



GOBIERNO DEL  
ESTADO DE MÉXICO

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE IXTAPALUCA  
ORGANISMO PÚBLICO DESCENTRALIZADO DEL ESTADO DE MÉXICO

**TES** I  
TECNOLÓGICO DE ESTUDIOS SUPERIORES  
IXTAPALUCA

## PROYECTO DE RESIDENCIA PROFESIONAL

DISEÑO Y SIMULACIÓN DE UN VEHÍCULO DE GUIADO AUTÓNOMO DE  
ACUERDO AL SISTEMA DE MANUFACTURA FLEXIBLE PARA LA INDUSTRIA 4.0.

INGENIERÍA ELECTRÓNICA

PRESENTA:

MATIAS ORTIZ EDSON BRASIL

ING. MAURICIO SÁNCHEZ IBARRA

ASESOR TÉCNICO

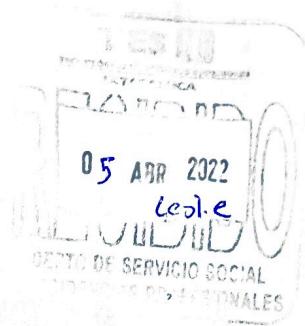
DR. JESÚS NOÉ RIVERA OLVERA

ASESOR METODOLÓGICO

MTRA. MARÍA ELENA OROZCO ÁLVAREZ

SUBDIRECTORA DE ESTUDIOS PROFESIONALES

Ixtapaluca, Estado de México, Marzo de 2022.



## **ÍNDICE**

**LISTA DE TABLAS Y FIGURAS.**

**INTRODUCCIÓN.**

**PLANTEAMIENTO DEL PROBLEMA.**

**OBJETIVO GENERAL.**

**OBJETIVO ESPECIFICO.**

**JUSTIFICACIÓN.**

### **1 MARCO REFERENCIAL.**

**1.1 Escuela Superior de Ingeniería Mecánica y Eléctrica.**

**1.1.1 Política de calidad**

**1.1.2 Misión**

**1.1.3 Visión**

**1.1.4 Ubicación y Contacto**

**1.1.5 Organigrama**

### **2 MARCO TEÓRICO.**

**2.1 NIVELES DE CONDUCCIÓN.**

**2.1.1 Nivel 0: Sin automatización de conducción.**

**2.1.2 Nivel 1: Asistencia al conductor.**

**2.1.2 Nivel 2: Automatización de conducción parcial.**

**2.1.3 Nivel 3: Automatización de conducción condicional.**

**2.1.4 Nivel 4: Alta automatización de conducción.**

**2.1.5 Nivel 5: Automatización de conducción completa.**

**2.2 VEHÍCULOS AUTÓNOMOS DE LAS GRANDES COMPAÑÍAS.**

**2.2.1 General Motors (GM).**

**2.2.2 Ford Motor Company.**

**2.2.3 Tesla Motors.**

**2.2.4 Waymo.**

**2.2.4.1 Waymo One.**

**2.2.4.2 Waymo Via.**

**2.2.4.3 Creación de mapas, intersecciones y semáforos de Waymo.**

**2.2.5 Toyota Motor Corporation.**

**2.2.6 NVIDIA, Samsung e Intel.**

**2.3 VEHÍCULOS AUTÓNOMOS Y SIMULADORES DE CÓDIGO ABIERTO.**

**2.3.1 F1/TENT.**

**2.3.2 CARLA.**

**2.3.3 SVL SIMULATOR.**

**2.3.4 QCar.**

### **3 ANTECEDENTES: VEHÍCULOS DE GUIADO AUTÓNOMO EN LA INDUSTRIA Y SUS APLICACIONES.**

**3.1 INTRODUCCIÓN.**

**3.5 Requerimientos de un sistema AGV.**

**3.2.1 Caracterización de instalaciones industriales y planificación de rutas.**

**3.2.2 Técnica de control del AGV en espacios cerrados.**

**3.2.3 Software para la conducción automatizada del AGV.**

**3.2.3.1 KogMo-RTDB.**

- 3.2.3.2** ADTF.
- 3.2.3.3** ROS.
- 3.2.4** Aplicaciones del AGV en otros entornos de trabajo.
- 3.2.5** Beneficios de un sistema AGV.

#### **4 SISTEMA DE MANUFACTURA FLEXIBLE.**

- 4.1 INTRODUCCIÓN**
- 4.2** ¿Qué es un Sistema de Manufactura Flexible?
- 4.3** Flexibilidad del sistema
- 4.4** Clasificación de los Sistemas de Manufactura
- 4.5** Esquema general del FMS.
- 4.6** Diseño del FMS.
- 4.7** Niveles de Flexibilidad.
- 4.8** Consideraciones para el manejo de materiales.
  - 4.8.1** Vehículos y transportes industriales para el manejo de materiales.
- 4.9** Transporte de materiales con AGV's.

#### **5 INDUSTRIA 4.0.**

- 5.1** Definición.
- 5.2** INICIOS.
- 5.3 TECNOLOGIAS.**
  - 5.3.1** Internet de las Cosas.
  - 5.3.2** Big Data.
  - 5.3.3** Inteligencia Artificial.
  - 5.3.4** Cloud Computing.
  - 5.3.5** Realidad Aumentada.
  - 5.3.6** Ciberseguridad.
  - 5.3.7** Simulación.
- 5.4** Industria 4.0 a nivel Mundial.
- 5.5** Industria 4.0 en Latinoamérica.

#### **6 INDUSTRIA 4.0 EN MÉXICO.**

- 6.1** Características del sector Manufacturero en México.
- 6.2** México hacia la Industria 4.0
- 6.3** Talento enfocado a la Industria 4.0 en México.

#### **7 DESARROLLO E IMPLEMENTACIÓN.**

- 7.1** VirtualBox.
- 7.2** Instalación de Ubuntu.
- 7.3** Instalación de ROS en Ubuntu.
  - 7.3.1** Configuración de Repositorios en Ubuntu.
  - 7.3.2** Configuración de llaves.
  - 7.3.3** Instalación.
  - 7.3.4** Configuración del entorno.
  - 7.3.5** Dependencias para construir paquetes.
  - 7.3.6** Inicializar rosdep.
  - 7.3.7** Nivel del sistema de archivos.
  - 7.3.8** Nivel del gráfico de computación.
  - 7.3.9** Nivel de la comunidad.
- 7.4** Sistema mecánico

**7.4.1** Vehículo a Escala.

**7.4.2** Suspensión.

**7.4.3** Dirección.

**7.4.4** Tracción.

#### **7.5 Electrónica y Eléctrica del Vehículo**

**7.5.1** Módulos.

#### **7.6 Integración por Computadora de Placa Única.**

**7.6.1** Raspberry Pi.

**7.6.2** Jetson Nano.

#### **7.7 Comunicación**

#### **7.8 Fuente de Energía**

#### **7.9 Percepción, localización y Mapeo**

**7.9.1** Cámara

**7.9.2** LiDAR

**7.9.3** SLAM

#### **7.7 Presupuesto**

**8** RESULTADOS.

**9** CONLUSIONES.

**10** TRABAJO FUTURO.

**11** REFERENCIAS.

## **LISTA DE TABLAS.**

- TABLA 2.1.-** Definición De los niveles de automatización de conducción. Elaboración propia.  
**TABLA 2.2.1.-** Piloto automático y capacidad de conducción autónoma total. Extraída de [25].  
**TABLA 2.2.-** Funciones destacadas de CARLA. Elaboración Propia.  
**TABLA 2.3.-** Versiones de SVL Simulator.  
**TABLA 4.7.-** Pruebas de Flexibilidad aplicadas a: Celda de una sola máquina, FMC y FMS [50].  
**TABLA 4.8.-** Características de los materiales para su manejo [52].  
**TABLA 7.7** Presupuesto de Amazon sin envío.

