fundamentales. Luego, se abordará el concepto de sistema robótico, explorando la interacción entre sus elementos y su arquitectura. Con estas definiciones claras, podemos avanzar en el análisis de cómo los robots pueden integrarse en redes de comunicación y colaborar eficazmente con otros dispositivos.

### 3.1. ¿Qué es un Robot?

Antes de entender que es un sistema robótico, debemos definir qué es lo que se considera un robot. Esta definición es un debate abierto tanto en campos tecnológicos como en dominios jurídicos. Podemos tomar como definición inicial de robótica a escritor de ciencia ficción Isaac Asimov y sus tres leyes de la robótica. Acercándonos más a nuestro mundo, podemos definir un robot como “Máquina o dispositivo que efectúa ciertas labores de modo autónomo en sustitución de los humanos” [47], definición aportada por la literatura japonesa en referencia a la obra de Capek.

Con el fin de alejarnos del debate humano-máquina, vamos a definir el concepto *robot* como una “máquina programable capaz de realizar una serie de tareas de forma autónoma o semi-autónoma”. Esta definición captura la esencia de lo que consideramos un robot en el contexto moderno: una entidad que puede interactuar con su entorno, tomar decisiones basadas en sus sensores y ejecutar acciones sin intervención parcial, o totalmente humana constante

Un robot puede incluir una amplia variedad de componentes físicos y tecnológicos. Los elementos hardware que suelen componer un robot incluyen sensores (como cámaras, micrófonos, sensores de proximidad), actuadores (motores, servos), y sistemas de procesamiento (CPU, RAM). Además, un robot está equipado con software especializado que le permite procesar información, tomar decisiones y ejecutar tareas.

Desde una perspectiva telemática, un robot se puede ver como un usuario más dentro de una red. Sin embargo, a diferencia de un usuario humano, el robot posee capacidades de computación propias que le permiten realizar cálculos complejos, procesar datos en tiempo real y ejecutar algoritmos avanzados de aprendizaje automático. Estas capacidades lo diferencian de otros dispositivos conectados a la red por su habilidad para interactuar con un entorno físico, a lo que denominamos contexto. En futuros capítulos, desarrollaremos con mayor profundidad lo que entendemos por contexto.

Por tanto, podemos considerar al robot como un nodo dentro de una red de comunicación, similar a otros dispositivos conectados, pero con la capacidad adicional de computar

y procesar datos independientemente. En esencia, un robot es un usuario autónomo, que puede comunicarse con otros dispositivos, procesar datos, y actuar en consecuencia de manera autónoma.

Al tratar los robots desde este punto de vista, es posible resaltar su papel no solo como máquinas autónomas, sino también como participantes activos en redes de comunicación, capaces de contribuir a la infraestructura de sistemas más amplios. Esto incluye su capacidad de ser orquestados dentro de redes complejas, optimizando tanto su rendimiento como el de la red en la que están integrados.

### 3.2. ¿Qué es un Sistema Robótico?

Después de definir qué es un robot, esta sección se centrará en el concepto de sistema robótico. Aquí se explorará la estructura física del robot, así como su arquitectura computacional, destacando cómo ambos componentes se integran para permitir al robot realizar tareas autónomas y semi-autónomas.

Procedamos entonces a entender qué es un sistema robótico. Frecuentemente en la literatura encontramos este término para hacer referencia a dos subsistemas diferentes [48]:

1. Por un lado, tenemos la propia **estructura del robot**. Este subsistema es fácilmente reconocible por todos los componentes hardware que lo componen (e.g. sensores, servos, terminales finales). Siguiendo nuestra definición aportada a este proyecto, la estructura del robot la componen todos los elementos físicos que permiten al robot llevar a cabo tareas autónomas o semi-autónomas. Esta estructura incluye una variedad de componentes como:
  - **Sensores:** Dispositivos que permiten al robot percibir su entorno, como cámaras, sensores de proximidad, sensores de temperatura y acelerómetros.
  - **Actuadores:** Mecanismos que permiten al robot interactuar físicamente con su entorno, como motores, servos, y actuadores neumáticos.
  - **Controladores:** Componentes electrónicos (o virtualizados) que reciben señales de los sensores y envían comandos a los actuadores para realizar acciones específicas.
2. Por otro lado, tenemos la **arquitectura del robot**. Esta hace referencia a los conceptos computacionales que permiten la existencia del sistema. La arquitectura del robot define como se estructuran y comunican los diferentes módulos de software, determinando la manera en que el robot procesa la información y toma decisiones. Algunos de los enfoques más comunes incluyen:
  - **Arquitectura master-slave**[49]: Un modelo en el que un nodo maestro controla uno o más nodos esclavos, delegando tareas específicas a estos últimos y

recopilando los resultados.

- **Arquitectura publish-subscribe**[50]: Un patrón en el que los módulos de software se comunican mediante la publicación y suscripción a mensajes en canales específicos, facilitando la integración y escalabilidad del sistema.
- **Arquitectura client-server**[51]: Un modelo en el que un cliente solicita servicios o recursos de un servidor, que responde con los datos o acciones requeridas.

Dado que nuestro estudio se enfoca desde un punto de vista telemático, tal como se ha definido anteriormente, el concepto *robot* no es más que un medio disponible para llevar a cabo cierto servicio. Por tanto, podemos definir un sistema robótico como una interacción entre un servicio (o servicios) alojado y un conjunto no nulo de robots.

Este tipo de interacción se suele descomponer en un cúmulo de pequeñas interacciones con un objetivo final último definido por el servicio. Sobre esta base, existe un debate abierto tratando de encontrar cuál es la forma más óptima de descomposición de tareas. Algunas soluciones siguen la propuesta de operar en rango de frecuencias diferentes para cada una de estas iteraciones [52]. Otras proponen un modelo más abstracto, donde las interacciones se descomponen en conjuntos más pequeños de interacciones [53].

Aunque este sea un debate abierto, la mejor solución siempre vendrá dada por aquella que se adapte mejor al sistema que se pretenda crear. Nuestra propuesta de arquitectura se verá definida en secciones posteriores.

Por ejemplo, en un servicio de vigilancia de seguridad, los robots pueden patrullar áreas designadas, identificar personas y objetos, y comunicar sus hallazgos a un servidor central que coordina las acciones de todos los robots involucrados. Cada robot, equipado con sensores y actuadores, ejecuta algoritmos de procesamiento de datos en tiempo real, toma decisiones autónomas basadas en el contexto local y comparte información relevante con otros nodos de la red.

### 3.3. Capacidades de los componentes del sistema

Finalmente, esta sección se centrará en cómo la información contextual puede ser utilizada para mejorar la orquestación de redes y servicios en aplicaciones robóticas. Se discutirá cómo aprovechar el contexto para optimizar el rendimiento de la red, la eficiencia operativa del robot y la calidad del servicio, integrando perspectivas telemáticas y del robot para una gestión más efectiva de los recursos.

Dada la propia naturaleza de los dos componentes principales del sistema (el robot y el servidor), existen ciertas diferencias fundamentales en términos de capacidades computacionales, de comunicación, procesado de datos, y autonomía operativa a tener en cuenta.

En primer lugar, las capacidades computacionales de un servidor suelen ser, en términos generales, significativamente superiores a las de un robot. Los servidores, en cuanto a

nuestro caso nos atañe, están diseñados para ejecutar algoritmos complejos, soportar múltiples tareas simultáneas, y manejar grandes volúmenes de datos. Por otro lado, el robot, generalmente, tiene limitaciones de computación (normalmente debido a su tamaño), de uso de energía y su propósito específico de interacción con el medio.

En cuanto a la comunicación, el robot utiliza sus capacidades para interactuar con el entorno inmediato y con el servidor, siendo este último un punto central que puede coordinar la actividad de múltiples robots. Si el servicio está completamente alojado en el servidor, el robot necesita una comunicación estable para poder operar de manera efectiva en diversos entornos.

El procesamiento de datos varía en función de estos dos componentes. El servidor ofrece la capacidad de cómputo y procesamiento de una cantidad masiva de datos que no es factible en el robot. Esto permite llevar a cabo tareas de aprendizaje automático y el análisis profundo de datos que no es posible a escala del robot por sí solo. Por el contrario, no siempre es necesario el procesamiento de datos masivos en servidores para cumplir con el objetivo de la aplicación. Dicho procesamiento puede ser llevado a cabo por el robot de forma autónoma, sin necesitar un flujo constante de datos que colapse la red.

Por último, en términos de autonomía operativa, el robot está diseñado para realizar tareas de forma independiente, dentro de un marco de colaboración con el servidor. El servidor, según las necesidades de la aplicación, puede actuar como centro de mando y control, no tiene autonomía en el mundo físico, pero es esencial en la toma de decisiones a nivel de sistema.

Tanto el robot como el servidor, ofrecen capacidades únicas dada su naturaleza. El robot tiene un enfoque en la interacción dinámica con el medio y el servidor se especializa en el procesamiento de tareas que requieren mayor capacidad de cómputo. En secciones posteriores, analizaremos las capacidades de cada uno con el fin de buscar un óptimo uso del sistema.

## 4. CONTEXTO SEMÁNTICO

En este capítulo, se explorará el concepto de contexto semántico y su importancia en la interpretación y utilización de datos en diversas aplicaciones del Machine Learning. Se discutirá cómo el contexto semántico enriquece la comprensión de los datos más allá de una interpretación lineal, facilitando tareas como la desambiguación, la inferencia, el razonamiento y la interacción natural, esenciales para la colaboración humano-robot.

### 4.1. ¿Qué se entiende por Contexto Semántico?

El contexto semántico se refiere a la información adicional que enriquece la interpretación de los datos recogidos. Esta información va más allá de una interpretación lineal de los datos, proporcionando una comprensión más profunda del significado.

Este concepto se presenta en diversos campos del Machine Learning, desde el procesamiento del lenguaje natural hasta la visión por computador, ya que permite:

- **Desambiguación:** Determinar el significado preciso de palabras o símbolos en funciones entorno en el que se presentan [54](Ej. “banco” en la oración “El banco estaba lleno de gente”)
- **Inferencia:** Inferir nuevas relaciones entre datos que no se ve correlación de forma explícita [55].
- **Razonamiento:** Realizar razonamientos basados en el conocimiento y las relaciones presentes en el entorno [56].
- **Interacción natural:** Facilitar una comunicación más fluida al comprender las intenciones y el significado detrás de las palabras o de las acciones [57].
- **Planificación y toma de decisiones:** Comprender el contexto para planificar acciones y tomar decisiones acertadas en situaciones complejas [58].

En tareas de colaboración humano-robot, es esencial obtener una perspectiva integral de la interacción entre humanos y robots, lo que implica la recopilación y transformación constante de la información del entorno en conocimiento que servirá de base para la toma de decisiones y la planificación anticipada (Ver Fig 4.1).

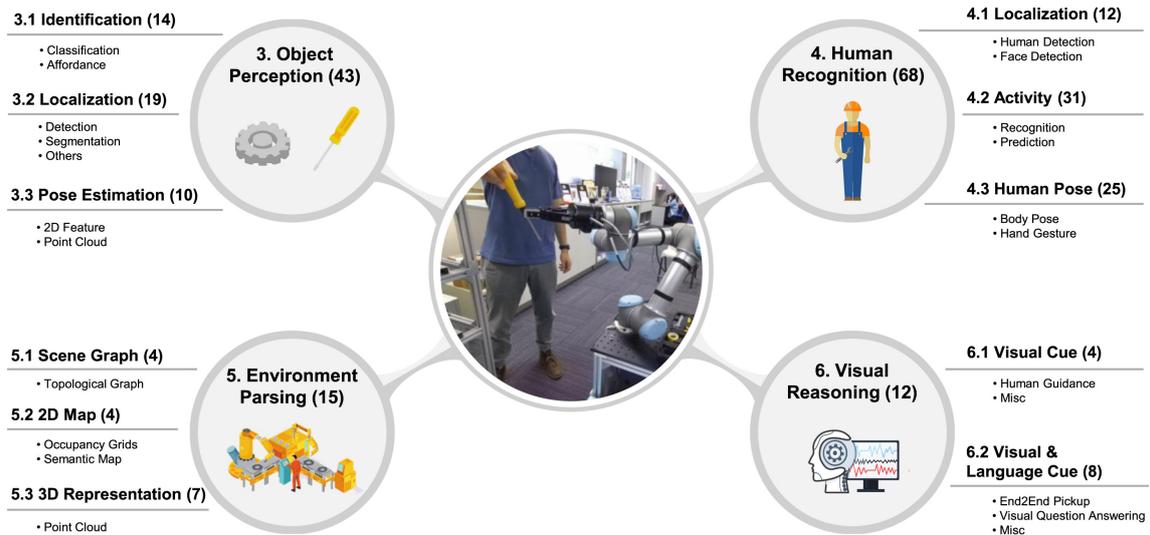


Fig. 4.1. Figura originaria de [59] que representa la información adquirida por el contexto de una interacción humano-robot.

Al integrar el contexto semántico, los robots móviles pueden optimizar su funcionamiento y sus interacciones con otros dispositivos de la red, adaptándose de manera inteligente a las dinámicas cambiantes de su entorno. La importancia de entender cuál es el contexto que rodea al robot radica en su capacidad para proporcionar al sistema una comprensión profunda del entorno, lo que permite adaptar su comportamiento y tomar decisiones más basadas.

Por ejemplo, al entender el contexto semántico, nuestro sistema puede determinar cuándo es el momento óptimo para descargar grandes cantidades de datos, o cuando priorizar ciertas tareas de comunicación en función de su aplicación actual, y las condiciones de red.

Desde un punto de vista puramente telemático, el contexto semántico hace referencia a la información adicional que puede ser capturada y utilizada para mejorar la eficiencia de las comunicaciones en red. Esto incluye la disponibilidad de ancho de banda, la latencia de la red, la calidad del servicio (QoS) y otros parámetros de la red que pueden influir en el rendimiento de las aplicaciones robóticas. Al utilizar esta información, los sistemas de red pueden priorizar el tráfico, ajustar las rutas de datos y optimizar los recursos para satisfacer mejor las necesidades de las aplicaciones robóticas.

Para el robot, el contexto semántico implica el análisis de datos del entorno físico inmediato. Esto incluye información sobre su ubicación, los objetos, y las personas presentes, las condiciones ambientales como la iluminación y el clima, y el estado interno del propio robot, como el nivel de batería y la carga de procesamiento. Esta información se usa para que el robot pueda tomar decisiones informadas, como evitar obstáculos, interactuar con personas de manera segura y eficiente, y gestionar su energía para maximizar el tiempo de operación.

La verdadera potencia del contexto semántico se revela cuando combinamos la pers-

pectiva telemática con la del robot. Esta integración permite una orquestación más efectiva de los recursos y de toma de decisiones en las tareas, mejorando tanto la eficiencia de la red como la capacidad del robot para desempeñar sus funciones. Por ejemplo, un robot puede ajustar su comportamiento en función de las condiciones de la red; si la latencia es alta, puede optar por procesar datos localmente en lugar de enviar todo a la nube. Del mismo modo, la red puede priorizar el tráfico de datos crítico de robots que se encuentran en situaciones complejas o de emergencia, como puede ser en un espacio con mucha gente.

La orquestación, por tanto, implica la gestión y la coordinación de los recursos y tareas para optimizar el rendimiento general del sistema. Con un conocimiento detallado del contexto, los sistemas pueden:

- **Adaptar dinámicamente los recursos de red:** Priorizando el tráfico y ajustando el ancho de banda (entre otros recursos) según las necesidades de los robots en tiempo real.
- **Optimizar el rendimiento del robot:** Ajustando las tareas y el procesamiento de datos en función de las condiciones actuales del entorno y de la red
- **Planificar acciones y tomar decisiones informadas:** Anticipándose a los cambios en el entorno y la red, y adaptando las estrategias de manera proactiva.

## 4.2. Tipos de Contextos

Después de entender qué es el contexto semántico, esta sección se enfocará en los diferentes tipos de contextos que pueden encontrarse en un sistema robótico. Cada tipo de contexto será definido y ejemplificado para ilustrar cómo contribuye a la funcionalidad y eficiencia de los robots en diversos entornos y aplicaciones.

### 4.2.1. Contexto Físico

El contexto físico hace referencia a la información relacionada con el entorno tangible en el que opera el robot. Incluye elementos como:

- **Ubicación:** La posición geográfica del robot. Aunque existen diferentes técnicas, la más común es el uso de *Point-Clouds* mediante el uso de *LiDAR*[60]. Hagamos un inciso para explicar estos dos conceptos, pues son importantes para entender la navegación autónoma de los robots:
  - **Light Detection and Ranging (LiDAR):** Es una tecnología de detección que utiliza pulsos de luz láser para medir distancias. Un sensor LiDAR emite pulsos de láser hacia un objeto y mide el tiempo que tarda el pulso en regresar

después de reflejarse en el objeto. Esto permite crear un mapa tridimensional preciso del entorno, que es crucial para la navegación autónoma y la evitación de obstáculos.