## **LISTA DE FIGURAS.**

- Figura 1.-** Revoluciones industriales y visión del futuro [3].  
**Figura 1.1.3.-** Mapa de calidad.  
**Figura 1.1.4.-** Ubicación de ESIME Unidad Azcapotzalco. Extraído de [13].  
**Figura 1.1.5.-** Organigrama de ESIME Unidad Azcapotzalco. Extraído de [14].  
**Figura 2.1.1.-** Estándar J3016TM “niveles de automatización de conducción”. Extraída de [16].  
**Figura 2.2.1.-** Mapeo detallado de la calle y objetos a su alrededor. Extraída de [21].  
**Figura 2.2.2.-** Exterior del Cruise Origin. Extraída de [21].  
**Figura 2.2.3.-** Funciones de conducción autónoma total. Extraída de [21].  
**Figura 2.2.4.-** Waymo One. Extraída de [27].  
**Figura 2.2.5.-** Waymo Via. Extraída de [28].  
**Figura 2.2.6.-** Disposición del sensor y sistema de coordenadas. Extraída de [29].  
**Figura 2.2.7.-** Escena de etiquetas 2D. Amarillo = vehículo. Rojo = peatón. Azul = signo. Rosa = ciclista. Extraída de [29].  
**Figura 2.2.8.-** Prototipo de LQ. Extraída de [30].  
**Figura 2.2.9.-** Toyota e-Palette en aplicaciones prácticas Autono-MaaS. Extraída de [31].  
**Figura 2.3.0.-** Modelado de permanencia del objeto. Extraída de [33].  
**Figura 2.3.1.-** Sitio web del curso para carreras autónomas F1/10 por la Universidad de Virginia [38].  
**Figura 2.3.2.-** Sitio web del Simulador de código abierto para la investigación de conducción autónoma: CARLA [40].  
**Figura 2.3.3.-** Sitio web de SVL Simulator [41].  
**Figura 2.3.4.-** Componentes principales de QCar [42].  
**Figura 2.3.5.-** Sitio web y productos de Quanser [43].  
**Figura 3.2.1.-** Mapa 3D (Beinschob 2017). Extraído de [8].  
**Figura 3.2.1.-** Prototipo de plataforma móvil hospitalaria para tareas de logística [9].  
**Figura 4.4.-** Clasificación de los Sistemas de Manufactura [50].  
**Figura 4.5.-** Celda de fabricación automatizada con dos máquinas CNC [50].  
**Figura 4.6.-** Esquema general de un FMS [51].  
**Figura 4.7.-** Celda de una sola máquina [50].  
**Figura 4.7.1.-** Celda de Manufactura Flexible (FMC) [50].  
**Figura 4.8.1.-** Transporte NO motorizado operado por el humano [50].  
**Figura 4.8.2.-** Transporte motorizado operado por el humano [50].  
**Figura 4.9.-** Tres tipos de sistemas de AGV [50].  
**Figura 6.1.** Tecnologías presentes en la i4.0 [54].

- Figura 6.2.-** Desempeño de México en el GCI, 2015-2016 [56].
- Figura 7.2.5.-** Creación de Disco Duro Virtual.
- Figura 7.2.4.-** Tamaño de Memoria RAM.
- Figura 7.2.7.-** Almacenamiento en unidad de disco duro física.
- Figura 7.2.6.-** Tipo de archivo del Disco Duro Virtual.
- Figura 7.2.8.-** Ubicación del archivo y tamaño del disco duro virtual.
- Figura 7.2.9.-** Previsualización de las configuraciones para Ubuntu 18.04 en VirtualBox.
- Figura 7.2.10.-** Almacenamiento de unidad óptica.
- Figura 7.2.11.-** Estado de Virtualización del equipo.
- Figura 7.2.12.-** Inicialización de Máquina Virtual.
- Figura 7.2.13.-** Pantalla de Bienvenida de Ubuntu.
- Figura 7.3.-** Pestaña de Software de Ubuntu.
- Figura 7.3.1.-** Comandos ejecutados de Configuración de llaves e índice de paquetes actualizado.
- Figura 7.3.1.-** Lista de paquetes actualizados, por actualizar y descargar.
- Figura 7.3.2.** Desempaquetado de librerías.
- Figura.- 7.3.3.-** Dependencias para construir paquetes
- Figura.- 7.3.4.-** Resumen de Instalación de ROS.
- Figura 7.6.1.** Perifericos de Raspberry Pi 4 [59].
- Figura 7.6.2.-** Jetson Nano Developer Kit [34]
- Figura 7.9.2.-** Sensor LiDAR Delta-2A de la empresa 3irobotix
- Figura 7.9.3.-** Vista inferior de Delta-2A con motor.
- Figura 7.9.4.-** Carpeta extraída de Delta-2A.
- Figura 7.9.5.-** Ejecución del archivo .exe.
- Figura 7.6.6.-** Primer escaneo del entorno con el sensor LiDAR Delta-2A.
- Figura 7.9.7.-** Vista isométrica del Sensor LiDAR y ordenador en proceso de mapeo.
- Figura 7.9.8.-** Vista isométrica de sensor LiDAR y módulo de control.
- Figura 7.9.9.-** Primera lectura del sensor sin el vehículo en movimiento.
- Figura 7.9.10.-** Lectura para generación de mapa 3D.
- Figura 8.1.1.-** Mapa 3D en SketchUp con los datos obtenidos del sensor LiDAR.
- Figura 8.1.2.-** Ruta de escaneo con sensor LiDAR en Gazebo.
- Figura 8.1.3.-** Localización de vehículo en mapa 3D.
- Figura 8.1.4.-** Rqt\_graph generado de Hector SLAM.
- Figura 8.1.5.-** Rqt\_tf\_tree generado para Hector SLAM ejecutando rosrun rqt\_tf\_tree rqt\_tf\_tree.

## INTRODUCCIÓN

A medida que la tecnología y la robótica aumentan, las industrias adoptan cada vez más la automatización para mejorar la calidad de las líneas de producción y la precisión para la administración del tiempo [1]. Las industrias no solamente han optado por adquirir tecnología, sino también facultad para procesar la información y los datos obtenidos en los procesos hacia los sistemas de administración, permitiendo monitorear en qué situación se encuentra la planta y habilitando gran parte de la maquinaria para el intercambio de información entre ellas, permitiendo una conexión en todos los procesos hacia lo que hoy se le conoce como la industria 4.0 [2]. En la Figura 1 se muestra la cuarta etapa de las tres revoluciones industriales predecesoras [3].

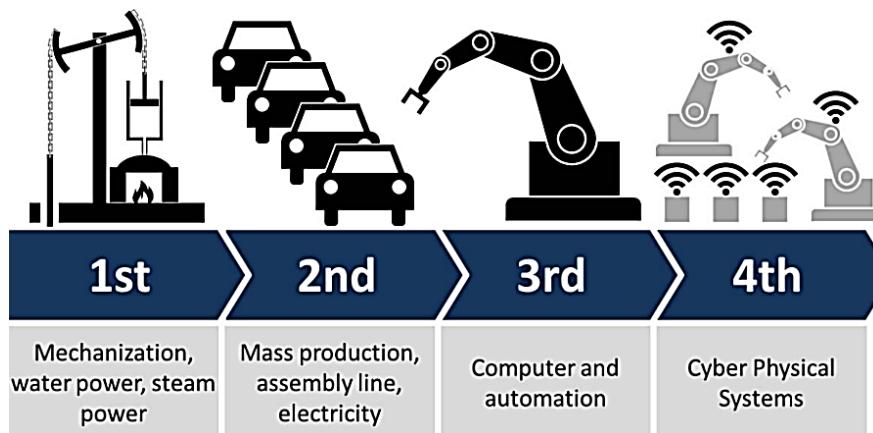


Figura 1.- Revoluciones industriales y visión del futuro [3].