- **Point Cloud:** Es una colección de puntos de datos en el espacio bi o tridimensional. Estos puntos son generados por sensores (como el propio LiDAR u otros más complejos como Depth Cameras) y representan la superficie de los objetos en el entorno. Cada punto en la nube de puntos (point cloud) tiene coordenadas X, Y y Z. Las point clouds son utilizadas para modelar la geometría del entorno, facilitando tareas de mapeo, localización y percepción.
- **Objetos y Obstáculos:** La presencia, posición y movimiento de objetos y obstáculos en el entorno inmediato del robot
- **Condiciones Ambientales:** Factores como la iluminación, temperatura, humedad y condiciones del suelo que pueden afectar la operación del robot.
- **Personas:** La presencia y ubicación de personas en el entorno, incluyendo sus movimientos y acciones. Esto es un aspecto crucial para cumplir con las normativas vigentes en esta área [61].

Por ejemplo, un robot de entrega de paquetería utiliza un sensor LiDAR para crear un mapa 2D de su entorno mientras se desplaza por una oficina, detectando y evitando muebles y personas en movimiento.

#### 4.2.2. Contexto Operacional

El contexto operacional incluye información sobre el estado interno del robot y su capacidad para llevar a cabo tareas. Esto abarca:

- **Estado de la Batería:** El nivel de carga de la batería y la tasa de consumo de energía.
- **Capacidades de Procesamiento:** La carga actual y disponibilidad de los recursos de cómputo que tiene el robot.
- **Estado de los Sensores y Actuadores:** El funcionamiento y la calibración de los sensores y actuadores, así como cualquier posible fallo o necesidad de mantenimiento.

Por ejemplo, un robot de vigilancia reduce la frecuencia de patrullaje cuando su batería está baja y se dirige a la estación de carga para recargarse antes de continuar con sus tareas.

### 4.2.3. Contexto de Red

El contexto de red se refiere a la información sobre la conectividad y el rendimiento de la red en la que opera el robot. Incluye:

- **Disponibilidad de Recursos de Red:** La cantidad de ancho de banda disponible y la capacidad de los canales de comunicación.
- **Latencia:** El tiempo de retraso en la transmisión de datos a través de la red.
- **Conectividad:** La estabilidad de la conexión de red y la probabilidad de pérdida de paquetes o interrupciones. Estos conceptos se conocen en telemática con el término *reliability*.
- **Calidad de Servicio (QoS):** Los parámetros que determinan la prioridad y el tratamiento del tráfico de datos en la red.

Por ejemplo, un robot industrial en una fábrica ajusta sus operaciones de transmisión de datos según la congestión de la red, priorizando la comunicación crítica durante períodos de alta demanda de red.

### 4.2.4. Contexto de la Tarea

El contexto de tarea se refiere a la información sobre las tareas específicas que el robot debe realizar. Este contexto incluye:

- **Objetivos de la Tarea:** Los objetivos específicos que el robot debe alcanzar, como la entrega de un paquete o la vigilancia de un área.
- **Prioridades de la Tarea:** La importancia relativa de diferentes tareas que el robot (o los robots) puede estar manejando simultáneamente dentro de un sistema.
- **Dependencias de la Tarea:** Las dependencias entre diferentes tareas y subtareas, así como la secuencia en la que deben ser ejecutadas.

Por ejemplo, un robot de entrega prioriza la entrega de un paquete urgente a un cliente sobre otras tareas menos críticas, reorganizando su ruta para cumplir con el plazo de entrega.

### 4.2.5. Contexto Social

El contexto social abarca la interacción del robot con los seres humanos y otros robots. Incluye:

- **Interacciones Humanas:** Cómo los humanos se comunican y colaboran con el robot, incluyendo comandos verbales, gestos y otras formas de interacción.
- **Interacciones con Otros Robots:** La coordinación y cooperación entre múltiples robots que trabajan juntos en una tarea.
- **Normas y Protocolos Sociales:** Las reglas sociales que el robot debe seguir al interactuar con humanos, como mantener una distancia adecuada y respetar la privacidad.

Por ejemplo, un robot asistente en una oficina responde a comandos verbales de los empleados, ajustando su comportamiento para ser más colaborativo y eficiente en sus interacciones.

#### **4.2.6. Contexto Semántico Combinado**

El contexto semántico combinado implica la integración de todos los tipos de contextos mencionados anteriormente para proporcionar una visión del todo y enriquecida del entorno operativo del robot. Esta combinación permite una orquestación más eficiente de los recursos y de las tareas, mejorando tanto la toma de decisiones como la adaptabilidad en tiempo real del sistema.

Un ejemplo final podría ser: Un robot de rescate en una zona de desastre utiliza información del contexto físico (mapas 3D del terreno), operacional (estado de batería y sensores), de red (conectividad y latencia), de tarea (prioridad de rescate de personas atrapadas) y social (coordinación con equipos humanos) para tomar decisiones efectivas durante la operación de rescate.

### **4.3. Uso de Contextos para la Orquestación de Redes y Servicios**

Entender contexto semántico debe formar parte no solo para diseñar aplicaciones robóticas eficientes, sino también para optimizar el rendimiento de la red en la que operan estos robots. Desde una perspectiva telemática, es necesario tratar de definir cómo la información captada por los robots puede mejorar el rendimiento general del sistema de red. El objetivo es usar esta información para gestionar de manera óptima los recursos de red asignados a las aplicaciones robóticas.

Para mejorar dicho rendimiento, debemos aprovechar al máximo la información contextual que proporcionan los robots. Esta información puede incluir datos sobre la ubicación del robot, las condiciones ambientales, el estado de la batería y la capacidad de procesamiento. Por ejemplo, si un robot de vigilancia detecta una actividad sospechosa, la red puede priorizar la transmisión de estos datos a los servidores centrales para una respuesta rápida. De esta forma, la información más relevante se transmite sin retardos

significativos. Además, la disponibilidad de recursos de red puede variar, por ello, la información contextual del robot puede ayudar a ajustar dinámicamente la asignación de ancho de banda, o de los recursos del servidor de borde. Si un robot está en una zona con alta congestión de red, puede reducir temporalmente la calidad de transmisión de datos no críticos para asegurar que los datos importantes se transmitan sin problemas.

Al comprender el contexto físico y de red, podemos ajustar las rutas de datos para minimizar la latencia. En situaciones donde la rapidez de la comunicación es vital, como en operaciones de rescate, la red puede reconfigurarse para asegurar que las rutas de datos más rápidas y menos congestionadas se utilicen, mejorando así la eficiencia y la efectividad de la operación.

El contexto semántico también permite a los robots adaptar su comportamiento según la información recibida del entorno y de la red. Por ejemplo, un robot de entrega puede decidir esperar a que disminuya la congestión de la red antes de transmitir grandes volúmenes de datos, o cambiar a una ruta alternativa si detecta obstáculos en su camino. Esto asegura que las tareas se realicen de manera eficiente, sin interrupciones innecesarias. Además, la información sobre el estado interno del robot, como el nivel de batería y la carga del CPU, puede ser utilizada para gestionar los recursos locales de manera más eficiente. Un robot puede reducir su actividad o retornar a la base para recargarse cuando el nivel de batería es bajo, asegurando así que siempre esté operativo cuando se le necesite. La colaboración entre múltiples robots y otros dispositivos de red puede ser mejorada mediante la compartición de información contextual. En un almacén automatizado, por ejemplo, varios robots pueden coordinarse para evitar colisiones y optimizar el flujo de trabajo, compartiendo datos sobre su ubicación y tareas actuales.

Integrar los contextos también permite a la orquestación de servicios adaptarse de manera proactiva a las condiciones cambiantes, mejorando así la calidad del servicio y la eficiencia operativa. Al comprender el contexto de tarea y el estado operativo de los robots, el sistema de orquestación puede asignar tareas de manera más eficiente. Puede delegar tareas más complejas a robots con mayores capacidades de procesamiento disponibles o redistribuir tareas entre robots para balancear la carga de trabajo. La información sobre el estado de los sensores y de los actuadores puede ser utilizada para prevenir fallos antes de que ocurran. La red puede alertar a los operadores sobre la necesidad de mantenimiento de ciertos componentes, minimizando el tiempo de inactividad y asegurando un rendimiento continuo. La información contextual también permite una gestión inteligente de los recursos de la red. En un entorno donde la red está compartida por múltiples aplicaciones y dispositivos, la orquestación puede garantizar que los recursos se distribuyan de manera que se minimicen los conflictos y se maximice la eficiencia global.

Entender el contexto del robot es crucial para mejorar el rendimiento del sistema de red. La integración del contexto semántico en la orquestación de redes y de los servicios alojados, permite una adaptación ágil a las condiciones cambiantes del entorno y del sistema operativo del robot. Esto no solo optimiza el rendimiento del robot y la red, sino que

también mejora la calidad del servicio y la eficiencia operativa en diversas aplicaciones.

## 5. IMPLEMENTACIÓN DEL MODELO

En este capítulo, presentamos la definición y la implementación del modelo propuesto para la orquestación semántica y políticas de descarga en redes para robots móviles. La definición constará de la definición formal del modelo y de la formulación del problema. La implementación se lleva a cabo mediante un test-bed experimental que permite validar y caracterizar el comportamiento de las funciones robóticas en diferentes contextos operativos. Este test-bed está diseñado para evaluar la calidad y el consumo de energía de las funciones robóticas utilizando diferentes configuraciones y políticas de descarga. La metodología empleada puede extraponerse a cualquier caracterización de otras funciones robóticas no consideradas en el estudio

### 5.1. Modelo del Sistema

Con el fin de capturar todos los componentes del sistema, proponemos el siguiente modelado. Tratando de capturar todas aquellas singularidades que definen a un sistema robótico, vamos a definir la arquitectura con un Grafo Multicapa (MLG, por sus siglas en inglés). Este grafo puede verse representado en la Figura 5.1

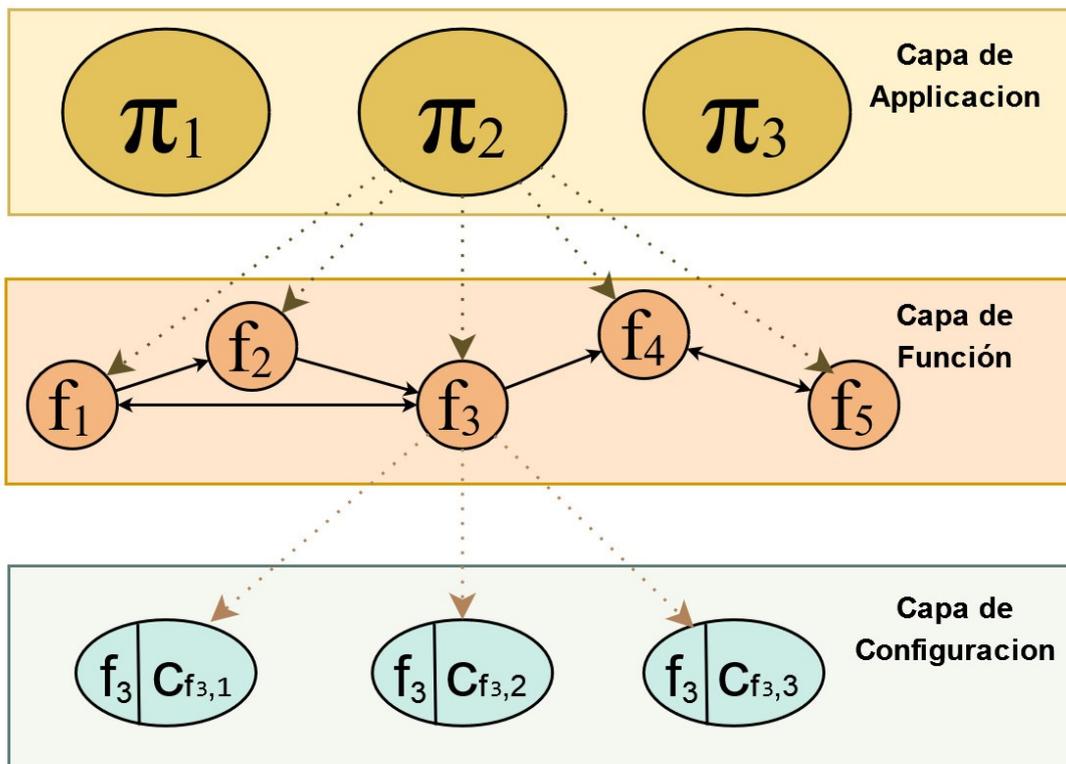


Fig. 5.1. MLG para modelar nuestra arquitectura

### 5.1.1. Aplicación Robótica

Por ende, definimos servicio, o aplicación  $\pi \in \Pi$ , como un objetivo de alto nivel que debe cumplirse mediante la realización de una, o varias funciones que cumplen criterios específicos. Este nivel funcional agrupa un conjunto relacionado de tareas que un robot puede realizar. Como ejemplos de aplicaciones tenemos servicio de entrega de paquetes, vigilancia de seguridad o servicios de limpieza. Una aplicación vendrá definida por una serie de requisitos que aseguren el correcto funcionamiento de la misma. Dentro de los muchos requisitos que podríamos definir para una aplicación (e.g. requisitos en tiempo, en latencia), vamos a declarar  $Q_\pi$  como un requisito de calidad fundamental que debe cumplirse para llevar a cabo con el objetivo del servicio.

### 5.1.2. Función Robótica

Una función  $f \in \mathcal{F}_\pi$  representa el objetivo de nivel más bajo que directamente contribuye a la ejecución de una aplicación específica, y nos sirve como bloque fundamental del sistema. Algunos ejemplos de funciones incluyen navegación autónoma, localización, controlador del robot o identificación de personas, entre otras. Por tanto, una aplicación se puede modelar como un grafo dirigido donde sus vértices ( $V_{f_\pi}$ ) y enlaces representan respectivamente: las funciones que componen un servicio, y las dependencias entre ellas. Un enlace será dirigido desde  $f$  hasta  $f'$  si la ejecución de  $f'$  es dependiente de la salida de  $f$ . Téngase en cuenta que puede darse el caso de la existencia de enlaces bi-direccionales cuando las ejecuciones de ambas funciones sean dependientes una de otra.

### 5.1.3. Configuración Robótica

Dada la capacidad que tiene un robot de llevar a cabo una misma función utilizando un conjunto de sensores, es preciso modelar nuestro sistema teniendo en cuenta esta habilidad especial. Definimos, por tanto,  $c \in C_f$  como el conjunto de todas las formas posibles de llevar a cabo una función. Pongamos un ejemplo: para una función de navegación autónoma, un robot podrá llevarla a cabo usando distintos tipos de sensores (como LIDAR [62], cámaras [63], Ondas milimétricas [64]) o una combinación de ellos. Por tanto, una función  $f$  desplegada con una configuración específica  $c$  se puede definir con el término  $f_c = (f, c)$ .

### 5.1.4. Factor de Compresión

Dado que las diferentes clases de objetivos tienen distintas tolerancias a la compresión de datos (principalmente en tareas de visión por computadora), definimos un factor de compresión  $z_{f_c} \in (0, 1]$  para modelar la calidad del flujo de datos. Cuando  $z_{f_c} = 1$ , la calidad de los datos será máxima, y disminuirá proporcionalmente a medida que  $z_{f_c}$

disminuye. Este factor de compresión permite optimizar el uso de los recursos al ajustar dinámicamente la calidad de los datos según los requisitos de la aplicación y las condiciones de la red, mejorando así el rendimiento del sistema sin comprometer la precisión de las tareas de visión por computadora. En la validación experimental de nuestra arquitectura, ofreceremos un estudio empírico en el que se trata de caracterizar diferentes factores de compresión basándose en una misma aplicación.

### 5.1.5. Recursos de Radio y Cómputo

El servidor de borde RAN puede proporcionar a nuestro marco recursos de tipo radio (por ejemplo, bloques de radio) y de cómputo (por ejemplo, CPU, GPU o RAM) para ejecutar nuestras funciones. Definimos  $K = 1, \dots, K$  como el conjunto de recursos y  $S = [S_1, S_2, \dots, S_K]$  como el vector que recoge la cantidad disponible de cada recurso de tipo  $k$ . De esta manera, definimos  $s_{f_c} = [s_{f_c,1}, \dots, s_{f_c,K}]$  como la cantidad de recursos del tipo  $k$  asignados para la función robótica  $f_c$ . La asignación eficiente de estos recursos es esencial para garantizar que las funciones robóticas se ejecuten de manera óptima, satisfaciendo los requisitos de la aplicación.

### 5.1.6. Políticas de Despliegue

Una capacidad fundamental que los robots móviles ofrecen a nuestro marco es la habilidad de realizar algunas funciones localmente, parcialmente localmente, o directamente en el servidor de borde. Definimos  $P$  como el conjunto de todas las políticas que nuestro sistema soporta. Cada función robótica debe ser implementada siguiendo una (y solo una) política, la cual se denota como  $p_{f_c}$ . Por ejemplo, una función robótica que maneja una tarea de visión por computadora puede ser desplegada en el borde si requiere un modelo de aprendizaje profundo más robusto para realizar la inferencia, o en el propio robot, donde se puede colocar un modelo de aprendizaje más ligero. Esta flexibilidad en las políticas de despliegue permite adaptar dinámicamente el sistema según las condiciones de la red, los recursos disponibles, y el propio contexto que rodea al robot, optimizando el rendimiento y la eficiencia de todo el sistema.

### 5.1.7. Consumo de Batería

Dependiendo de la política más adecuada a seguir, el consumo de energía será diferente. Definimos  $E_{tot} \in (0, 1]$  como el nivel de batería restante del robot. Cuando  $E_{tot} = 1$ , el nivel de batería es máximo y disminuirá gradualmente hasta que llegue a 0. Cuando se alcanza cierto límite de umbral de batería  $b$ , el sistema deberá enviar al robot a recargar hasta que la batería esté completamente llena. Teniendo en cuenta que los componentes mecánicos internos del robot (por ejemplo, ruedas, efectores finales, servos) están alimentados por una batería integrada separada, el consumo de energía de las funciones robóticas

depende del tiempo de CPU y de la política de despliegue seguida. Por tanto, podemos definir una función de energía que modele el consumo de energía para una función robótica como:

$$e_{f_c}^p(\mu_{p_{f_c}}, T_\pi) = \phi_{Robot} \cdot \mu_{f_c}^p \cdot T_\pi$$

donde  $\mu_{p_{f_c}}$  es el promedio de activaciones de CPU por segundo para la función  $f_c$  desplegada con la política  $p$  en el robot,  $\phi_{Robot}$  es la batería consumida por activación de CPU, y  $T_\pi$  es el tiempo de ejecución de la aplicación. Tanto  $\mu_{p_{f_c}}$  como  $\phi_{Robot}$  son dependientes del hardware. Por ende, deben ser previamente validados mediante una validación experimental para cada política, función robótica y hardware del robot.

### 5.1.8. Calidad de la Aplicación

Definimos  $q_{p_{f_c}}$  como la puntuación de calidad dada por una función de calidad  $q_{f_c}^p(z_{f_c})$ . La puntuación de calidad depende únicamente de la compresión de los datos de entrada  $z_{f_c}$ . Nótese que es necesario derivar una función de calidad específica para cada política  $p$  soportada en el sistema. Por otro lado, la relación entre la asignación de recursos, el factor de compresión y los resultados de calidad no es directa. La calidad está fuertemente influenciada por la respuesta compleja y no lineal de un modelo de aprendizaje automático. Por lo tanto, incorporar un modelo matemático intrincado para considerar cada factor involucrado es inviable y está fuera del objetivo de esta tesis. Por ello, en lugar de desarrollar un modelo matemático complejo, la estrategia a seguir implica un enfoque basado en datos en el cual la función de calidad se desarrolla utilizando un modelo de regresión, que se proporcionará al problema como entrada.

### 5.1.9. Selección de Configuración de Funciones

Una vez definido el modelo del sistema, ahora introducimos la selección de configuración de funciones. Dada  $\mathcal{F}_\pi$  como el conjunto de funciones  $f$  que componen una aplicación. Para cada  $f \in \mathcal{F}_\pi$ , la solución deberá seleccionar, dentro del conjunto  $C_f$ , la configuración adecuada para la función  $f$ . Esta selección se denota como  $f_c$ . Definimos  $x_{f_c}$  como una variable binaria de decisión que toma el valor 1 si se selecciona la configuración  $c$  para la función  $f$ . Queremos declarar que (i) si una función del conjunto  $\mathcal{F}_\pi$  no puede ser asignada siguiendo alguna configuración, la aplicación debe ser descartada, y (ii) cada función debe seguir solo una configuración.

$$\sum_{c \in C_f} x_{f_c} \leq 1, \quad \forall f \in \mathcal{F}_\pi \quad (5.1)$$

### 5.1.10. Selección de Políticas de Despliegue

Adicionalmente a la selección de configuración, la solución elegirá una política adecuada  $p$  del conjunto de políticas soportadas  $P$  para cada función robótica  $f_c$ . Denotamos  $y_{f_c}^p$  como una variable binaria de decisión que toma el valor 1 si se selecciona la política  $p$  para  $f_c$ . Dado que la elección de una política es crucial para decidir dónde asignar la función robótica (i) cada función robótica debe tener asignada una (y solo una) política y (ii) si la solución no puede determinar una política adecuada para seguir una función robótica, toda la clase de aplicación será descartada.

$$\sum_{f_c \in \mathcal{F}\pi} y_{f_c}^p f_c = x_{f_c}, \quad \forall p \in P \quad (5.2)$$

### 5.1.11. Satisfacción de los Requisitos de la Aplicación

Cada una de las funciones robóticas debe cumplir con el objetivo de calidad que definen una aplicación. Por lo tanto, para cumplir con el objetivo de calidad, la solución deberá elegir para cada  $f \in \mathcal{F}\pi$  una cantidad de recursos  $s_{f_c}$  y un factor de compresión,  $z_{f_c}$  además de una configuración  $c \in C_f$  y una política de despliegue  $p_{f_c}$ . Así, las siguientes restricciones para cada  $f \in \mathcal{F}\pi$  y  $c \in C_f$  deben cumplirse:

$$\text{mín} \{q_{f_c}^p(z_{f_c}) \cdot x_{f_c}\} \geq Q_\pi \cdot h_\pi \quad (5.3)$$

$$(5.4)$$

### 5.1.12. Mantener el Consumo de Energía Limitado

Dado que los robots dependen de baterías, si el uso esperado de energía del robot es mayor que el nivel actual de batería, la aplicación no puede mantenerse en el sistema. Es decir, para todo  $f \in \mathcal{F}\pi$  y  $c \in C_f$ :

$$\text{mín} \{e_{f_c}^p(\mu_{p_{f_c}}, T_\pi) \cdot x_{f_c}\} \geq E_{tot} \quad (5.5)$$

Cuando se alcanza un cierto límite de umbral de batería  $b$ , la solución enviará al robot a recargar y la aplicación asignada será descartada.

### 5.1.13. Limitar la Utilización de Recursos al Presupuesto Total

La cantidad de recursos elegidos para asignar una función robótica en el borde  $S_{f_c}$  no debe exceder el presupuesto total de recursos  $S$

$$\left\{ S_{f_c} \text{ es factible} \leftrightarrow S_{f_c,i} \leq S_i \quad \forall i = 1, \dots, k \right. \\ \left. \forall f_c \in \mathcal{F}_\pi \right\} \quad (5.6)$$

#### 5.1.14. Función Objetivo a resolver

Determinar una configuración adecuada, una política de despliegue y un factor de compresión de datos por función que (i) maximice el número de funciones robóticas  $f$  de  $\mathcal{F}_\pi$  asignadas en el sistema, pero (ii) minimice los recursos asignados por función robótica en el borde y (iii) minimice el uso de energía del robot es el objetivo de Nombre del problema. Así, el problema se puede formular como:

Función General Objetivo con las Restricciones

$$\begin{aligned} \max_{\mathbf{x}, \mathbf{y}, \mathbf{z}} \sum_{f \in \mathcal{F}_\pi} \sum_{c \in C_f} \sum_{p \in P} x_{f_c} y_{f_c}^p (E_{tot} - e_{f_c}^p) - \frac{1}{K} \sum_{i \leq K} \frac{S_{f_c,i}}{S_i} \\ \text{s.t. (5.1) (5.2) (5.3) (5.5) (5.6)} \end{aligned}$$

Podemos establecer la dificultad NP-hard del despliegue de aplicaciones robóticas, junto con el resto de las restricciones, mediante una reducción del problema de programación mixta-entera (MIP, por sus siglas en inglés), que se sabe que es NP-Hard. Para ello, revisamos el objetivo del MIP, que es optimizar una función lineal sujeta a un conjunto de restricciones lineales, donde algunas variables de decisión están restringidas a ser enteras y otras pueden tomar valores continuos.

En esta reducción, las variables de decisión binarias  $x_{f_c}$  y  $y_{f_c}^p$  se asignan a ser variables enteras en MIP. Simultáneamente, las variables continuas  $S_{f_c}$  y  $z_{f_c}$  corresponden a las variables continuas en MIP. La función objetivo de nuestro problema que combina la maximización del número de funciones implementadas y la minimización del uso de recursos en el borde y del uso de energía en el robot, se alinea con una función objetivo lineal en MIP, y nuestras diversas restricciones,  $Q_\pi$  y  $E_{tot}$ , se corresponden con las restricciones lineales en MIP, asegurando así una correspondencia directa entre los dos problemas.

La resolución completa del problema de optimización presentado implica el desarrollo y la implementación de algoritmos avanzados que puedan manejar la alta no linealidad y la interdependencia de las variables del sistema. Como tal, la tesis se centra en identificar y formalizar los desafíos y consideraciones fundamentales para investigaciones futuras que puedan desarrollar soluciones específicas. El propósito de esta tesis es proporcionar un análisis preliminar, una base conceptual sólida, y una validación experimental de contextos de red para un problema de orquestación semántica y políticas de descarga en redes para robots móviles. Dado el enfoque en el contexto semántico de la red, este trabajo se

centra en definir y modelar los componentes esenciales y las relaciones entre ellos, en lugar de resolver directamente el problema de optimización.

### 5.1.15. Arquitectura Física

Una vez definido el modelo, y todos sus términos, podemos definir la arquitectura física del sistema. Definimos nuestra arquitectura física como un conjunto de robots  $R$  no nulo, un conjunto de servidores  $D$ , y una conexión  $w_{d,r}$  que modelan la conexión entre el robot  $r \in R$  y el servidor  $d \in D$ . Este sistema físico puede ser implementado usando diferentes estándares de telecomunicaciones (e.g. O-RAN, ETSI. . .). La implementación de nuestro sistema definido queda fuera del objetivo de este trabajo y será propuesto como trabajo futuro. Por simplicidad, pero ajustándonos al objetivo del proyecto, vamos a considerar un sistema compuesto por un único robot, un servidor y una conexión estándar WiFi. Un ejemplo gráfico se puede encontrar en la Figura 5.2.



Fig. 5.2. Arquitectura simplificada compuesta por un servidor (a la derecha de la imagen) y un robot (a la izquierda de la imagen)

## 5.2. Definición del test-bed de la Experimentación

El test-bed experimental se centra en una aplicación de Vigilancia de Seguridad, compuesta por tres funciones principales: Navegación Autónoma, Identificación de Personas y el Control del Robot. El enfoque se centrará específicamente en la tarea de identificación de personas dentro del contexto robótico de la aplicación. La implementación nos permitirá validar y caracterizar el comportamiento de esta función robótica bajo diferentes políticas de despliegue. Este test-bed está diseñado para evaluar la calidad y el consumo de energía de la identificación de personas utilizando una transmisión de video continua desde el robot y dos políticas de descarga:

### 5.2.1. Política 1: Descarga completa en la nube

En esta política, la función de identificación de personas se descarga y ejecuta en un servidor en la nube. Esta configuración permite evaluar el impacto de la presencia de GPU en tareas de visión por computador

- **Calidad:** Se evalúa la precisión del modelo de identificación de personas con un factor de compresión cambiante y en diferentes contextos.
- **Consumo de Energía:** Se mide el consumo de batería de la cámara y la transmisión de datos a la nube con un factor de compresión cambiante y en diferentes contextos.

### 5.2.2. Política 2: Computación local en el robot

En esta política, la función de identificación de personas se ejecuta localmente en el robot, permitiendo evaluar el rendimiento del procesamiento local.

- **Calidad:** Se evalúa la precisión del modelo de identificación de personas con un factor de compresión cambiante y en diferentes contextos
- **Consumo de Energía:** Se mide el consumo de batería de la cámara y el procesamiento local de datos con un factor de compresión cambiante y en diferentes contextos.

## 6. EXPERIMENTACIÓN Y RESULTADOS

Con el fin de validar el uso de nuestro modelo planteado, en este capítulo vamos a trabajar resolviendo dos suposiciones:

1. Para el uso de diferentes aplicaciones robóticas, existen distintos umbrales de compresión de imagen que mantienen niveles equivalentes de precisión, pero se necesita una menor asignación de recursos.
2. Podemos identificar el impacto de diferentes contextos basándonos en el número de detecciones por unidad de tiempo, para posteriormente aplicar diversas estrategias de descarga en función del contexto, de la situación de la red/computación, y de la batería del robot, sin comprometer los criterios de rendimiento de la aplicación.

### 6.1. Configuración del Hardware y el Software

Esta sección describirá el hardware y software utilizados para la experimentación. Se detallará la configuración del robot Turtlebot y del servidor de borde, así como el entorno de software que incluye el uso de ROS y Docker para gestionar las políticas de descarga y la medición del consumo energético.

#### 6.1.1. Hardware

Para la configuración del hardware, se utilizará un robot Turtlebot [65] equipado con una cámara RGB y un sistema de monitoreo de batería. Este robot será el encargado de capturar imágenes y procesar datos en función de las políticas de descarga definidas.

El **robot Turtlebot** está equipado con una cámara RGB y sistemas de monitoreo de batería, lo que permite una evaluación precisa del consumo energético durante las pruebas. Estos robots cuentan con capacidades computacionales suficientes para ejecutar modelos de aprendizaje automático en local, así como para transmitir datos a servidores de borde o en la nube, dependiendo de la política de descarga implementada.



Fig. 6.1. Turtlebot equipado con una cámara RGB, un módulo de conexión cliente de WiFi y un LIDAR para el posicionamiento

El servidor de borde está configurado con un sistema operativo Linux (Ubuntu 20.04) y cuenta con una CPU de múltiples núcleos de alta frecuencia para manejar tareas de procesamiento intensivo. Además, el servidor está equipado con 32 GB de RAM y una GPU compatible con CUDA [66], lo que facilita el procesamiento de grandes volúmenes de datos y la ejecución de modelos de aprendizaje automático. La conectividad se asegura mediante routers *Turris Omnia*[67] de WiFi usando la banda de 5 GHz, creando una red inalámbrica de prueba que permite una comunicación suficiente entre el robot y el servidor de borde.

### 6.1.2. Software

El entorno de software está diseñado para maximizar la eficiencia y la compatibilidad con las herramientas utilizadas en el control robótico. El sistema operativo elegido es Ubuntu 20.04, una plataforma estable y ampliamente utilizada en aplicaciones robóticas.

Para el control robótico, utilizaremos ROS. **Robot Operating System (ROS)** [68] es un marco flexible para el desarrollo de software para robots. ROS es un conjunto de bibliotecas y herramientas que ayudan a los desarrolladores a crear aplicaciones robóticas complejas con una estructura modular. Entre las características principales de ROS se encuentran la facilidad para la integración de distintos módulos, la comunicación entre procesos, el soporte para múltiples lenguajes de programación (como Python y C++), y

una extensa colección de paquetes predefinidos que abordan una amplia gama de funcionalidades robóticas.

Entre alguna de las características principales de ROS encontramos:

- Permite dividir una aplicación robótica compleja en varios nodos independientes, cada uno responsable de una tarea específica. Estos nodos pueden ser reutilizados en diferentes proyectos, promoviendo así la modularidad y la reutilización del código.
- Proporciona varios mecanismos de comunicación entre nodos, como *topics*, *servicios* y *acciones*. Estos mecanismos permiten la transferencia eficiente de datos y la coordinación de actividades entre los diferentes componentes del sistema robótico.
- Es compatible con varios lenguajes de programación, incluyendo Python y C++, lo que permite a los desarrolladores elegir el lenguaje que mejor se adapte a sus necesidades y habilidades.
- Se integra de manera efectiva con simuladores como Gazebo, permitiendo la creación y prueba de algoritmos robóticos en entornos simulados antes de desplegarlos en robots físicos. Esto reduce significativamente el tiempo y el costo de desarrollo.
- Cuenta con una comunidad activa y una vasta colección de paquetes desarrollados por terceros que ofrecen soluciones a problemas comunes en robótica. Estos paquetes incluyen controladores para hardware específico, algoritmos de SLAM (Simultaneous Localization and Mapping), navegación, percepción, y más.