Uno de los sistemas que ha tomado gran relevancia en la industria 4.0 y que permite una gestión de datos y procesos y comunicación, son los Vehículos de Guiado Autónomo (AGV) que han sido considerados en varias aplicaciones como; la logística industrial y el transporte de productos. Los vehículos de Guiado Autónomo son especiales en la robótica que realiza trabajos relacionados con el transporte, recuperación y ubicación de productos sin la intervención directa de humanos [4]. De acuerdo al entorno donde cada AGV se va a desempeñar se debe tomar en cuenta el diseño para la tarea específica que va a ejercer. Dicho de otra manera, es de suma importancia tener en cuenta los espacios y las características de los productos que van a ser transportados de un punto a otro. Más allá de estos aspectos de diseño se debe considerar que el AGV debe de estar conectado con otra maquinaria para el intercambio de datos con otros dispositivos y prepararlo para otros procesos a futuro (Industria 4.0). A causa de la complejidad de este sistema y sus aplicaciones, debe tomarse en cuenta aspectos como: números de vehículos que va a contener el sistema, diseño para el seguimiento de trayectorias, acondicionamiento del entorno de trabajo, tipo de material a transportar tomando en cuenta el punto de carga y descarga, uso de baterías y sistemas de seguridad para evitar colisiones que afecten al sistema y por consiguiente al él o los usuarios con quienes va a interactuar [5].

En este trabajo de investigación se diseña y simula un Vehículo de Guiado Autónomo para el transporte de mercancía y productos, preparándolo para la industria 4.0 y otros entornos.

## **PLANTEAMIENTO DEL PROBLEMA**

Una fábrica inteligente debe poder recopilar datos en tiempo real, almacenarlos, analizarlos y tomar decisiones de acuerdo con los nuevos hallazgos. Esto no solo se limita a cómo se vende el producto y la necesidad de stock, sino también a incidencias en los procesos de fabricación, como el fallo de una máquina en la línea de producción. Esto también contribuye a la flexibilidad y a la optimización de la producción.

Tal transformación requerirá una gran inversión en nuevas tecnologías, que enajena a las empresas más pequeñas y penalizará en el futuro su participación en el mercado.

Aunque todavía es pronto para especular sobre las condiciones de empleo con la adopción de la Industria 4.0 a nivel mundial, es seguro decir que los trabajadores necesitarán adquirir habilidades diferentes o un conjunto de habilidades completamente nuevo.

La base completa de la nueva industria se basa en que los dispositivos inteligentes pueden interactuar con el entorno circundante. Esto significa que los trabajadores que asisten en la producción serán despedidos y reemplazados por dispositivos inteligentes equipados con cámaras, sensores y actuadores. En consecuencia, la demanda de dichos trabajadores disminuirá y será reemplazada por coordinadores de robots.

# **OBJETIVOS**

## **OBJETIVO GENERAL**

Diseñar y simular un Vehículo de Guiado Autónomo que permita realizar maniobras individuales o cooperativas para el seguimiento de trayectorias y la coordinación de tareas con manejo de materiales en un entorno de un Sistema de Manufactura Flexible. Partiendo del estudio del modelo cinemático y adaptando técnicas de control de dirección y comunicación, con el objetivo de minimizar el error de orientación y posición del vehículo.

## **OBJETIVOS ESPECÍFICOS**

1. Operar sobre un meta sistema operativo de código abierto como: Robot Operating System (ROS)
2. Simular bajo leyes físicas para garantizar que el robot o el proceso simulado presente una alta confiabilidad al momento de implementarlo en el mundo real.
3. Comunicación flexible que pueda trabajar en armonía con diferentes dispositivos y contribuyan a un mejor desempeño del AGV.
4. Seleccionar un modelo de navegación que se ajuste fácilmente al vehículo y al ambiente de trabajo donde va a operar.
5. Diseñar e implementar un prototipo de pruebas para evaluar el sistema y control de navegación.
6. El AGV resultante de la unión de la simulación y el prototipo debe de ser capaz de navegar en un entorno dinámico para seguir exitosamente una trayectoria.

## **JUSTIFICACIÓN**

El robot industrial se ha visto favorecido notablemente por el avance de las técnicas de programación y control por ordenador, lo que supone una aportación decisiva a los procesos

- Aumentos de productividad debido a la alta velocidad y bajos tiempos de ciclo de producción, así como, la disminución de pasos en los procesos productivos.
- Mayor utilización de máquinas disponibles.
- Aumento de la flexibilidad productiva.
- Reducción de costes, disminución de piezas defectuosas, y fallos humanos, contribución al ahorro de materias primas, etc.
- Mejora de la calidad, como resultado de una mayor predictibilidad y repetitividad de los resultados de fabricación.
- Mayor calidad del trabajo de forma que las personas puedan dedicarse a tareas de valor añadido como la creatividad, tareas que implican cambios de respuesta, etc.

Sin embargo, un aspecto a considerar es que estos robots se compran en el extranjero. Esto implica un incremento en costos para las empresas porque el mantenimiento será costoso debido a que se tendrá que contratar mano de obra especializada. La solución a esta problemática es diseñar robots en México y capacitar ingenieros que den mantenimiento a los mismos.

Por otro lado el AGV ejecuta tareas específicas para el que fue programado y no necesita la intervención de un operador. Actualmente existen diferentes lenguajes de programación, y metodologías para programar robots, en este caso en particular AGV, que suelen ser sistemas complejos que demandan cierto grado de conocimiento en varios campos de la ingeniería debido a que son sistemas que integran diversos componentes de hardware, tanto mecánicos, eléctricos y electrónicos, convergiendo en un equipo que se debe desempeñar satisfactoriamente en un área o lugar de trabajo específico [6].

## **CAPÍTULO 1**

### **1 MARCO REFERENCIAL**

#### **1.1 Escuela Superior de Ingeniería Mecánica y Eléctrica**

##### **1.1.1 Política de calidad**

Nuestro compromiso es asegurar servicios educativos de calidad, mediante la oferta de programas académicos pertinentes, que logren la satisfacción de las y los alumnos, en su formación integral como profesionistas competentes, para contribuir al desarrollo científico, tecnológico, cultural y socioeconómico del país; sustentado en el cumplimiento de los requisitos normativos aplicables y en el mantenimiento de la mejora continua del SGC.

##### **1.1.2 Misión**

Formar integralmente profesionistas con calidad, capaces de ejercer el liderazgo en las áreas de Ingeniería Mecánica, Robótica Industrial, Sistemas Automotrices, Manufactura y Termofluidos, con una visión global, satisfaciendo las competencias que las industrias de manufactura, metal-mecánica y de transformación de la energía demandan con un enfoque sustentable, para contribuir con el desarrollo social y económico de México.

##### **1.1.3 Visión**

Unidad académica rectora y de vanguardia en las ramas de Ingeniería Mecánica, Robótica Industrial, Sistemas Automotrices, Manufactura y Termofluidos, que contribuye al desarrollo global, a través de sus procesos sustantivos de manera incluyente y transparente, con calidad y valores desde la perspectiva de la responsabilidad social.