Para el experimento se utilizará ROS-Kinetic que, aunque no es una aplicación nativa para esta versión, es compatible con la distribución de Ubuntu 20.04 y proporciona un marco flexible para el desarrollo y la implementación de algoritmos robóticos.

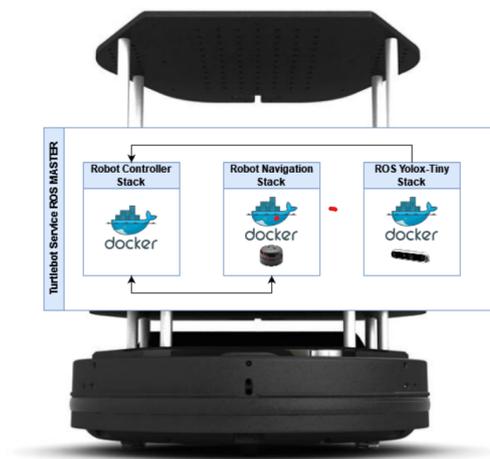


Fig. 6.2. Turtlebot con un servicio de ejemplo que define la aplicación Vigilancia de Seguridad

Además, se emplea Docker para ejecutar las pilas de contenedores de cada política de descarga. Docker permite aislar y gestionar las aplicaciones de manera eficiente, garantizando que cada política de descarga se pueda implementar y evaluar de manera independiente sin interferencias. Esta configuración facilita la repetición de experimentos y la comparación de resultados bajo diferentes condiciones operativas.

Para medir y evaluar el consumo energético del robot, hemos utilizado la herramienta Powertop. Esta herramienta nos permite registrar el uso de energía del robot bajo diferentes contextos y condiciones operativas. Con ella, y bajo las medidas que toma de un proceso corriendo en una máquina, obtenemos tanto la media como la desviación típica del consumo de batería para cada escenario, ya que esto nos permite comparar cómo varía el uso de energía en distintos contextos y validar que, a mayor cantidad de mediciones, mayor es el uso de batería.

## 6.2. Modelos de Detección de Objetos

Esta sección se centrará en los modelos de detección de objetos YoloX\_s y YoloX\_tiny, explicando sus características, diferencias y aplicaciones en entornos robóticos. Se analizará cómo estos modelos equilibran precisión y velocidad, y su adecuación para diferentes capacidades computacionales.

### 6.2.1. ¿Qué son YoloX\_s y YoloX\_tiny?

**YOLOX** (You Only Look Once eXtra) [69] es una familia de modelos de detección de objetos basados en la arquitectura YOLO (You Only Look Once), desarrollada para proporcionar una detección rápida y precisa en tiempo real. YOLOX mejora las versiones anteriores de YOLO al implementar técnicas modernas como el aprendizaje por anclaje libre y el entrenamiento de aumentación. Dentro de esta familia, Yolo\_TINY y YoloX\_S son dos variantes diseñadas para equilibrar precisión y velocidad, lo que las hace ideales para aplicaciones en entornos con recursos computacionales limitados, como los robots móviles.

**YoloX\_tiny** es una versión más ligera y rápida del modelo YOLOX, optimizada para dispositivos con capacidades computacionales reducidas. Está diseñado para funcionar eficientemente en tiempo real en hardware limitado, como los procesadores de los robots Turtlebot. A pesar de su tamaño reducido, Yolo\_tiny mantiene una buena precisión en la detección de objetos, lo que lo hace adecuado para tareas de vigilancia donde la velocidad es crítica.

Algunos datos de YoloX\_tiny:

- Tamaño del Modelo: Aproximadamente 6.2 MB.

- Velocidad: Capaz de procesar hasta 100-200 FPS (frames por segundo) en dispositivos con GPU.
- Precisión: mAP (mean Average Precision) alrededor de 40 % en el *benchmark* COCO [70].
- Aplicaciones: Ideal para sistemas embebidos y dispositivos con recursos limitados

**Yolox\_s** es una versión ligeramente más pesada que **Yolox\_tiny**, ofreciendo un mejor equilibrio entre precisión y velocidad. Este modelo es adecuado para aplicaciones donde se dispone de mayores capacidades computacionales, como en servidores de borde, y se requiere una mayor precisión en la detección. **Yolox\_s** es ideal para tareas donde la calidad de la detección es más importante que la velocidad absoluta, pero aún se necesita un rendimiento rápido.

Algunos datos de **Yolox\_s**:

- Tamaño del Modelo: Aproximadamente 17 MB.
- Velocidad: Capaz de procesar hasta 60-120 FPS en dispositivos con GPU.
- Precisión: mAP alrededor de 43 % en el *benchmark* COCO [70].
- Aplicaciones: Adecuado para entornos con mayor capacidad de procesamiento donde la precisión es crucial.

Model	Size	mAPval(0.5:0.95)	PARAMS(M)	FLOPs(G)
Yolox_S	640x640	40.5	9.0	26.8
Yolox_tiny	416x416	32.8	5.06	6.45

En la tabla anterior podemos ver una comparativa entre los dos modelos. En caso de que el lector no esté familiarizado con los términos definidos en la tabla, a continuación se expone una breve explicación de cada uno de ellos:

- **Size** se refiere al tamaño de entrada de las imágenes en píxeles.
- **mAP** es la métrica que se usa para evaluar los modelos, a mayor mAP, podemos decir que el modelo es mejor.
- **Params** es el número de parámetros que usa el modelo (en millones). A mayor número de parámetros, mayor es la complejidad del modelo. Por el contrario, un número mayor de parámetros también hace que el modelo consuma una mayor cantidad de recursos.

- **Floating Points Operations (FLOPs)** mide la cantidad de cálculos que el modelo necesita para hacer la inferencia. A mayor valor, mayor cantidad de recursos consumidos.

En conclusión, YOLOX\_s nos da un mayor rendimiento en escenarios que requieren una mayor precisión en la detección. A cambio, es computacionalmente más costoso. Por el contrario, YOLOX\_tiny ofrece un mayor balance entre eficiencia y velocidad de inferencia, a un consumo de recursos más bajo.

Durante la implementación de ambos modelos, se ha redefinido el código base para poder adquirir como datos el número de detecciones totales, las clases de objetos detectadas, el factor de compresión usado, la media de detecciones por milisegundo y el tiempo de duración del experimento. Con la media sería suficiente, pues estamos interesados en la estabilidad del contexto, más allá de los posibles picos de detecciones. De aumentar por encima de la media de detecciones en un intervalo de tiempo, se considera que el robot ha cambiado de contexto. Ambos modelos se han configurado con un intervalo de confianza del 0.3 (ninguna detección se dará por válida si el porcentaje de inferencia de la clase observada es menor al 30 %), y con un Non-Maximum Suppression (NMS) del 0.3 (técnica utilizada en los modelos de detección de objetos para filtrar y refinar las propuestas de detección y asegurarse de que cada objeto en una imagen sea detectado solo una vez).

### **6.3. Análisis del Factor de Compresión.**

En esta sección, analizamos cómo diferentes factores de compresión afectan el rendimiento de los modelos de detección de objetos YOLOX\_tiny y YOLOX\_s. Para este análisis, hemos utilizado dos imágenes donde se encuentran una persona, un objeto mediano (una silla) y un objeto pequeño (una taza) en diferentes posiciones. Las imágenes analizadas son las siguientes:



Fig. 6.3. Dos imágenes que presentan escenarios similares para la evaluación del modelo

Estas imágenes fueron elegidas porque presentan un escenario típico que un robot de vigilancia podría encontrar en su entorno operativo. La presencia de una persona, un objeto mediano y un objeto pequeño en diferentes posiciones y contextos de uso proporciona una base adecuada para evaluar la capacidad de los modelos de detección de objetos bajo distintas condiciones de compresión

Podemos pararnos un momento a analizar el porqué del uso de estas dos imágenes. Sabemos que ambos modelos utilizan un dataset (COCO, como citado anteriormente) que cuenta con una gran cantidad de muestras de personas, por tanto, esperamos que haga inferencia sobre las personas con una gran precisión en cualquiera de los dos casos.

Fijémonos primero en la imagen de la izquierda (Fig. 6.3-Izq). En esta imagen, podemos observar a una persona cogiendo una taza, apoyada en una silla. Imaginemos, por tanto, que, tratamos de identificar qué es lo que lleva la persona en la mano. Esto es especialmente necesario si el robot lleva a cabo una aplicación de vigilancia de seguridad, con el fin de, en un futuro, poder identificar las intenciones de esa persona. Con esta imagen, pretendemos evaluar ambos modelos en la inferencia, no solo de detectar personas, si también los objetos que portan.

Centrándonos en la imagen de la derecha (Fig. 6.3-Der). Ahora, observamos a una persona apoyada en una silla, la cual, tiene una taza encima. Esta imagen sirve para validar contextos en los que es necesario ser precisos en objetos pequeños, encima de otros. Un ejemplo de aplicación robótica sería aquellas en las que el robot debe monitorear espacios

para detectar comportamientos sospechosos. Por ejemplo, en una aplicación de vigilancia de seguridad, el robot puede necesitar identificar si un objeto pequeño, como una taza (u otro más sospechoso), ha sido dejado en un lugar inusual, lo cual podría indicar una actividad no autorizada en el espacio.

### 6.3.1. YoloX\_s: Descripción de las Imágenes con diferentes factores de compresión

A continuación, se presentan las dos mismas imágenes que ha sido comprimida usando diferentes factores de compresión (100,50,10,1). Estas imágenes muestran los resultados de detección de objetos utilizando el modelo YoloX\_s. La estructura de análisis será la siguiente: Comenzaremos analizando la imagen 6.3-Izq, desde el factor de compresión más alto hasta el más bajo. Acto seguido, continuaremos con 6.3-Der siguiendo el mismo análisis

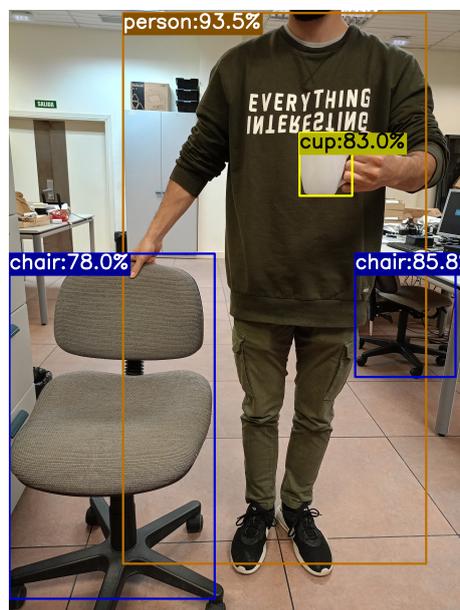


Fig. 6.4. Imagen 6.3-Izq con un factor de compresión de 100 % en YoloX\_s.

En la imagen 6.4, la calidad de la imagen es la más alta posible (factor de compresión 100). La detección de la persona, la silla y la taza es muy precisa. El modelo YoloX\_s identifica a la persona con un porcentaje de confianza del 93.5 %, la taza con un 83.0 % y la silla con un 78.0 %. La imagen muestra claramente a una persona sosteniendo una taza y apoyada en una silla. Este nivel de precisión es ideal para aplicaciones donde es crucial no perder detalles, como en sistemas de vigilancia de alta seguridad. Pero veremos que no es necesario un nivel tan alto de compresión para extraer la misma información.

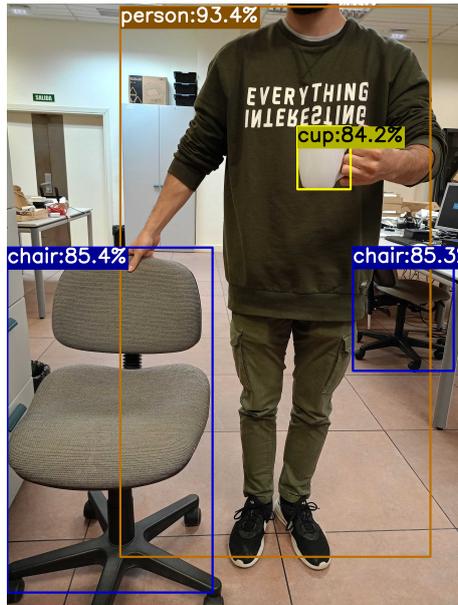


Fig. 6.5. Imagen 6.3-Izq con un factor de compresión de 50 % en Yolox\_s.

Con un factor de compresión de 50, la calidad de la imagen sigue siendo alta, aunque ligeramente inferior a la anterior. La persona es detectada con una confianza del 93.4 %, la taza con un 84.2 % y la silla con un 85.4 %. Aunque la precisión es aún alta, se observa una pequeña variación en los porcentajes de confianza, lo cual es aceptable en la mayoría de las aplicaciones de vigilancia. A niveles de transmisión de datos, hemos reducido el peso de los mismos a la mitad, lo que supone un uso eficiente de los recursos disponibles en el sistema.



Fig. 6.6. Imagen 6.3-Izq con un factor de compresión de 10 % en Yolox\_s.

En este caso, en la imagen 6.6, la compresión es significativa, y se nota una disminución en la calidad de la imagen. La persona es detectada con un 91.6 % de confianza,

la taza con un 67.6 % y la silla del fondo con un 82.1 %. Se observa que aún tenemos un alto porcentaje de confianza para con la persona, sin embargo, la silla frontal ha dejado de detectarse y el porcentaje de confianza de la taza es mucho menor. Esto implica que, este factor de compresión, podría ser óptimo para aquellas aplicaciones que solo necesiten estar advertidas de presencias de personas, como puede ser, una vigilancia de pasillos.

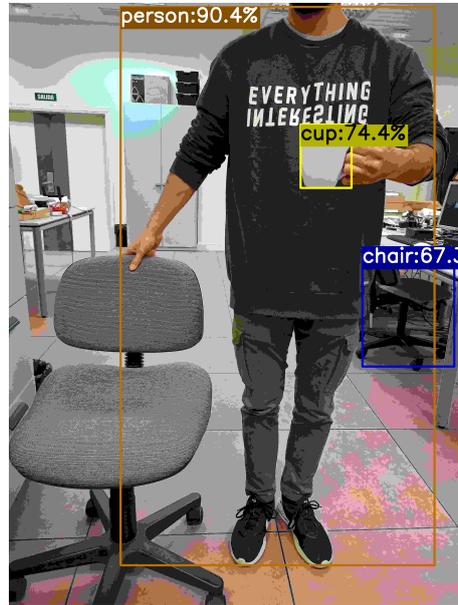


Fig. 6.7. Imagen 6.3-Izq con un factor de compresión de 1 % en Yolox\_s.

La imagen 6.7 muestra el mayor nivel de compresión (factor de compresión 1), lo que resulta en una calidad muy baja. La persona es detectada con un 90.4 % de confianza, la taza con un 74.4 % y la silla con un 67.3 %. La disminución en la precisión es significativa, especialmente para la detección de la silla. Este nivel de compresión podría no ser adecuado para aplicaciones de vigilancia que requieren una identificación precisa de todos los objetos, ya que la pérdida de calidad puede llevar a errores críticos en la detección. El aumento del porcentaje de confianza de la taza con respecto al anterior puede deberse al color de la taza. Al ser blanca y codificar los píxeles a blanco y negro, es posible que para el modelo sea más fácil identificar la taza blanca que cualquier otro objeto.

Procedamos ahora con el análisis de la siguiente imagen 6.3-Der.



Fig. 6.8. Imagen 6.3-Der con un factor de compresión de 100 % en YOLOx\_s.

En esta imagen, la calidad es la más alta posible (factor de compresión 100). La detección de la persona, la silla y la taza es muy precisa. El modelo YOLOx\_S identifica a la persona con un porcentaje de confianza del 94.1 %, la taza con un 90.7 %. La imagen muestra claramente a una persona y apoyada en una silla, pero en esta posición, no es capaz de detectar la silla. El motivo puede deberse a un desajuste con el NMS, este valor determina cuanto deben superponerse dos cuadros delimitadores antes de que uno de ellos sea suprimido. En otras palabras,  $nms=0.3$  significa que si dos cuadros delimitadores tienen un coeficiente de superposición mayor al 30 %, el cuadro con la menor puntuación de confianza será suprimido.



Fig. 6.9. Imagen 6.3-Der con un factor de compresión de 50 % en YOLOx\_s.

Con un factor de compresión de 50, como el que encontramos en la imagen 6.9, la calidad de la imagen sigue siendo alta, aunque ligeramente inferior a la anterior. La persona es detectada con una confianza del 94.0 % y la taza con un 90.8 %. Aunque la precisión es aún alta, se observa una pequeña variación en los porcentajes de confianza, lo cual es aceptable en la mayoría de las aplicaciones de vigilancia donde la claridad total no es crítica, pero sí la correcta identificación de los objetos (en caso de que se vigilen los objetos) y de las personas.



Fig. 6.10. Imagen 6.3-Der con un factor de compresión de 10 % en Yolox\_s.

Aquí, la compresión es significativa, y se nota una disminución en la calidad de la imagen. La persona es detectada con un 93.0 % de confianza y la taza con un 90.1 %. De igual forma, seguimos detectando a la persona y a la taza con un alto porcentaje de confianza.



Fig. 6.11. Imagen 6.3-Der con un factor de compresión de 1 % en Yolox\_s.

Esta imagen muestra el mayor nivel de compresión (factor de compresión 1), lo que resulta en una calidad muy baja. La persona es detectada con un 91.6 % de confianza, la taza con un 63.1 %. La disminución en la precisión es significativa, especialmente para la detección de la silla. Este nivel de compresión podría no ser adecuado para aplicaciones de vigilancia que requieren una identificación precisa de todos los objetos, dado que la pérdida de calidad puede llevar a errores críticos en la detección.

### 6.3.2. Yolox\_tiny: Descripción de las Imágenes con diferentes factores de compresión

Procedamos con el siguiente modelo. En esta sección, se presentan cuatro imágenes que han sido comprimidas utilizando diferentes factores de compresión (100, 50, 10 y 1). Estas imágenes muestran los resultados de detección de objetos utilizando el modelo Yolox\_tiny. Cada imagen contiene una persona, una silla y una taza en distintos escenarios, y se utilizan para analizar cómo la compresión afecta la precisión de detección de los objetos.



Fig. 6.12. Imagen 6.3-Izq con un factor de compresión de 100 % en YoloX\_tiny.

En la imagen 6.12, la calidad es la más alta posible (factor de compresión 100). La detección de la persona es muy precisa, con un porcentaje de confianza del 94.5 %. Sin embargo, la taza no ha sido detectada. Esto puede ser un indicativo de que, aunque la calidad de imagen es alta, el modelo YoloX\_tiny tiene limitaciones en la detección de objetos pequeños en este escenario particular. La silla también está presente en la escena, pero no ha sido detectada. Esto implica que, para tareas en las que se requiera alta precisión en la detección de objetos, este modelo no es adecuado, sin embargo, en contextos como pasillos, este modelo podría ser completamente óptimo.



Fig. 6.13. Imagen 6.3-Izq con un factor de compresión de 50 % en YoloX\_tiny.

Con un factor de compresión de 50, la calidad de la imagen sigue siendo alta. La

persona es detectada con una confianza del 94.6 %. Al igual que en la imagen anterior, la taza no ha sido detectada. Esto podría señalar una limitación en la capacidad del modelo para identificar objetos pequeños, independientemente de la calidad de la imagen. La silla, aunque claramente visible, tampoco ha sido detectada. Un factor de compresión de 50 reduce a la mitad la calidad del *stream* de video, lo que implica que la carga de datos, también se reduce a la mitad. Si nuestra tarea es identificar la presencia de personas, este factor de compresión sería completamente viable.



Fig. 6.14. Imagen 6.3-Izq con un factor de compresión de 10 % en YoloX\_tiny.

Aquí, la compresión es significativa, y se nota una disminución en la calidad de la imagen. La persona es detectada con un 94.4 % de confianza, la taza no ha sido identificada. La compresión adicional parece no afectar significativamente la detección de la persona, pero la taza sigue sin ser reconocida por el modelo. La silla tampoco ha sido detectada en esta configuración.



Fig. 6.15. Imagen 6.3-Izq con un factor de compresión de 1 % en YOLOX\_tiny.

Esta imagen muestra el mayor nivel de compresión (factor de compresión 1), lo que resulta en una calidad muy baja. La persona es detectada con un 93.9 % de confianza. La detección de la taza no se logra en esta imagen. Aun con esta compresión, seguimos detectando a la persona.

Sigamos, pues, con el análisis de la siguiente imagen 6.3-Der, procesado con el modelo YOLOX\_tiny.



Fig. 6.16. Imagen 6.3-Der con un factor de compresión de 100 % en YOLOX\_tiny.

Esta primera imagen cuenta con un factor de compresión del 100 %, lo que implica que es la imagen original. El modelo mencionado detecta con alta precisión la presencia

de una persona y una taza. La persona se detecta con una confianza del 95.2 % y la taza con un 85.2 %, pero no se detecta la silla. Esta imagen tiene la calidad más alta, lo que permite al modelo identificar claramente los objetos presentes. Con esta imagen inferida, podemos determinar que este modelo es más adecuado para escenarios en los que no se solapan objetos. En el caso de uso de vigilancia de seguridad, podríamos inferir con este modelo la presencia o no de personas, y si se detecta una persona, descargar el servicio a la nube para detectar qué lleva dicha persona en la mano.

Como curiosidad, observamos que el modelo ha detectado, como frigorífico, un armario. Esto se debe al conjunto de entrenamiento que se le ha dado al modelo. Para mejorar esta inferencia, podríamos entrenar al modelo con imágenes referentes al lugar donde se encuentra al robot, generando nuestro propio conjunto de datos. Pero esto queda fuera del objetivo del trabajo.



Fig. 6.17. Imagen 6.3-Der con un factor de compresión de 50 % en Yolox\_tiny.

En la imagen 6.17, a pesar de la reducción de la calidad, el modelo sigue detectando los mismos objetos que en la imagen original, con ligeras variaciones en los niveles de confianza. La persona se detecta con una confianza del 95.1 %, y la taza con un 86.2 %. La compresión aquí no ha afectado la capacidad del modelo para identificar estos objetos, manteniendo un alto nivel de precisión.



Fig. 6.18. Imagen 6.3-Der con un factor de compresión de 10 % en Yolox\_tiny.

Con una compresión significativa como la que encontramos en la imagen 6.18, la detección de la persona sigue siendo alta, con una confianza del 95.5 %. Sin embargo, la confianza en la detección de la taza cae drásticamente al 62.2 %. Esta reducción en la calidad comienza a afectar la capacidad del modelo para detectar correctamente todos los objetos presentes, especialmente los más pequeños como la taza.



Fig. 6.19. Imagen 6.3-Der con un factor de compresión de 1 % en Yolox\_tiny.

Esta imagen sufre la mayor pérdida de calidad debido a la compresión extrema. La persona sigue siendo detectada con una alta confianza del 94.5 %, pero la taza no es detectada en absoluto. Esto indica que, con un nivel de compresión tan alto, el modelo

pierde la capacidad de identificar objetos más pequeños y menos prominentes, resaltando los límites de detección bajo condiciones de baja calidad de imagen.

#### 6.4. Análisis de Gráficos de detecciones

Esta sección discutirá cómo la información contextual capturada por los robots puede mejorar la orquestación de redes y servicios. Se explorará cómo ajustar dinámicamente la asignación de recursos de red y optimizar el rendimiento del robot, basándose en el contexto operativo para garantizar una eficiencia y calidad de servicio óptimas.

Aquí se presentarán y analizarán gráficos que muestran el número de clases de objetos detectadas y el total de detecciones en función del factor de compresión. Estos gráficos ayudarán a visualizar el impacto de la compresión en la capacidad de los modelos para identificar objetos y la eficiencia en el uso de los recursos.

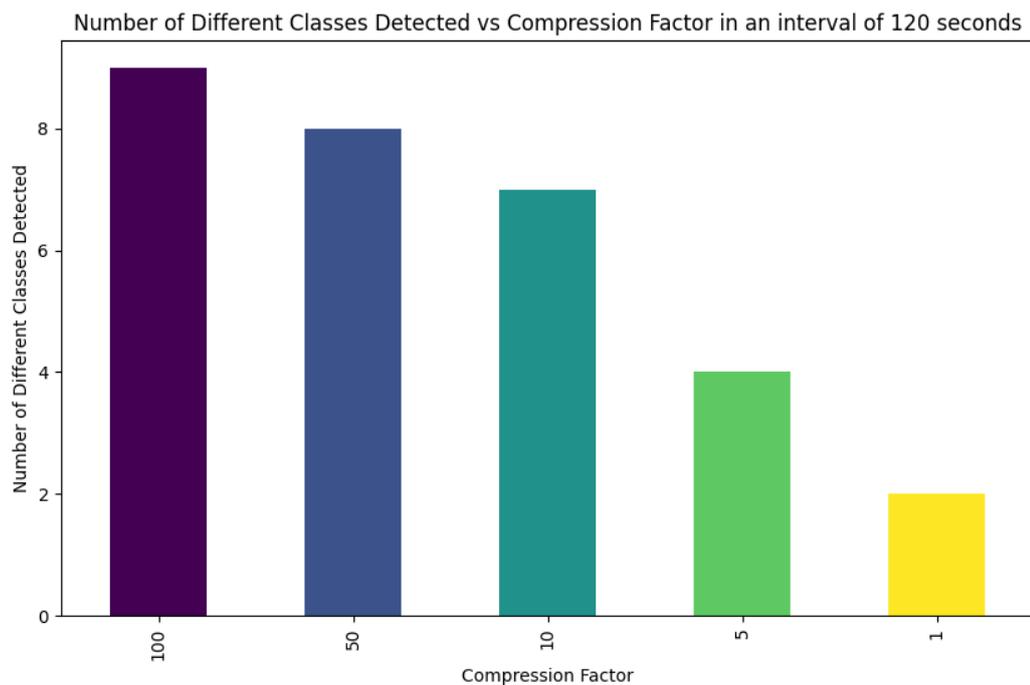


Fig. 6.20. Diagrama de barras que muestra el número de clases de objetos diferentes detectados por factor de compresión.

El primer gráfico (Fig. 6.20) muestra la cantidad de diferentes clases detectadas en función del factor de compresión en un intervalo de 120 segundos. Se observa que a medida que el factor de compresión disminuye, la cantidad de clases detectadas también disminuye. Con un factor de compresión del 100 %, se detectan 9 clases diferentes, mientras que con un factor de compresión del 1 %, solo se detectan 2 clases. Esto indica que la compresión reduce la capacidad del modelo YoloX\_s para identificar una variedad de

objetos, pero la caída se acentúa más conforme pasamos del factor de compresión del 50 %. Esta caída se observa más claramente en la siguiente gráfica.

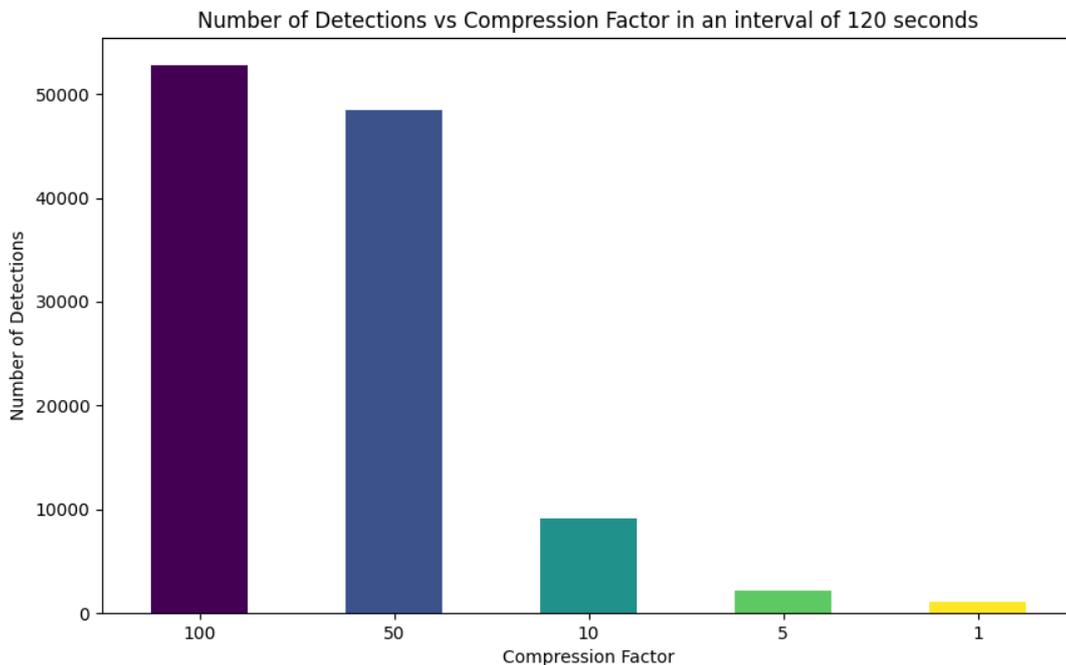


Fig. 6.21. Diagrama de barras que muestra el total de detecciones de objetos por factor de compresión.

En el gráfico 6.21 muestra el número de detecciones totales en función del factor de compresión en un intervalo de 120 segundos. Similar al primer gráfico, se observa una tendencia decreciente en el número de detecciones a medida que el factor de compresión disminuye. Con un factor de compresión del 100 %, se realizan más de 50,000 detecciones, mientras que con un factor de compresión del 1 %, las detecciones caen a un número muy bajo. El gran salto de número de detecciones lo encontramos entre el factor 50 % y el 10 %.

Comprimiendo los datos de salida de los sensores del robot al 50 % conseguimos reducir directamente el flujo de datos que atraviesan nuestra red. Esto implica un uso eficiente de la red, consiguiendo cumplir practicante los mismos objetivos. Con estos dos gráficos, y el previo análisis, queda demostrado que el uso de un factor de compresión óptimo para los objetivos de cada tarea específica, hace del sistema una red mucho más eficiente, ahorrando recursos (tanto computacionales como de señal) y consumo de energía.

## 6.5. Análisis de los Contextos.

Este análisis se centra en dos contextos distintivos: un laboratorio y un pasillo, cada uno con sus peculiaridades que afectan el funcionamiento y el consumo energético del

robot.

### 6.5.1. Contexto de Laboratorio (Laboratory)

El laboratorio se caracteriza por ser un entorno dinámico y variable. Este contexto incluye:

1. **Interacción Humana Frecuente:** Los robots deben interactuar constantemente con investigadores y técnicos, lo que implica un mayor número de detecciones de personas y objetos.
2. **Espacio de Trabajo:** El entorno incluye mesas, equipos y otros obstáculos que deben ser detectados y evitados.
3. **Conectividad:** La red en el laboratorio puede ser menos estable debido a la interferencia de otros dispositivos, afectando la transmisión de datos en tiempo real.

### 6.5.2. Contexto de Pasillo (Corridor)

El pasillo, considerando un pasillo vacío, es un entorno más estático y predecible. Las características incluyen:

1. **Tráfico Peatonal Ocasional:** El número de personas y la frecuencia de su movimiento son generalmente bajos, con picos en momentos específicos.
2. **Espacio Abierto:** Menos obstáculos comparado con el laboratorio, lo que facilita la navegación del robot.
3. **Conectividad:** La señal en el pasillo suele ser más estable, facilitando una transmisión de datos más confiable.

La herramienta *Powertop* ha sido utilizada para medir y analizar el consumo energético del robot en diferentes contextos y factores de compresión. *Powertop* proporciona métricas detalladas sobre el uso de la CPU y los "wakeups", que son activaciones del procesador por segundo. Estas activaciones del procesador impactan significativamente en el consumo de energía de la batería del robot.

El uso de la CPU en milisegundos por segundo (ms/s) es una métrica que indica cuánto tiempo de procesamiento se dedica por segundo. Un mayor uso de la CPU generalmente significa que se están realizando más operaciones de procesamiento. El uso alto de la CPU implica que el procesador está activo por más tiempo, lo que aumenta el consumo de energía y reduce la duración de la batería. Esta métrica nos sirve para entender cómo diferentes factores de compresión y contextos afectan la eficiencia energética del robot

Para llevar a cabo este análisis, se ha usado el modelo yolox\_tiny en el propio robot. No se lleva a cabo un análisis para el modelo Yolox\_s, puesto que se presupone que este modelo va a estar ejecutándose en el servidor de borde, conectado siempre a una red eléctrica estable.