## Mapa y Objetivos de Calidad [12].

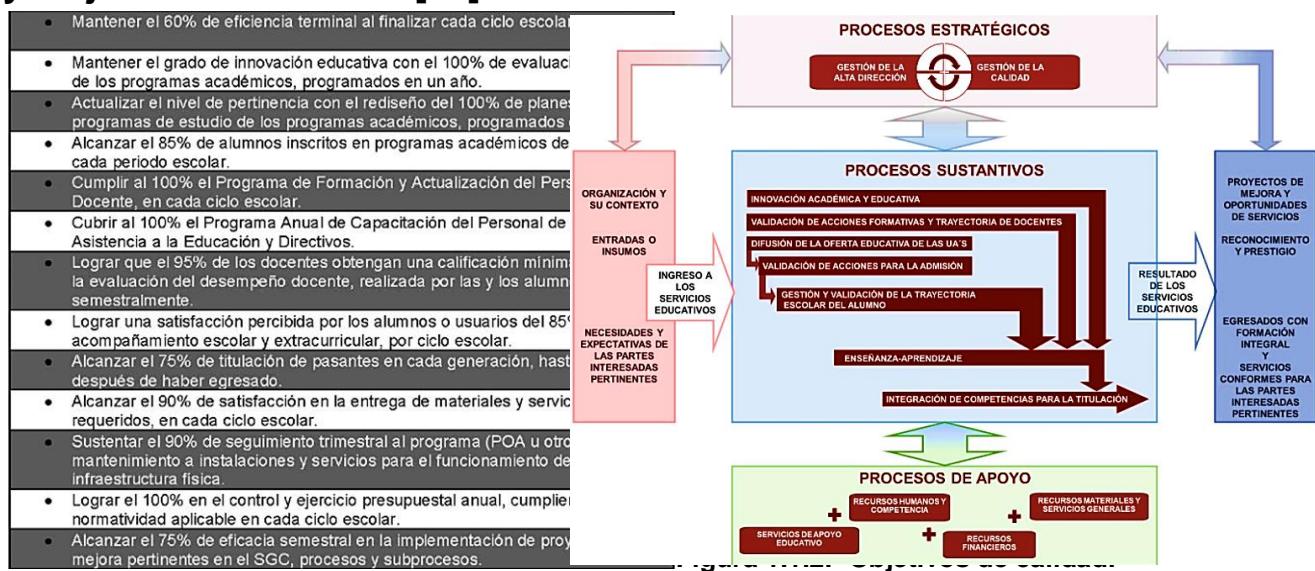


Figura 1.1.3.- Mapa de calidad.

### 1.1.4 Ubicación y Contacto

#### Dirección

Av. de las Granjas 682, Col. Santa Catarina, Delegación Azcapotzalco Ciudad de México, CDMX C.P. 02250

#### Teléfono

55 5729 6000

En la Figura 1.1.4 se muestra la ubicación en el Mapa de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Azcapotzalco.

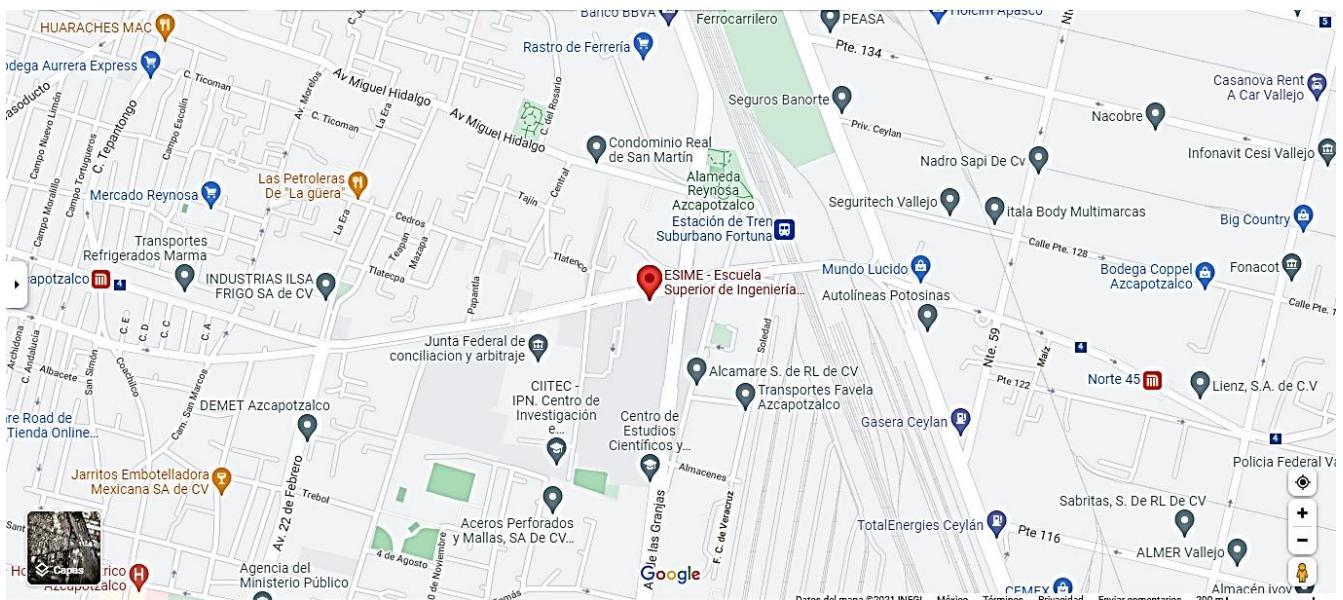


Figura 1.1.4.- Ubicación de ESIME Unidad Azcapotzalco. Extraído de [13].

### 1.1.5 Organigrama

En la Figura 1.5 se muestra el organigrama de la Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Azcapotzalco.