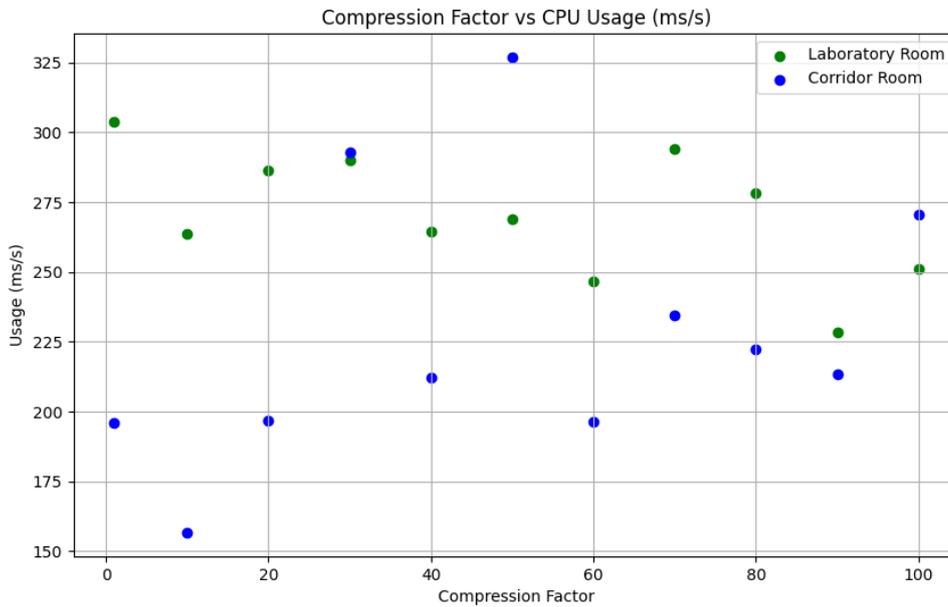


Fig. 6.22. Diagrama de puntos representando el uso de la CPU en milisegundos por segundo por factor de compresión en los diferentes contextos estudiados

La imagen proporcionada en 6.22 muestra un gráfico que relaciona el factor de compresión con el uso de la CPU en milisegundos por segundo (ms/s) para los contextos de laboratorio y pasillo.

En el contexto del laboratorio, representado por los puntos verdes, indican que el uso de la CPU es relativamente alto en este contexto, con valores que oscilan entre 200 y 325 ms/s. La alta actividad en la CPU se debe a la necesidad constante de procesar imágenes y datos de detección en un entorno con muchas interacciones y objetos. El consumo de energía es consistentemente alto debido a la necesidad de procesamiento intensivo de más datos.

En el contexto del pasillo, representado por los puntos azules, muestran un uso de la CPU más disperso, con valores entre 150 y 275 ms/s. La variabilidad en el uso de la CPU refleja los picos ocasionales de tráfico peatonal. El consumo de energía es más variable, con picos durante los períodos de alta actividad y menores niveles de consumo cuando el pasillo está vacío.

El análisis demuestra que el contexto operativo tiene un impacto significativo en el consumo energético del robot. En entornos dinámicos como el laboratorio, la carga constante y la necesidad de procesar múltiples interacciones y objetos aumentan el consumo

de energía, reflejado en un mayor uso de la CPU. En contraste, en contextos más estáticos como el pasillo, el consumo de energía es más moderado y depende principalmente de los picos de actividad.

Con este análisis de los contextos de laboratorio y pasillo, hemos demostrado cómo la dinámica del entorno impacta directamente en el uso de la CPU y, por ende, en el consumo energético de los robots móviles. Utilizando Powertop para medir el uso de la CPU en milisegundos por segundo (ms/s), hemos demostrado que en contextos más dinámicos como el laboratorio, el robot requiere un mayor procesamiento debido a la constante interacción con múltiples objetos y personas, resultando en un mayor consumo energético. En contraste, en el contexto de un pasillo, donde las interacciones son menos frecuentes y más previsibles, el uso de la CPU es menor y más variable, con picos de actividad durante eventos específicos (Ver Fig. 6.23). Estos hallazgos demuestran la importancia de considerar el contexto operativo al diseñar sistemas robóticos, permitiendo ajustes en tiempo real que mejoren la eficiencia energética sin comprometer la calidad del servicio. La capacidad de los robots para adaptarse dinámicamente a diferentes contextos no solo optimiza el rendimiento, sino que también prolonga la vida útil de la batería, haciendo que los sistemas sean más sostenibles.

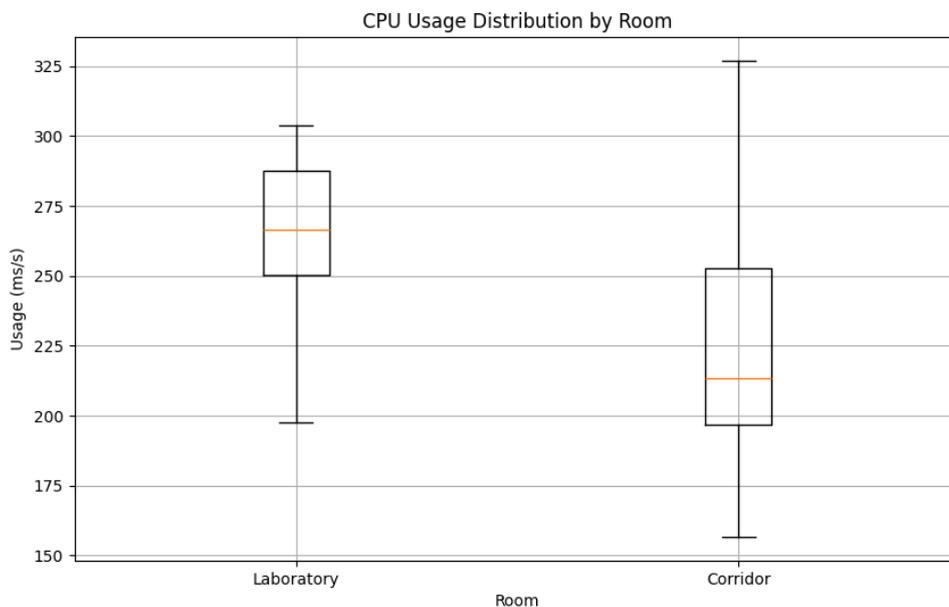


Fig. 6.23. Diagrama de cajas que expone el uso de la CPUs en los dos contextos estudiados, denotando lo explicado anteriormente en esta sección

## 7. IMPACTO SOCIOECONÓMICO DEL PROYECTO

El presente capítulo analiza el impacto socioeconómico del proyecto, enfocándose en dos dimensiones esenciales: el análisis sociológico y el análisis organizativo y económico. A medida que la tecnología robótica avanza y se integra cada vez más en nuestra vida diaria, es necesario comprender no solo los aspectos técnicos, sino también cómo estos desarrollos afectan a la sociedad. Este capítulo proporciona una visión de cómo la implementación de robots móviles puede transformar varios sectores, influir en la vida cotidiana de las personas y reconfigurar estructuras organizativas y económicas.

### 7.1. Análisis Sociológico del Proyecto

El impacto sociológico de la introducción de robots móviles en diversos contextos operativos involucra múltiples factores. Este análisis se enfoca en cómo los robots móviles afectan la interacción humana, la seguridad, el empleo y la vida cotidiana, destacando tanto los beneficios como los desafíos que esta tecnología presenta.

#### 7.1.1. Interacción Humana y Robots

La interacción entre humanos y robots es uno de los aspectos más destacados y sensibles del impacto sociológico. El concepto de "Human-In-The-Loop"(HITL) [71] juega un papel fundamental en la interacción entre humanos y robots. Este enfoque asegura que los humanos permanezcan en control de los sistemas robóticos, interviniendo en procesos críticos y tomando decisiones informadas. Los robots móviles están diseñados para operar en entornos compartidos con humanos, lo que plantea cuestiones sobre la convivencia y la cooperación. La integración de robots en espacios públicos y privados puede mejorar significativamente la eficiencia y la calidad de los servicios, pero también requiere un diseño cuidadoso para asegurar una interacción fluida y segura.

La integración de robots en entornos compartidos con humanos plantea tanto oportunidades como desafíos. Por un lado, los robots pueden mejorar la eficiencia de los servicios al asumir tareas monótonas o peligrosas, liberando a los humanos para relegar al humano a tareas más creativas. Por otro lado, es fundamental diseñar los sistemas robóticos de manera que puedan interactuar de forma segura y efectiva con las personas, minimizando riesgos y maximizando la aceptación social.

#### **Beneficios:**

- **Asistencia Personalizada:** Los robots pueden proporcionar asistencia personalizada a personas mayores o con discapacidades, mejorando su calidad de vida y autonomía. Por ejemplo, los robots de asistencia como los desarrollados por Toyota

[72], diseñados para ayudar a personas con movilidad reducida, demuestran cómo la tecnología puede integrarse para ofrecer soporte específico y personalizado. Estos robots pueden realizar tareas como recordar a los usuarios tomar sus medicamentos, ayudarles a levantarse de la cama o avisar a los cuidadores en caso de emergencia.

- **Reducción de Tareas Repetitivas:** En el ámbito laboral, los robots pueden asumir tareas repetitivas y monótonas, permitiendo a los empleados humanos centrarse en actividades más creativas y gratificantes. En fábricas automotrices, por ejemplo, los robots realizan tareas de ensamblaje repetitivas, lo que permite a los trabajadores humanos dedicarse a la supervisión y a la resolución de otros problemas complejos.

#### **Desafíos:**

- **Aceptación Social:** La aceptación de los robots por parte de la sociedad es un desafío crucial. Es necesario educar y familiarizar a las personas con la tecnología para reducir el miedo y la desconfianza. Iniciativas educativas y demostraciones públicas, como las realizadas por Boston Dynamics con su robot Spot [73], son esenciales para aumentar la confianza del público. Además, es importante involucrar a las comunidades en el diseño y la implementación de soluciones robóticas para asegurar que se aborden las preocupaciones y se maximicen los beneficios.
- **Privacidad:** La presencia de robots equipados con cámaras y sensores plantea preocupaciones sobre la privacidad y el uso de datos personales. Es fundamental desarrollar políticas transparentes sobre el manejo y la protección de los datos recogidos por los robots para abordar estas preocupaciones. Dichas políticas deben contemplar quién tiene acceso a los datos, cómo se almacenan y durante cuánto tiempo se conservan, así como implementar medidas de seguridad para protegerlos contra accesos no autorizados.

### **7.1.2. Vida Cotidiana**

La introducción de robots móviles en la vida cotidiana tiene el potencial de transformar diversos aspectos de nuestro día a día, desde la asistencia en el hogar hasta la mejora de los servicios públicos.

Los robots están comenzando a desempeñar un papel importante en los hogares y en las comunidades, realizando tareas que van desde la limpieza y el cuidado del jardín hasta la vigilancia y la entrega de paquetes. Esta tendencia tiene el potencial de cambiar la forma en que vivimos y trabajamos, haciéndonos la vida más fácil y eficiente.

#### **Beneficios:**

- **Asistencia en el Hogar:** Los robots pueden realizar tareas domésticas como la limpieza, el cuidado de jardines o la vigilancia del hogar, facilitando la vida diaria de

las personas. Ejemplos de ello son los robots de limpieza autónomos como *Roomba* [74], que se han convertido en una herramienta común en muchos hogares.

- **Mejora de Servicios Públicos:** En el ámbito urbano, los robots pueden contribuir a mejorar la eficiencia de servicios públicos como la recolección de basura, el mantenimiento de infraestructuras y la gestión del tráfico. En ciudades como Singapur, se están implementando robots para la recolección de basura y el mantenimiento de parques, mejorando la eficiencia y reduciendo los costos operativos. Además, los robots pueden desempeñar un papel importante en la gestión del tráfico, monitoreando y ajustando los flujos vehiculares para reducir la congestión.

#### **Desafíos:**

- **Dependencia Tecnológica:** La creciente dependencia de la tecnología robótica puede llevar a problemas si los sistemas fallan o si hay una falta de infraestructura adecuada. Es crucial diseñar sistemas redundantes y planes de contingencia para asegurar la continuidad de los servicios en caso de fallos tecnológicos. Además, es importante educar a los usuarios sobre el uso y el mantenimiento adecuado de los robots para minimizar el riesgo de fallos.
- **Impacto en la Vida Social:** La interacción social puede verse afectada si las personas comienzan a depender excesivamente de los robots para tareas que tradicionalmente involucraban interacción humana. Es importante fomentar un equilibrio entre el uso de la tecnología y la interacción humana para mantener el tejido social y evitar el aislamiento. Las comunidades y las familias deben ser conscientes de la importancia de mantener las relaciones humanas y no depender únicamente de los robots para la interacción.

### **7.1.3. Seguridad**

La seguridad es un aspecto crítico en la implementación de robots móviles. Estos dispositivos deben operar de manera segura y confiable en contextos dinámicos y a menudo impredecibles. La seguridad abarca tanto la prevención de accidentes como la protección de datos y la ciberseguridad.

La introducción de robots en diversos entornos, desde fábricas hasta espacios públicos, requiere un enfoque riguroso para garantizar la seguridad tanto de las personas como de los propios robots. Los sistemas deben ser capaces de detectar y reaccionar a situaciones peligrosas, evitar colisiones y operar de manera estable incluso en condiciones adversas. Además, la protección de datos y la ciberseguridad son esenciales para prevenir el acceso no autorizado y la manipulación maliciosa de los sistemas.

#### **Beneficios:**

- **Prevención de Accidentes:** Los robots pueden ayudar a prevenir accidentes en entornos peligrosos, como fábricas o plantas de energía, mediante la realización de tareas arriesgadas que de otro modo serían llevadas a cabo por humanos. Por ejemplo, los robots utilizados en la inspección y mantenimiento de plantas nucleares pueden acceder a áreas de alto riesgo, reduciendo la exposición humana a la radiación. En la industria de la construcción, los robots pueden manejar materiales pesados o trabajar en alturas, minimizando el riesgo de lesiones para los trabajadores.
- **Monitoreo Continuo:** La capacidad de los robots para operar las 24 horas del día permite un monitoreo continuo y una respuesta rápida ante emergencias, mejorando la seguridad general. Un caso destacado es el uso de drones para la vigilancia de grandes áreas forestales [75], ayudando a la detección temprana de incendios y facilitando una respuesta rápida. En el ámbito urbano, los robots pueden patrullar áreas sensibles, proporcionando información en tiempo real a los equipos de seguridad.

#### **Desafíos:**

- **Ciberseguridad:** Los robots conectados a redes pueden ser vulnerables a ataques cibernéticos, lo que podría comprometer la seguridad del sistema. Es necesario implementar medidas de ciberseguridad robustas para proteger estos sistemas contra accesos no autorizados y manipulaciones malintencionadas.
- **Mantenimiento y Fallos:** Asegurar el mantenimiento adecuado y la prevención de fallos técnicos es un enfoque básico para evitar accidentes y garantizar el funcionamiento seguro de los robots. El desarrollo de protocolos de mantenimiento preventivo y sistemas de diagnóstico automático puede ayudar a mitigar estos riesgos. Además, es importante capacitar a los operarios para que puedan identificar y resolver problemas técnicos rápidamente.

#### **7.1.4. Empleo**

El impacto de la robótica en el empleo es un tema de debate significativo. Si bien la automatización puede llevar a la eliminación de ciertos puestos de trabajo, también crea nuevas oportunidades y puede transformar positivamente el panorama laboral.

La automatización y la robótica están cambiando la naturaleza del trabajo en muchos sectores. Mientras que algunas tareas pueden ser totalmente automatizadas, otras requieren la supervisión y el control humano. Esto implica una necesidad creciente de trabajadores con habilidades técnicas y conocimientos específicos en robótica y tecnologías relacionadas.

#### **Beneficios:**

- **Creación de Nuevos Empleos:** La demanda de especialistas en robótica, ingenieros y técnicos en mantenimiento aumenta con la adopción de tecnologías robóticas. Además, surgen nuevas profesiones relacionadas con el desarrollo, la programación y la supervisión de sistemas robóticos avanzados. El crecimiento del sector de la robótica ha llevado a una mayor demanda de desarrolladores de software especializados en inteligencia artificial y aprendizaje automático.
- **Mejora de Condiciones Laborales:** Al asumir tareas peligrosas o repetitivas, los robots pueden mejorar las condiciones laborales, reduciendo el riesgo de lesiones asociado con trabajos peligrosos. En el sector de la construcción, por ejemplo, los robots que manejan materiales pesados pueden disminuir significativamente el riesgo de lesiones entre los trabajadores humanos. Además, los robots pueden trabajar en entornos peligrosos o insalubres, protegiendo así la salud de los empleados

#### **Desafíos:**

- **Desplazamiento de Trabajadores:** La automatización puede resultar en el desplazamiento de trabajadores en sectores donde las tareas pueden ser completamente automatizadas. Es fundamental desarrollar estrategias y políticas de reubicación laboral para mitigar este impacto y ofrecer oportunidades de reentrenamiento. Los programas de reconversión laboral y las políticas de protección social pueden ayudar a los trabajadores a adaptarse a los cambios en el mercado laboral.
- **Necesidad de Reentrenamiento:** En relación con el punto anterior, es esencial invertir en programas de reentrenamiento y educación continua para ayudar a los trabajadores a adaptarse a nuevas funciones en un entorno cada vez más automatizado. Iniciativas como las ofrecidas por instituciones educativas y empresas tecnológicas para la capacitación en habilidades digitales y robóticas son cruciales para facilitar esta transición. Además, la colaboración entre el sector público y privado puede garantizar que los programas de formación estén alineados con las necesidades del mercado laboral.

## **7.2. Análisis Organizativo y Económico del Proyecto**

### **7.2.1. Organización Secuencial Temporal del Proyecto**

El desarrollo e implementación de este trabajo se ha estructurado en una serie de fases definidas para asegurar un progreso ordenado. A continuación, se presenta una descripción detallada de cada fase, acompañada de un diagrama de Gantt para una visualización clara del cronograma del proyecto.

### **Fase 1: Investigación y Planificación (Meses 1-3)**

- **Objetivo:** Establecer los fundamentos teórico-prácticos del proyecto.
- **Actividades:** Revisión de la literatura, definición de objetivos específicos, planificación detallada de tareas y recursos.
- **Entregables:** Documento de planificación del proyecto, lista de recursos necesarios.

### **Fase 2: Diseño del Sistema (Meses 4-6)**

- **Objetivo:** Desarrollar un diseño detallado del sistema robótico y las redes semánticas.
- **Actividades:** Diseño de la arquitectura del sistema, selección de componentes hardware y software, creación de diagramas de flujo y esquemas de la red.
- **Entregables:** Documentación del diseño del sistema, diagramas de flujo, esquemas de la red.

### **Fase 3: Desarrollo e Implementación (Meses 7-12)**

- **Objetivo:** Implementar el sistema de acuerdo con el diseño especificado.
- **Actividades:** Desarrollo de software, ensamblaje del hardware, integración de componentes, pruebas iniciales de sistema.
- **Entregables:** Prototipo funcional del sistema, informes de pruebas iniciales.

### **Fase 4: Pruebas y Validación (Meses 13-15)**

- **Objetivo:** Validar el sistema mediante pruebas.
- **Actividades:** Realización de pruebas funcionales, ajustes y optimizaciones, validación de resultados contra criterios establecidos.
- **Entregables:** Informe de validación, sistema ajustado y optimizado.

### **Fase 5: Documentación y Presentación (Meses 16-18)**

- **Objetivo:** Documentar el proyecto y preparar su presentación final.
- **Actividades:** Elaboración de la documentación final, preparación de la presentación, revisión y refinamiento de todos los materiales finales.

- **Entregables:** Documentación completa del proyecto, presentación final.

A continuación, se presenta el diagrama de Gantt que ilustra la secuencia y la duración de cada fase del proyecto:

**Diagrama de Gantt del proyecto**



## **7.2.2. Costes Asociados al Proyecto**

El análisis económico del proyecto incluye la estimación de todos los costes necesarios para llevar a cabo cada una de las fases descritas anteriormente. Estos costes se dividen en categorías de personal, hardware, software, y otros gastos operativos.

### **Costes de Personal**

Sobre el bruto anual

- Ingeniero investigador telemático (30.000€)
- Personal de Soporte (2 personas de 28.320€)

### **Costes de Hardware**

- Robots móviles: Turtlebot (5 robots de 1600€)
- Sensores: Cámaras (5 de 314€)
- Servidores de borde (2 de 45.215€)

### **Costes de Software**

Sobre el global de cada software, se estima un gasto de 50.000€ que incluye:

- Licencias de software especializado
- Herramientas de desarrollo y pruebas

### **Otros Gastos Operativos**

- Formación del personal (15.000€)
- Gastos administrativos (10.000€)
- Contingencias (5.000€)

A continuación se presenta una tabla detallada con los costes estimados para cada categoría y un diagrama circular en la Figura 7.1:

Categoría	Desglose	Coste Estimado (€)
Personal	Investigador y Soporte	86,640
Hardware	Robots y Sensores	100,000
Software	Licencias y Herramientas	50,000
Otros Gastos Operativos	Formación y Administración	30,000
<b>Total</b>		<b>266,640</b>

TABLA 7.1. COSTES ESTIMADOS DEL PROYECTO

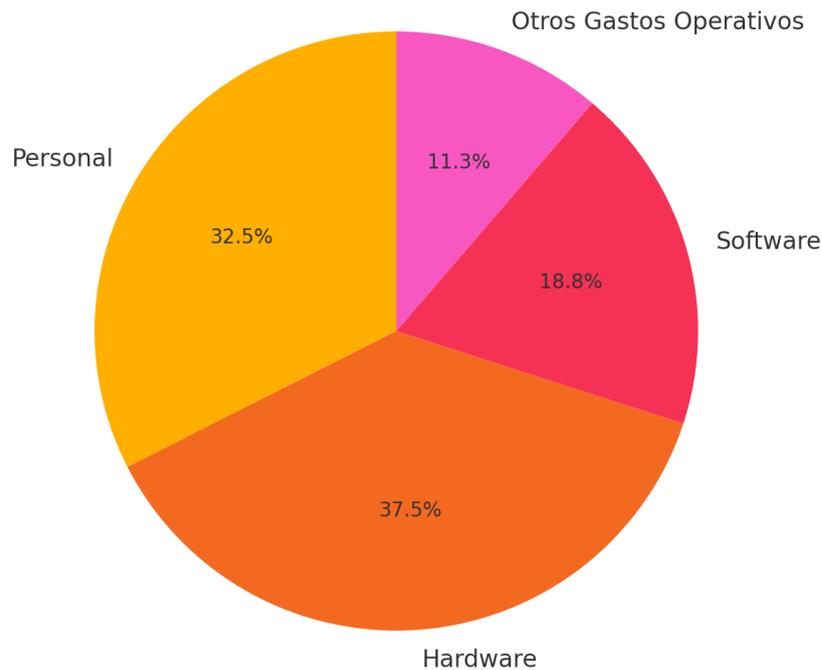


Fig. 7.1. Distribución de Costes Estimados del Proyecto

### 7.2.3. Análisis Económico

El análisis económico del proyecto se basa en el retorno de la inversión (ROI) y en la evaluación de los beneficios a largo plazo. Se estima que la implementación de este sistema puede generar ahorros significativos en diversas industrias, gracias a la optimización de procesos y la mejora en la eficiencia operativa. A continuación se presenta un análisis detallado de los beneficios esperados y el ROI proyectado

#### Beneficios Esperados

- **Ahorro en Costes Operativos:** Reducción de costos laborales y operativos mediante la automatización de tareas peligrosas
- **Incremento de la Eficiencia:** Mejora en la eficiencia operativa gracias a la optimización de procesos y la reducción de errores humanos.

- **Mejora en la Calidad del Servicio:** Incremento en la calidad y rapidez de los servicios prestados, lo que puede traducirse en una mayor satisfacción del cliente.

### **Retorno de la Inversión (ROI)**

El ROI se calcula considerando los ahorros y beneficios generados en relación con la inversión total realizada. Se estima que los ahorros anuales podrían superar los 200,000 €, lo que implicaría un retorno positivo en menos de tres años.

Por ende, la implementación del proyecto no solo es viable desde el punto de vista técnico, sino también desde el económico, ofreciendo significativos beneficios a largo plazo y un retorno de inversión atractivo.

## 8. CONCLUSIONES Y TRABAJO FUTURO

La investigación presentada en este trabajo destaca la relevancia de integrar contextos semánticos en la orquestación y gestión de redes para sistemas robóticos. A través de una validación experimental utilizando datos reales y en tiempo real, hemos demostrado que la consideración de la información contextual permite mejorar significativamente la eficiencia operativa, el consumo de energía y la calidad del servicio en sistemas robóticos.

Uno de los logros más importantes de este estudio es la propuesta de un modelo de colaboración entre el robot y el sistema, que tiene en cuenta el contexto en el que opera el robot. Este modelo no solo optimiza el uso de la red, sino que también facilita una interacción más inteligente entre los diferentes componentes del sistema. La validación de las políticas de descarga se ha realizado mediante un conjunto de métricas que incluyen la precisión de la detección de objetos, el consumo de energía y la calidad del servicio, mostrando que es posible lograr un equilibrio adecuado entre eficiencia y calidad de servicio mediante la adaptación dinámica de las políticas de descarga basadas en el contexto.

Actualmente, se está trabajando en el desarrollo de un algoritmo que solucione la función objetivo propuesta en este trabajo. Este algoritmo se centrará en optimizar la asignación de recursos y la gestión de la red, considerando las condiciones contextuales en tiempo real. Además, se están explorando diferentes funciones de selección de políticas basadas en el contexto y en las métricas obtenidas en tiempo real del ambiente del robot. Estas futuras investigaciones no solo mejorarán la eficiencia y la robustez de los sistemas robóticos, sino que también abrirán nuevas posibilidades para aplicaciones en entornos dinámicos y no estructurados.

La integración de estos avances permitirá desarrollar sistemas robóticos más inteligentes y autónomos, capaces de operar de manera eficiente en una variedad de contextos. Este enfoque contribuirá a la creación de soluciones tecnológicas avanzadas que pueden ser aplicadas en múltiples sectores, desde la vigilancia y la seguridad hasta la manufactura y la asistencia personal, impulsando así el progreso y la innovación en la robótica

### 8.1. Desafíos y Futuras Direcciones

La robótica contextualmente consciente enfrenta una serie de desafíos que deben ser abordados para maximizar su potencial y permitir la integración efectiva de estas tecnologías en diversos campos. Uno de los principales desafíos es la percepción y manejo de la incertidumbre en entornos no estructurados y dinámicos. Los sistemas robóticos deben ser capaces de adaptarse a cambios imprevistos y gestionar la variabilidad del contexto en tiempo real. Esto requiere el desarrollo de algoritmos avanzados de percepción y toma de decisiones que puedan operar con información incompleta o ruidosa[76]. Los frameworks

como [77] son ejemplos de enfoques que buscan mejorar la interacción humano-robot mediante el uso de representaciones semánticas del entorno para reducir las ambigüedades en la percepción y la acción.

Otro desafío significativo es la coordinación y colaboración entre múltiples robots. La comunicación y la sincronización de tareas en sistemas de robots distribuidos requieren métodos eficientes para el intercambio de datos contextuales y la planificación conjunta. La robótica en la Industria 4.0, por ejemplo, debe abordar cómo los robots móviles pueden trabajar en conjunto en fábricas inteligentes, utilizando computación en el borde para reducir la latencia y mejorar la precisión de las acciones coordinadas [78].

Además, el desarrollo de interfaces intuitivas para la interacción humano-robot sigue siendo un área de enfoque importante. Los robots deben ser capaces de comprender y responder adecuadamente a las señales humanas, incluyendo el lenguaje natural y las expresiones faciales. Esto requiere la integración de tecnologías de procesamiento de lenguaje natural y reconocimiento de emociones para mejorar la interacción y la colaboración [79].

Desde el punto de vista más telemático, se requiere abordar la eficiencia en la asignación de recursos en redes de comunicaciones avanzadas, como se propone en frameworks como SEM-O-RAN, que permite una asignación flexible de recursos en el borde de la red para soportar tareas de aprendizaje profundo. Estos sistemas deben optimizar la latencia y la precisión de las tareas ejecutadas en el borde, garantizando al mismo tiempo que no se sobreprovisionen recursos, lo cual es esencial para la sostenibilidad y eficiencia de las redes 5G y futuras [80].

Por último, la integración de tecnologías avanzadas como el 6G y la computación en el borde presenta nuevos desafíos y oportunidades. La orquestación de servicios en estas redes requiere soluciones innovadoras para manejar la complejidad y variabilidad de los requisitos de calidad del servicio y latencia. Soluciones como OREO proponen enfoques para la orquestación eficiente de aplicaciones en redes abiertas y programables, lo que es crucial para el desarrollo de la robótica contextualmente consciente [81].

## BIBLIOGRAFÍA

- [1] P.-6. Consortium, *PREDICT-6G: Towards a deterministic 6G network: reliable, time sensitive and predictable*, <https://predict-6g.eu/>, Funded by the European Union's Horizon Europe Research and Innovation programme under grant agreement N° 101095890, 2023.
- [2] H.-X.-I. Consortium, *Hexa-X-II: The second phase of the European 6G flagship initiative*, <https://hexa-x-ii.eu/>, Funded by the European Commission under the Smart Network and Services Joint Undertaking (SNS-JU), 2023.
- [3] E. A. Oyekanlu et al., "A Review of Recent Advances in Automated Guided Vehicle Technologies: Integration Challenges and Research Areas for 5G-Based Smart Manufacturing Applications," *IEEE Access*, vol. 8, pp. 202 312-202 353, 2020. doi: [10.1109/ACCESS.2020.3035729](https://doi.org/10.1109/ACCESS.2020.3035729).
- [4] S. Rahmanpour y R. M. Esfanjani, "Decentralized Energy-Aware Co-Planning of Motion and Communication Strategies for Networked Mobile Robots," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 12, pp. 519-528, 2020. doi: [10.1109/TCDS.2019.2932751](https://doi.org/10.1109/TCDS.2019.2932751).
- [5] J. Crespo, J. C. Castillo, Ó. M. Mozos y R. Barber, "Semantic Information for Robot Navigation: A Survey," *Applied Sciences*, 2020. doi: [10.3390/app10020497](https://doi.org/10.3390/app10020497).
- [6] X. Mu, Y. Liu, L. Guo, J. Lin y R. Schober, "Intelligent Reflecting Surface Enhanced Indoor Robot Path Planning: A Radio Map-Based Approach," *IEEE Transactions on Wireless Communications*, vol. 20, pp. 4732-4747, 2020. doi: [10.1109/TWC.2021.3062089](https://doi.org/10.1109/TWC.2021.3062089).
- [7] Z. Chen, K.-C. Chen, C. Dong y Z. Nie, "6G Mobile Communications for Multi-Robot Smart Factory," *J. ICT Stand.*, vol. 9, 2021. doi: [10.13052/jicts2245-800x.934](https://doi.org/10.13052/jicts2245-800x.934).
- [8] G. Patti, L. Leonardi y L. L. L. Bello, "A Novel MAC Protocol for Low Datarate Cooperative Mobile Robot Teams," *Electronics*, 2020. doi: [10.3390/electronics9020235](https://doi.org/10.3390/electronics9020235).
- [9] I. M. R. Fern'andez, "Advancing Robot Autonomy for Long-Horizon Tasks," *ArXiv*, vol. abs/2307.13163, 2023. doi: [10.48550/arXiv.2307.13163](https://doi.org/10.48550/arXiv.2307.13163).
- [10] A. Barnawi, "An Advanced Search and Find System (ASAFS) on IoT-Based Mobile Autonomous Unmanned Vehicle Testbed (MAUVET)," *Arabian Journal for Science and Engineering*, vol. 45, pp. 3273-3287, 2020. doi: [10.1007/s13369-020-04364-2](https://doi.org/10.1007/s13369-020-04364-2).
- [11] S. Wang y H. Chen, "Optical Integrated Sensing and Communication for Cooperative Mobile Robotics: Design and Experiments," *2023 21st International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 207-214, 2023. doi: [10.23919/WiOpt58741.2023.10349823](https://doi.org/10.23919/WiOpt58741.2023.10349823).