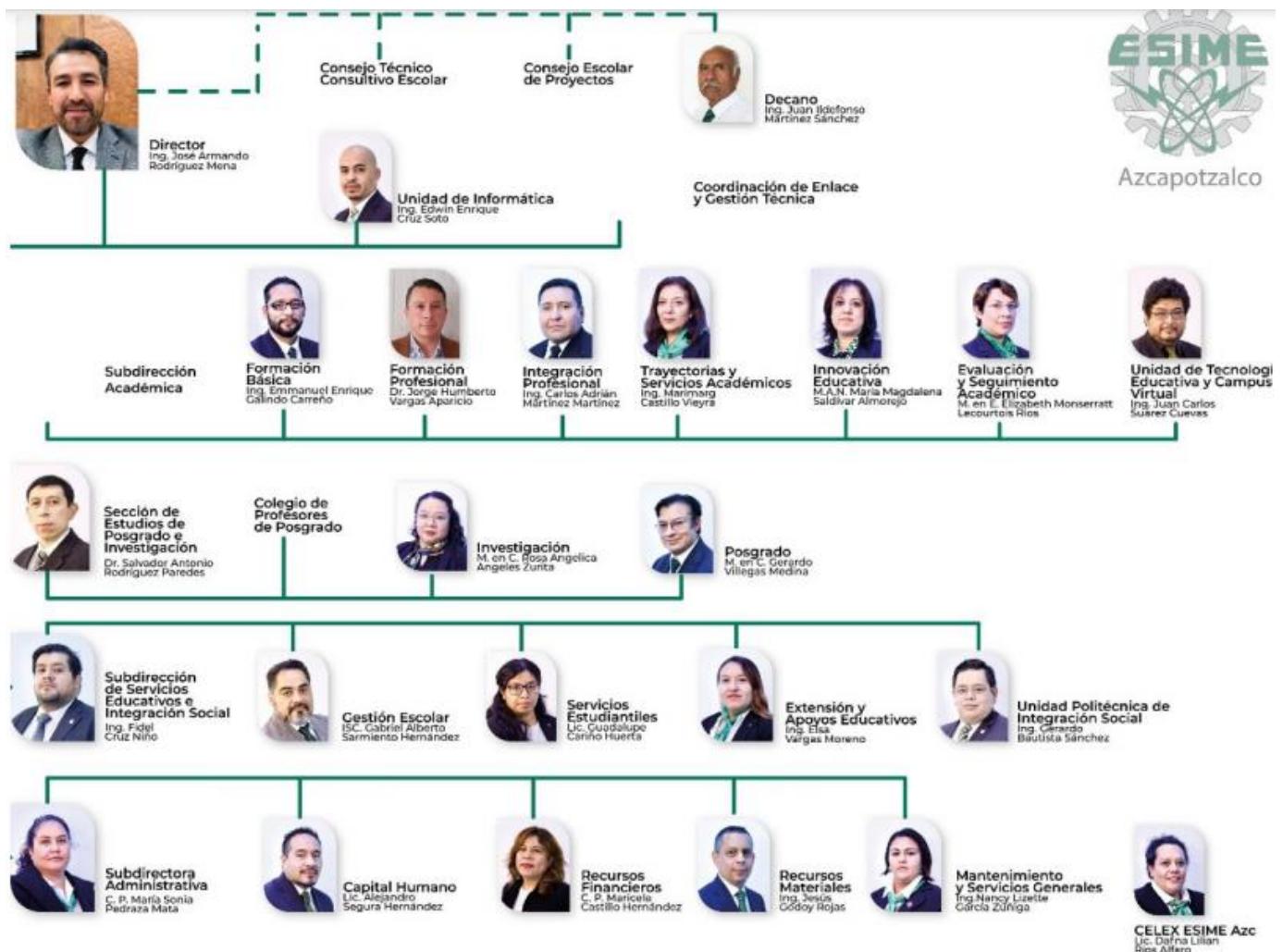


Figura 1.1.5.- Organigrama de ESIME Unidad Azcapotzalco. Extraído de [14].

## Capítulo 2

### 2 MARCO TEÓRICO

En este capítulo se presenta una revisión de los niveles de automatización de conducción y lo que las grandes compañías automovilísticas y tecnológicas están desarrollando en este

campo. Posteriormente se hace una revisión de los simuladores de código abierto más usados en vehículos autónomos.

## 2.1 NIVELES DE CONDUCCIÓN.

Según la SAE (*Society of Automobile Engineers*) [15] existen seis niveles de conducción en los vehículos autónomos, que va desde la ausencia de la automatización (nivel 0) hasta la automatización de la conducción completa (nivel 5).

Para tener un mayor panorama de los niveles de conducción, a continuación se explica cada uno de ellos (Tabla 2.1).

### 2.1.1 Nivel 0: Sin automatización de conducción.

El nivel 0 es el punto de partida, que es la que todos conocen por estar en la mayoría de los autos que hoy existen y que no cuentan con ningún tipo de automatización. Ya que el conductor está al mando.

### 2.1.2 Nivel 1: Asistencia al conductor.

En la actualidad gran parte de los coches comerciales están en este nivel de autonomía. Aquí se posicionan todos los vehículos que presentan algún sistema que les otorga una ayuda a la conducción sin estos sean muy sofisticados. Un ejemplo de estos sistemas es la alerta al cambio involuntario de carril.

La diferencia entre el nivel 0 y el nivel 1 es que este último hace cambios por su cuenta. Un ejemplo de esto son los sistemas de frenado de emergencia autónoma (AEB). En [17] explica que este sistema monitorea continuamente el camino por delante y aplica los frenos de forma autónoma si detecta algún obstáculo y no recibe información por parte del conductor.

### 2.1.2 Nivel 2: Automatización de conducción parcial.

En este nivel se cumplen determinadas condiciones. El sistema es capaz de ocuparse de forma continua tanto el movimiento lateral del vehículo (guiado) como el longitudinal (acelerar/decelerar). Tomando en cuenta que el conductor sigue teniendo la responsabilidad del resto de tareas de la conducción así como de monitorizar el entorno para responder ante un posible evento, asumiendo el control [18].

### 2.1.3 Nivel 3: Automatización de conducción condicional.

A partir de este nivel, el sistema ya es capaz de asumir por completo el control del vehículo, incluyendo el monitoreo de lo que hay a su alrededor. El conductor debe permanecer atento por si se le requiere que asuma el control debido a una falla del sistema.

#### **2.1.4 Nivel 4: Alta automatización de conducción**

El sistema se encarga por completo de la conducción, en determinadas condiciones, incluyendo la monitorización del entorno y sin esperar la intervención del conductor. El conductor puede tener la opción de controlar el vehículo [18].

#### **2.1.5 Nivel 5: Automatización de conducción completa**

El sistema se encarga por completo de la conducción, en todas las condiciones. No se necesitarán volantes ni pedales dado que el conductor no se involucra de ninguna manera.

**TABLA 2.1.- Definición De los niveles de automatización de conducción. Elaboración propia.**

Nivel	Definición
0	Sin automatización de conducción
1	Asistencia al conductor
2	Automatización de conducción parcial
3	Automatización de conducción condicional
4	Alta automatización de conducción
5	Automatización de conducción completa

En la Figura 2.1.1 se muestra el estándar J3016TM “niveles de automatización de conducción” por la SAE. La cual tiene como objetivo el acelerar la entrega de un marco regulatorio inicial para guiar a los fabricantes, investigadores y otras entidades para el diseño seguro y despliegue de vehículos altamente automatizados.



## SAE J3016™ LEVELS OF DRIVING AUTOMATION

	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?	You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety		You are not driving when these automated driving features are engaged – even if you are seated in "the driver's seat"		
What do these features do?	These are driver support features	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/acceleration support to the driver	These features provide steering AND brake/acceleration support to the driver	These are automated driving features	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met
Example Features	*automatic emergency braking *blind spot warning *lane departure warning	*lane centering OR *adaptive cruise control	*lane centering AND *adaptive cruise control at the same time	*traffic jam chauffeur	*local driverless taxi *pedals/steering wheel may or may not be installed	*same as level 4, but feature can drive everywhere in all conditions

Figura 2.1.1.- Estándar J3016TM “niveles de automatización de conducción”. Extraída de [16].

En [16] detalla que el Departamento de Transporte de EE.UU. (DoT) utiliza los seis niveles de automatización de J3016 para vehículos de motor en carretera en su "Política federal de vehículos automatizados" y el documento se convirtió en un estándar global de facto adoptado por las partes interesadas en la tecnología de vehículos automatizados.

## 2.2 VEHÍCULOS AUTÓNOMOS DE LAS GRANDES COMPAÑÍAS.

### 2.2.1 General Motors (GM)

En 2016, General Motors adquirió *Cruise Automation*. Con esto, GM se introduce al negocio de la conducción autónoma con su primera generación de vehículos autónomos construidos a medida sobre el Chevrolet Bolt, siendo este el *Cruise AV*. Principalmente cuenta con más de 40 sensores, los cuales brindan una vista de 360° para el vehículo a lo largo y ancho, y pueden hacer un mapeo de objetos que se encuentran circundantes en centímetros. Así mismo, los vehículos analizan y evalúan los movimientos complejos de las calles con mayor tránsito de las ciudades con gran eficiencia.

La actuación rápida detecta, predice, y responde al movimiento de personas, animales y objetos más rápido que cualquier cerebro humano. En la Figura 2.2.1 se muestra un mapeo detallado del funcionamiento de los sensores del vehículo *Cruise AV* y se muestra como detecta aún los vehículos estacionados en doble fila. Esta es la primera generación y ya ha recorrido millones de millas por las ciudades para hacer realidad la conducción autónoma. Tienen todo lo que necesitan para conducir directamente a bordo: el 40% de su hardware es exclusivo para la conducción autónoma [21].