- [12] J. Guo y J. Ma, “Advances in collaborative control of multiple mobile robots in human-robot co-integration environment,” vol. 12940, pp. 1294018 - 1294018-5, 2023. doi: [10.1117/12.3010792](https://doi.org/10.1117/12.3010792).
- [13] J. Roening y J. Riekkki, “Context-aware mobile systems for managing services,” vol. 4572, 2001. doi: [10.1117/12.444221](https://doi.org/10.1117/12.444221).
- [14] C. Hong, J. Cho, K.-W. Lee, Y.-H. Suh, H. Kim y H.-C. Lee, “Developing a Context-Aware System for Providing Intelligent Robot Services,” pp. 174-189, 2006. doi: [10.1007/11907503\\_13](https://doi.org/10.1007/11907503_13).
- [15] D. Chang y B. Han, “Knowledge-based Visual Context-Aware Framework for Applications in Robotic Services,” *2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pp. 1-9, 2023. doi: [10.1109/WACVW58289.2023.00012](https://doi.org/10.1109/WACVW58289.2023.00012).
- [16] B. N. Schilit, N. Adams y R. Want, “Context-Aware Computing Applications,” *1994 First Workshop on Mobile Computing Systems and Applications*, pp. 85-90, 1994. doi: [10.1109/WMCSA.1994.16](https://doi.org/10.1109/WMCSA.1994.16).
- [17] T. Stipanovic, B. Jerbic y P. Curkovic, “A context-aware approach in realization of socially intelligent industrial robots,” *Robotics and computer-integrated manufacturing*, vol. 37, pp. 79-89, 2016.
- [18] C. C. Phiri, Z. Ju y H. Liu, “Accelerating Humanoid Robot Learning from Human Action Skills Using Context-Aware Middleware,” pp. 563-574, 2016. doi: [10.1007/978-3-319-43506-0\\_49](https://doi.org/10.1007/978-3-319-43506-0_49).
- [19] L. Sabri, S. Bouznad, S. Fiorini, A. Chibani, E. P. e Silva e Y. Amirat, “An integrated semantic framework for designing context-aware Internet of Robotic Things systems,” *Integr. Comput. Aided Eng.*, vol. 25, pp. 137-156, 2017. doi: [10.3233/ICA-170559](https://doi.org/10.3233/ICA-170559).
- [20] A. A. Al-Shargabi, F. Siewe y A. T. Zahary, “Quality of Context in Context-Aware Systems,” *EAI Endorsed Trans. Context aware Syst. Appl.*, vol. 4, e5, 2017. doi: [10.4108/eai.6-7-2017.152761](https://doi.org/10.4108/eai.6-7-2017.152761).
- [21] M. Benerecetti, P. Bouquet y M. Bonifacio, “Distributed Context-Aware Systems,” *Human-Computer Interaction*, vol. 16, pp. 213-228, 2001. doi: [10.1207/S15327051HCI16234\\_06](https://doi.org/10.1207/S15327051HCI16234_06).
- [22] F. Schreiber y M. E. Valcher, “Formal assessment of some properties of Context-Aware Systems,” *ArXiv*, vol. abs/2005.00373, 2019.
- [23] J. Zheng, Q. Zhang, S. Xu, H. Peng y Q. Wu, “Cognition-Based Context-Aware Cloud Computing for Intelligent Robotic Systems in Mobile Education,” *IEEE Access*, vol. 6, pp. 49 103-49 111, 2018. doi: [10.1109/ACCESS.2018.2867880](https://doi.org/10.1109/ACCESS.2018.2867880).

- [24] N. Brgulja, R. Kusber, K. David y M. Baumgarten, "Measuring the Probability of Correctness of Contextual Information in Context Aware Systems," *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pp. 246-253, 2009. doi: [10.1109/DASC.2009.114](https://doi.org/10.1109/DASC.2009.114).
- [25] J. R. Hoyos, J. Molina, J. A. Blaya y D. Preuveneers, "A model-driven approach for quality of context in pervasive systems," *Comput. Electr. Eng.*, vol. 55, pp. 39-58, 2016. doi: [10.1016/j.compeleceng.2016.07.002](https://doi.org/10.1016/j.compeleceng.2016.07.002).
- [26] H. Xu, M. Chen, Y. Zhou, B. Du y L. Pan, "A Novel Comprehensive Quality Index QoX and the Corresponding Context-aware System Framework," *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pp. 2415-2419, 2018. doi: [10.1109/CompComm.2018.8781007](https://doi.org/10.1109/CompComm.2018.8781007).
- [27] F. Mastrogiovanni, A. Sgorbissa y R. Zaccaria, "Distributed Context Assessment for Robots in Intelligent Environments," pp. 161-172, 2008. doi: [10.1007/978-3-642-00644-9\\_14](https://doi.org/10.1007/978-3-642-00644-9_14).
- [28] F. Mastrogiovanni, A. Sgorbissa y R. Zaccaria, "Context assessment strategies for Ubiquitous Robots," *2009 IEEE International Conference on Robotics and Automation*, pp. 2717-2722, 2009. doi: [10.1109/ROBOT.2009.5152568](https://doi.org/10.1109/ROBOT.2009.5152568).
- [29] S. Rizou, K. Häussermann, F. Dürr, N. Cipriani y K. Rothermel, "A System for Distributed Context Reasoning," *2010 Sixth International Conference on Autonomic and Autonomous Systems*, pp. 84-89, 2010. doi: [10.1109/ICAS.2010.21](https://doi.org/10.1109/ICAS.2010.21).
- [30] T. Halsted, O. Shorinwa, J. Yu y M. Schwager, "A Survey of Distributed Optimization Methods for Multi-Robot Systems," *ArXiv*, vol. abs/2103.12840, 2021.
- [31] I. Birs, C. Muresan, I. Nascu y C. Ionescu, "An Energy-Efficient Context Aware Solution for Environmental Assessment," *IFAC-PapersOnLine*, 2020. doi: [10.1016/J.IFACOL.2021.04.169](https://doi.org/10.1016/J.IFACOL.2021.04.169).
- [32] M. P. Polverini, A. Zanchettin y P. Rocco, "A computationally efficient safety assessment for collaborative robotics applications," *Robotics and Computer-integrated Manufacturing*, vol. 46, pp. 25-37, 2017. doi: [10.1016/J.RCIM.2016.11.002](https://doi.org/10.1016/J.RCIM.2016.11.002).
- [33] G. Wilson et al., "Robot-enabled support of daily activities in smart home environments," *Cognitive Systems Research*, vol. 54, pp. 258-272, 2019. doi: [10.1016/j.cogsys.2018.10.032](https://doi.org/10.1016/j.cogsys.2018.10.032).
- [34] S. Jatsun, N. Korenevskiy, S. Efimov y E. N. Korovin, "An Automated System for Monitoring the Environment and Assessing People's Status in Extreme Situations Using a Flying Robot," *Biomedical Engineering*, vol. 52, pp. 287-290, 2018. doi: [10.1007/S10527-018-9832-X](https://doi.org/10.1007/S10527-018-9832-X).
- [35] J. Wan, S. Tang, Q. Hua, D. Li, C. Liu y J. Lloret, "Context-Aware Cloud Robotics for Material Handling in Cognitive Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 5, pp. 2272-2281, 2018. doi: [10.1109/JIOT.2017.2728722](https://doi.org/10.1109/JIOT.2017.2728722).

- [36] M. Bahrin, M. Othman, N. Azli y M. F. Talib, "INDUSTRY 4.0: A REVIEW ON INDUSTRIAL AUTOMATION AND ROBOTIC," vol. 78, pp. 137-143, 2016. doi: [10.11113/JT.V78.9285](https://doi.org/10.11113/JT.V78.9285).
- [37] R. Goel y P. Gupta, "Robotics and Industry 4.0," pp. 157-169, 2019. doi: [10.1007/978-3-030-14544-6\\_9](https://doi.org/10.1007/978-3-030-14544-6_9).
- [38] M. Javaid, A. Haleem, R. Singh y R. Suman, "Substantial capabilities of robotics in enhancing industry 4.0 implementation," vol. 1, pp. 58-75, 2021. doi: [10.1016/J.COGR.2021.06.001](https://doi.org/10.1016/J.COGR.2021.06.001).
- [39] T. Bamigbala, M. Onkamo, I. Safonova y S. Rahman, "Towards Adoption of Autonomous Mobile Cobots in Intralogistics Picking Process: Review of Current Development," *Lappeenranta–Lahti University of Technology LUT: Lahti, Finland*, 2022.
- [40] J. Contreras, "Industrial Robots Migration Towards Industry 4.0 Components," pp. 1-12, 2019. doi: [10.1007/978-3-030-40309-6\\_1](https://doi.org/10.1007/978-3-030-40309-6_1).
- [41] M. Indri, A. Grau-Saldes y M. Ruderman, "Guest Editorial Special Section on Recent Trends and Developments in Industry 4.0 Motivated Robotic Solutions," *IEEE Trans. Ind. Informatics*, vol. 14, pp. 1677-1680, 2018. doi: [10.1109/TII.2018.2809000](https://doi.org/10.1109/TII.2018.2809000).
- [42] V. E. Hansen, "Using contextual sequences for the control of autonomous robots," *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell*, pp. 554-558, 1998. doi: [10.1109/ISIC.1998.713723](https://doi.org/10.1109/ISIC.1998.713723).
- [43] D. Calisi, L. Iocchi, D. Nardi, C. M. Scalzo y V. Ziparo, "Context-based design of robotic systems," *Robotics Auton. Syst.*, vol. 56, pp. 992-1003, 2008. doi: [10.1016/j.robot.2008.08.008](https://doi.org/10.1016/j.robot.2008.08.008).
- [44] A. Kupcsik, M. Deisenroth, J. Peters, A. Loh, P. Vadakkepat y G. Neumann, "Model-based contextual policy search for data-efficient generalization of robot skills," *Artif. Intell.*, vol. 247, pp. 415-439, 2017. doi: [10.1016/J.ARTINT.2014.11.005](https://doi.org/10.1016/J.ARTINT.2014.11.005).
- [45] M. R. Walter, S. Hemachandra, B. Homberg, S. Tellex y S. Teller, "A framework for learning semantic maps from grounded natural language descriptions," *The International Journal of Robotics Research*, vol. 33, pp. 1167-1190, 2014. doi: [10.1177/0278364914537359](https://doi.org/10.1177/0278364914537359).
- [46] W. Liu, A. Daruna y S. Chernova, "A survey of Semantic Reasoning frameworks for robotic systems," *Robotics Auton. Syst.*, vol. 159, p. 104 294, 2022. doi: [10.1016/j.robot.2022.104294](https://doi.org/10.1016/j.robot.2022.104294).
- [47] M. C. Tirado Robles, "¿ Qué es un robot? Análisis jurídico comparado de las propuestas japonesas y europeas," *inf. téc.*, 2020.

- [48] D. Kortenkamp, R. Simmons y D. Brugali, “Robotic systems architectures and programming,” *Springer handbook of robotics*, pp. 283-306, 2016.
- [49] N. Noguchi, J. Will, J. Reid y Q. Zhang, “Development of a master–slave robot system for farm operations,” *Computers and Electronics in agriculture*, vol. 44, n.º 1, pp. 1-19, 2004.
- [50] B. P. Gerkey y M. J. Matarić, “Murdoch: Publish/subscribe task allocation for heterogeneous agents,” en *Proceedings of the fourth international conference on Autonomous agents*, 2000, pp. 203-204.
- [51] O. Kubitz, M. O. Berger y R. Stenzel, “Client-server-based mobile robot control,” *IEEE/ASME transactions on mechatronics*, vol. 3, n.º 2, pp. 82-90, 1998.
- [52] J. S. Albus et al., *A reference model architecture for intelligent systems design*. Citeseer, 1994.
- [53] R. G. Simmons, “Structured control for autonomous robots,” *IEEE transactions on robotics and automation*, vol. 10, n.º 1, pp. 34-43, 1994.
- [54] F. Olsson, “A literature survey of active machine learning in the context of natural language processing,” 2009.
- [55] M. Gardner et al., “Allennlp: A deep semantic natural language processing platform,” *arXiv preprint arXiv:1803.07640*, 2018.
- [56] S. J. Russell y P. Norvig, *Artificial intelligence: a modern approach*. Pearson, 2016.
- [57] J. Speaks, “Theories of meaning,” 2010.
- [58] D. Kitchin, W. R. Cook y J. Misra, “A language for task orchestration and its semantic properties,” en *International Conference on Concurrency Theory*, Springer, 2006, pp. 477-491.
- [59] J. Fan, P. Zheng y S. Li, “Vision-based holistic scene understanding towards proactive human–robot collaboration,” *Robotics and Computer-Integrated Manufacturing*, vol. 75, p. 102 304, 2022.
- [60] X. Li, S. Du, G. Li y H. Li, “Integrate point-cloud segmentation with 3D LiDAR scan-matching for mobile robot localization and mapping,” *Sensors*, vol. 20, n.º 1, p. 237, 2019.
- [61] M. L. L. Mantecón, *Robots y personas: una aproximación jurídica a la subjetividad cibernética*. Editorial Reus, 2021.
- [62] Q. Zou, Q. Sun, L. Chen, B. Nie y Q. Li, “A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, n.º 7, pp. 6907-6921, 2021.
- [63] S. Yang, S. A. Scherer, X. Yi y A. Zell, “Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles,” *Robotics and Autonomous Systems*, vol. 93, pp. 116-134, 2017.

- [64] P. Picazo-Martínez, C. Barroso-Fernández, J. Martín-Pérez, M. Groshev y A. de la Oliva, “IEEE 802.11 az Indoor Positioning with mmWave,” *arXiv preprint arXiv:2303.05996*, 2023.
- [65] W. Garage, *TurtleBot*, 2024. [En línea]. Disponible en: <https://wiki.ros.org/Robots/TurtleBot>.
- [66] NVIDIA, P. Vingelmann y F. H. Fitzek, *CUDA, release: 10.2.89*, 2020. [En línea]. Disponible en: <https://developer.nvidia.com/cuda-toolkit>.
- [67] *Router Turrís Omnia 2020 Wi-Fi Dual b/g/n/ac, 5x GLAN, 1x SFP, 2x USB 3.0, 3x MiniPCI-e*, 2024. [En línea]. Disponible en: <https://wifisafe.com/producto/router-turris-omnia-2020-wi-fi-dual-b-g-n-ac-5x-glan-1x-sfp-2x-usb-3-0-3x-minipci-e/>.
- [68] M. Quigley et al., *ROS: an open-source Robot Operating System*, Available at <http://www.ros.org>, Accessed: 2024-05-20, 2009.
- [69] Z. Ge, S. Liu, F. Wang, Z. Li y J. Sun, “YOLOX: Exceeding YOLO Series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.
- [70] T. Lin et al., “Microsoft COCO: Common Objects in Context,” *CoRR*, vol. abs/1405.0312, 2014. arXiv: [1405.0312](https://arxiv.org/abs/1405.0312). [En línea]. Disponible en: <http://arxiv.org/abs/1405.0312>.
- [71] K. Eder, C. Harper y U. Leonards, “Towards the safety of human-in-the-loop robotics: Challenges and opportunities for safety assurance of robotic co-workers’,” en *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, IEEE, 2014, pp. 660-665.
- [72] Y. Ota, “Partner robots—From development to implementation,” en *3rd International Conference on Human System Interaction*, IEEE, 2010, pp. 14-16.
- [73] E. M. Wetzel, J. Liu, T. Leathem y A. Sattineni, “The use of boston dynamics SPOT in support of LiDAR scanning on active construction sites,” en *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, IAARC Publications, vol. 39, 2022, pp. 86-92.
- [74] J. Forlizzi y C. DiSalvo, “Service robots in the domestic environment: a study of the roomba vacuum in the home,” en *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, 2006, pp. 258-265.
- [75] J. Zhang, J. Hu, J. Lian, Z. Fan, X. Ouyang y W. Ye, “Seeing the forest from drones: Testing the potential of lightweight drones as a tool for long-term forest monitoring,” *Biological Conservation*, vol. 198, pp. 60-69, 2016.
- [76] L. Koren, T. Stipančić, A. Ricko y J. Benić, “Context-Driven Method in Realization of Optimized Human-Robot Interaction,” *Tehnički glasnik*, 2022. doi: [10.31803/tg-20220504100707](https://doi.org/10.31803/tg-20220504100707).

- [77] P. Chevalier, B. R. Schadenberg, A. Aly, A. Cangelosi y A. Tapus, "Context-Awareness in Human-Robot Interaction: Approaches and Challenges," *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 1241-1243, 2022. doi: [10.1109/HRI53351.2022.9889584](https://doi.org/10.1109/HRI53351.2022.9889584).
- [78] A. S. Baxi et al., "Towards Factory-Scale Edge Robotic Systems: Challenges and Research Directions," *IEEE Internet of Things Magazine*, vol. 5, pp. 26-31, 2022. doi: [10.1109/IOTM.001.2200056](https://doi.org/10.1109/IOTM.001.2200056).
- [79] K. E. Schaefer, J. Oh, D. Aksaray y D. Barber, "Integrating Context into Artificial Intelligence: Research from the Robotics Collaborative Technology Alliance," *AI Mag.*, vol. 40, pp. 28-40, 2019. doi: [10.1609/aimag.v40i3.2865](https://doi.org/10.1609/aimag.v40i3.2865).
- [80] C. Puligheddu, J. Ashdown, C. F. Chiasserini y F. Restuccia, "SEM-O-RAN: Semantic and flexible O-RAN slicing for NextG edge-assisted mobile systems," en *IEEE Infocom 2023-IEEE Conference on Computer Communications*, IEEE, 2023, pp. 1-10.
- [81] F. Mungari, C. Puligheddu, A. Garcia-Saavedra y C. F. Chiasserini, "OREO: O-RAN intelligence Orchestration of xApp-based network services," *arXiv preprint arXiv:2405.18198*, 2024.