Figura 2.2.1.- Mapeo detallado de la calle y objetos a su alrededor. Extraída de [21].

Por otro lado, después de 4 años de usar el Chevrolet Bolt modificado para probar sus sistemas autónomos, la empresa *Origin* presentó su nuevo vehículo autónomo diseñado especialmente para transporte exclusivo el cual tiene una similitud a una camioneta y es totalmente eléctrico,

pero carece de volante, pedales tradicionales y espejos, lo cual lo hace un diseño exclusivo para pasajeros (Figura 2.1.2).

Este vehículo autónomo llamado *Origin* tendrá un software AV central que puede superar a un conductor humano promedio en términos de rendimiento y seguridad. Además, el vehículo está diseñado para durar un millón de millas, con numerosas piezas reemplazables, desde componentes interiores hasta sensores y la computadora a bordo, lo que reduce el costo promedio por milla [19].

La eliminación del asiento del conductor y el tablero de instrumentos convencional permitió a *Cruise* proporcionar un amplio espacio en la cabina para seis pasajeros, con dos filas de 3 asientos enfrentadas. "Cada asiento es como un asiento con más espacio para las piernas en un avión", dijo Kyle Vogt, presidente y director de tecnología de *Cruise*. Dijo que el *Origin* proporciona tres veces más área para entrar y salir que un vehículo convencional de tamaño similar [23].



Figura 2.2.2.- Exterior del Cruise Orgin. Extraída de [21].

## 2.2.2 Ford Motor Company

Ford adquirió *Quantum Signal*, empresa que se ha caracterizado por su constante innovación en el desarrollo de modelos de navegación autónoma virtual, tecnología que ha sido usada por el programa de robótica del ejército de los Estados Unidos. Ford implementará Tecnologías de *Quantum Signal* en el Desarrollo de Vehículos Autónomos

Con la integración de la experiencia de *Quantum Signal*, Ford busca seguir desarrollando tecnologías que le permitan mejorar la capacidad de los vehículos autónomos de analizar el entorno que les rodea y que mejorar la movilidad de las personas alrededor del mundo en el futuro [20].

*Quantum Signal* es reconocida por su entorno de simulación y modelado ANVEL, el cual ha sido empleado para la exploración de los sistemas remotos no tripulados y autónomos. Ha desarrollado e implementado algoritmos para el guiado de vehículos autónomos militares Por

otro lado, la empresa también es conocida por su conocimiento en el campo de la robótica así como en sistemas de detección y percepción [24].

### 2.2.3 Tesla Motors

Los autos Tesla vienen de fábrica con un avanzado hardware que puede proporcionar funciones de *Autopilot*, y capacidad de conducción autónoma total, a través de actualizaciones de software diseñadas para mejorar la funcionalidad a medida que pasa el tiempo.

Ocho cámaras periféricas brindan 360 grados de visibilidad alrededor del auto con un alcance de hasta 250 metros. Doce sensores ultrasónicos actualizados complementan esta visión, lo que permite la detección de objetos duros y blandos a casi el doble de distancia comparado con el sistema anterior.

La función “Navegar” en *Autopilot* sugiere cambios de carril para optimizar su ruta y realiza ajustes para que no quede atascado detrás de autos o camiones que se desplazan lento. Cuando está activa, la función Navegar en *Autopilot* también dirige de forma automática el vehículo hacia enlaces y salidas de carreteras según su destino [21].

Aunque Tesla ofrece la conducción autónoma total en sus vehículos, las características de seguridad están diseñadas para ayudar a los conductores, pero no pueden responder a todas las situaciones. La empresa dice que las funciones actualmente habilitadas requieren una supervisión activa del ser humano y esto hace que el vehículo no sea autónomo (Figura 2.2.3). La autonomía dependerá del desempeño del sistema, ya que se requiere que sea de mayor confiabilidad a la conducción del ser humano.



Figura 2.2.3.- Funciones de conducción autónoma total. Extraída de [21].

Sin embargo, no todos los modelos traen este sistema, pero se puede actualizar conforme el cliente lo vaya solicitando para brindar una nueva funcionalidad a su Tesla y sea menos estresante y con mayor seguridad para su manejo. Los paquetes disponibles se pueden apreciar en la Tabla 2.2.1.

**TABLA 2.2.1.- Piloto automático y capacidad de conducción autónoma total. Extraída de [25].**

<b>Piloto Automático</b>	<b>Control de crucero consciente del tráfico :</b> Hace coincidir la velocidad de su automóvil con la del tráfico circundante <b>Dirección automática :</b> Ayuda a conducir dentro de un carril claramente marcado y utiliza el control de crucero consciente del tráfico
<b>Capacidad de Conducción Autónoma Total</b>	<b>Navegar en piloto automático (Beta) :</b> Guía activamente su automóvil desde la rampa de entrada a la rampa de salida de una autopista, lo que incluye sugerir cambios de carril, navegar por los intercambios, activar automáticamente la señal de giro y tomar la salida correcta <b>Cambio de carril automático:</b> Ayuda a moverse a un carril adyacente en la autopista cuando el <i>autotester</i> está activado. <b>Autopark:</b> Ayuda a estacionar automáticamente su automóvil en paralelo o perpendicular, con un solo toque. <b>Summon:</b> Mueve tu auto dentro y fuera de un espacio reducido usando la aplicación móvil o la llave. <b>Invocación inteligente:</b> Su automóvil navegará por entornos y espacios de estacionamiento más complejos, maniobrando alrededor de los objetos según sea necesario para encontrarlo en un estacionamiento. <b>Control de señales de tráfico y de alto (Beta) :</b> Identifica las señales de alto y los semáforos y reduce automáticamente la velocidad de su automóvil al detenerse al acercarse, con su supervisión activa

## 2.2.4 Waymo

La empresa *Alphabet Inc.* Es una matriz de Google que lleva desarrollando un sistema de conducción autónoma desde hace ya varios años. *Waymo* comenzó como un proyecto de Google en el 2009 y desde entonces ha estado trabajando en una tecnología de conducción autónoma, a fin de brindar a las personas una nueva libertad: ir a donde quieran y cuando lo deseen sin preocuparse ni frustrarse por tener que conducir [26]. *Waymo Driver* es el sistema de conducción más experimentado del mundo, prueba de ello son los kilómetros que ha recorrido en varios escenarios que le han permitido recolectar una gran cantidad de datos y poder aprender de ellos para ser la empresa más avanzada en poder desarrollar tecnología de conducción autónoma más que cualquier otra empresa. Por otro lado *Waymo Driver* se puede implementar desde vehículos y camionetas hasta camiones tipo 8, así como en varios tipos de servicios y experiencias, como viajes, transporte y entregas locales.

### 2.2.4.1 Waymo One

La visión de *Waymo One* es que ningún ser humano conduzca sus vehículos, una experiencia que puedes vivir de primera mano en el área metropolitana de *Phoenix*. Con tan solo presionar un botón, un agente de Asistencia al pasajero estará disponible en caso de que el pasajero tenga una pregunta o requiera alguna asistencia [27].



Figura 2.2.4.- *Waymo One*. Extraída de [27].

#### **2.2.4.2 Waymo Via**

Se dedica a transportar bienes comerciales. Su solución de conducción autónoma está diseñada para mejorar la seguridad y aumentar la eficiencia, mientras que su solución de entrega local fue creada para revolucionar la experiencia de tramo final [26].



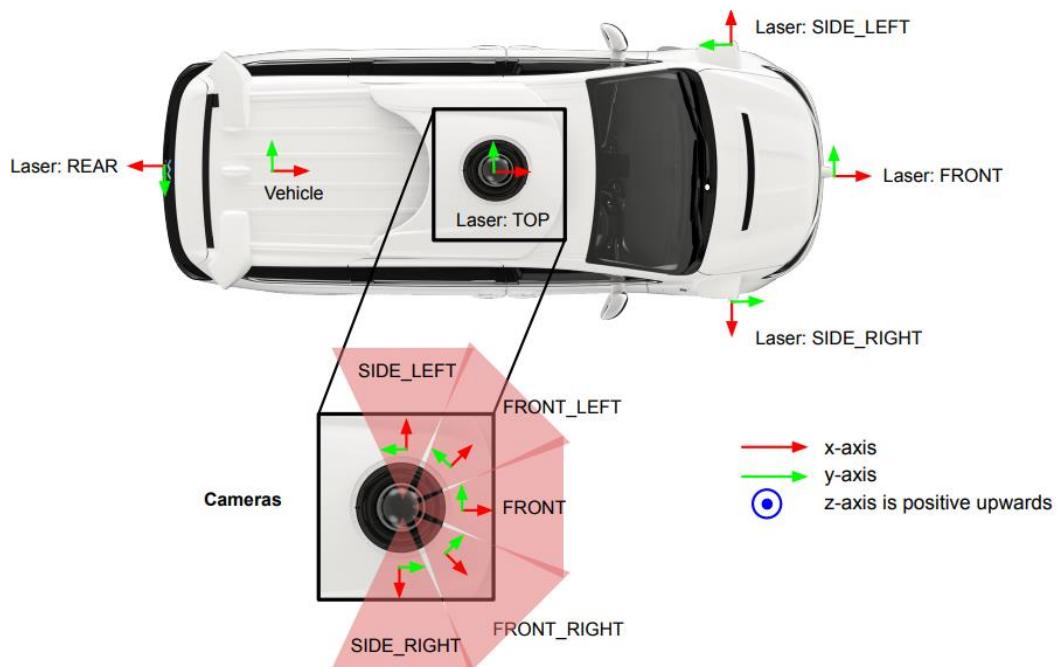
**Figura 2.2.5.- Waymo Via. Extraída de [28].**

En 2019 se lanzó por primera vez el *Waymo Open Dataset*, el cual contiene datos de sensores de alta resolución. Para 2021 *Waymo Open Dataset* incluyó un conjunto de datos de movimiento que comprende trayectorias de objetos y sus correspondientes mapas 3D. Esto permitirá a los investigadores a realizar avances en la percepción de las máquinas y la tecnología de conducción autónoma.

#### **2.2.4.3 Creación de mapas, intersecciones y semáforos de Waymo**

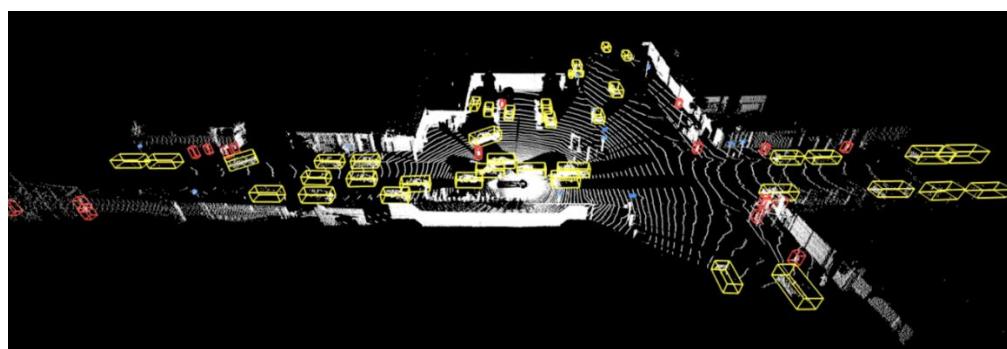
Para que el sistema empiece a operar en un nuevo entorno real, se tiene que generar un mapa con gran nivel de precisión y detalle, esto incluye desde las líneas de los carriles hasta señalamientos de paro, bordes y aceras. Después de basarse solo en los datos externos como los que se reciben por GPS (que pueden perder intensidad de la señal), Waymo utiliza estos mapas realizados extremadamente a detalle, y los combina con los datos que se reciben del sensor en tiempo real para determinar la posición exacta del vehículo en todo momento. Por otro lado el sistema de percepción está atento en todo momento, tomando todos los datos que recopilaron sus sensores avanzados, y analiza todo lo que lo rodea y hace uso de una tecnología que es muy similar al aprendizaje automático, estos datos pueden ser desde ciclistas, peatones, vehículos, edificios, etc. Además también actúa sobre las señales e indicadores de su entorno, como semáforos e

indicaciones de paro. En la escalabilidad de la percepción para la conducción autónoma del conjunto de datos abiertos de Waymo disponible en [29], proporciona los datos que se han obtenido a través de sus cinco sensores *LiDAR* y por otro lado, cinco cámaras estenopeicas de alta resolución. Las imágenes son capturadas por un obturador rodante, donde el modo de grabación puede tener una variación entre escenas. Los tamaños reflejan los resultados de cortar y reducir el muestreo de los datos originales del sensor. El campo de visión horizontal de la cámara es proporcionado como un rango de ángulo en el eje *x* en el plano *x – y* de la cámara (Figura 2.2.6).



**Figura 2.2.6.-Disposición del sensor y sistema de coordenadas. Extraída de [29].**

También en La Figura 2.2.7 se describen los objetos que son etiquetados con la ayuda del sensor *LiDAR* y la cámara, haciendo cada componente su trabajo por separado en la cual se define el largo, ancho y alto como los tamaños a lo largo del eje *x, y, z* respectivamente. se etiquetan exhaustivamente vehículos, peatones, letreros y ciclistas con las lecturas que se obtienen con el sensor *LiDAR*. En la (Figura 2.2.7) se puede apreciar un ejemplo de una escena.



**Figura 2.2.7.- Escena de etiquetas 2D. Amarillo = vehículo. Rojo = peatón. Azul = signo. Rosa = ciclista. Extraída de [29].**

## 2.2.5 Toyota Motor Corporation

Como parte del sistema de conducción autónoma de *Toyota Motor Corporation*, *Toyota Safety Sense* es una tecnología puntera en el mercado que ofrece cierto grado de autonomía. Ayuda a mantener en orden las situaciones altamente peligrosas para el conductor y terceras personas. La alerta de cambio involuntario de carril se asiste de una cámara y sensores para detectar los vehículos que hay por delante avisando si el vehículo se desplaza involuntariamente a otro carril. Como esta y otras alertas que incorpora el sistema, solamente son una ayuda para el conductor, y no están diseñados para sustituirle. Por esto mismo el conductor debe de estar alerta en todo momento ya que, el sistema puede verse afectado por factores externos.

Dentro de los proyectos de Toyota, existe el vehículo *LQ* que no solo está equipado con una función de conducción automatizada equivalente al nivel 4 SAE, sino que también tiene un sistema automatizado de *parking* proporcionando accesibilidad y flexibilidad. Con este vehículo, Toyota busca entablar un vínculo emocional entre el conductor y el vehículo bajo el propósito de desarrollo: aprender, crecer y amar. *LQ* está equipado con un sistema de conducción automatizada y con un poderoso agente llamado “*Yui*” que está diseñado para aprender de su conductor y ofrecerle una experiencia de movilidad personalizada.



**Figura 2.2.8.- Prototipo de LQ. Extraída de [30].**

El primer *BEV* de Toyota desarrollado específicamente para *Autono-MaaS* (*Autonomous Mobility as a Service*). En los Juegos Olímpicos y Paralímpicos, *e-Palette* respaldará las necesidades de transporte del personal y de los atletas con una docena o más circulando en un circuito continuo dentro de la Villa Olímpica y Paralímpica (Figura 2.2.9).

Como plataforma de servicio de movilidad, *Toyota* se enfoca en crear características nuevas y atractivas para los socios que les ayudan a expandir su cadena de valor y mejorar la vida de los clientes. *E-Palette* es una de las movilidades sociales para realizar la visión. *Toyota* tiene como objetivo hacer esto colaborando con otros proveedores de servicios y compañías de desarrollo tecnológico [31].



**Figura 2.2.9.- Toyota e-Palette en aplicaciones prácticas Autono-MaaS. Extraída de [31].**

Por otra parte el *TRI* (*Toyota Research Institute*) comparte en [32] las investigaciones y parte de la tecnología que llevan estos vehículos, en la cual destaca el aprendizaje para rastrear con permanencia de objetos. Para tener un panorama a lo que esto se refiere, se presenta una situación que se ve en numerosas ocasiones en las calles: una persona caminando detrás del vehículo en movimiento de momento está en un punto visible, pero de un momento a otro ya no lo está (Figura 2.2.9). Sin embargo no hay respuesta de que el hombre esté todavía ahí, y por ende no se puede definir su ubicación en el momento. Esta capacidad se le conoce en el mundo de los científicos cognitivos como permanencia del objeto. En [33] se propone el aprendizaje profundo para el seguimiento de objetos múltiples que ayuda a entender este tipo de razonamientos.

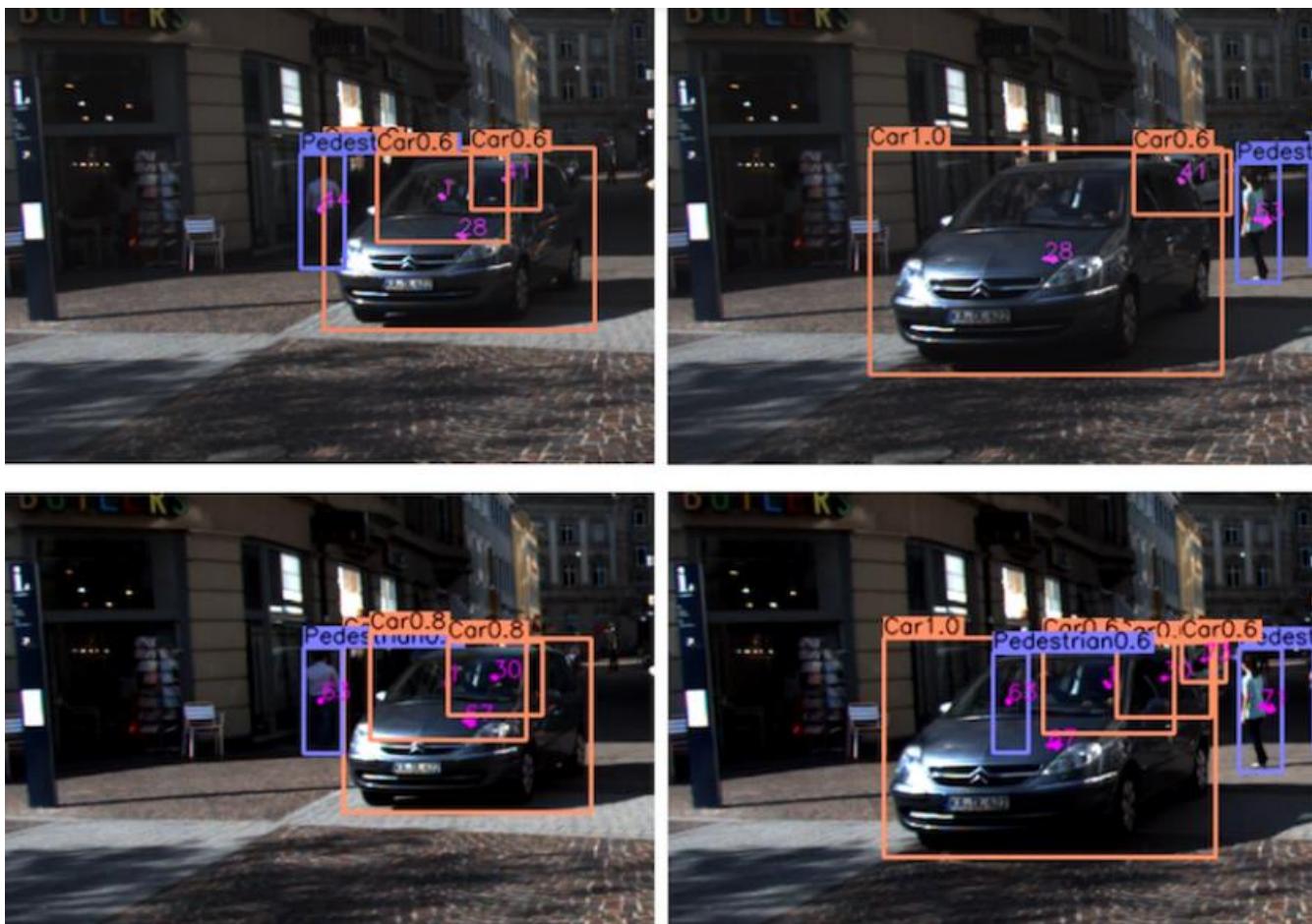


Figura 2.3.0.- Modelado de permanencia del objeto. Extraída de [33].

## 2.2.6 NVIDIA, Samsung e Intel.

Aunque estas empresas no sean de la industria automotriz, están sumándose al desarrollo de vehículos autónomos, mismos que requieren de una potencia computacional masiva y una experiencia de software de producción a gran escala. Una de ellas es **NVIDIA Corporation**, que con la experiencia por varias décadas en el campo de la computación, imágenes e inteligencia artificial de alto rendimiento, ha creado una plataforma definida por software para la industria del transporte y que permite la mejora continua a través de actualizaciones inalámbricas la cual ofrece todo lo necesario para el desarrollo de vehículos autónomos a escala. **NVIDIA DRIVE** procesa datos de cámaras, radares y sensores *LiDAR* para percibir el entorno circundante, localizar el vehículo en un mapa y luego planificar y ejecutar un camino seguro hacia adelante. Esta plataforma de inteligencia artificial admite la conducción autónoma, las funciones en la cabina y la supervisión del conductor, además de otras características de seguridad, todo en un paquete compacto y de bajo consumo [34].

Por otro lado, **Samsung Electronics Co. Ltd**, no entrará en el negocio de fabricación de vehículos, pero hace su aparición en este campo con su software y hardware de conducción autónoma, el cual ya ha obtenido licencias para pruebas en ruta, dicho proyecto está

actualmente en desarrollo en Corea y en California. También la empresa líder mundial en tecnología creó el *Samsung Automotive Innovation Fund*, un nuevo fondo de 300 millones de dólares enfocado exclusivamente en el mercado automotriz. La iniciativa tiene como objetivo el desarrollo de tecnologías para automóviles conectados y autónomos, incluyendo sensores inteligentes, inteligencia artificial, computación de alto desempeño, soluciones de conectividad, soluciones de seguridad, protección y privacidad [35]. Con esto deja claro la empresa que se mantendrá enfocada en trabajar en asociación con los fabricantes de automóviles, así como con los facilitadores de movilidad para desarrollar la próxima generación de la innovación automotriz.

**Intel Corporation** no se quiere quedar fuera en el desarrollo de vehículos autónomos, por lo cual ha hecho una gran inversión en la empresa *Mobileye* al adquirirla por 15.000 millones de dólares, quien es la desarrolladora del *Autopilot* de Tesla. La apuesta de Intel no solo es por los vehículos sino también por el software. Esta adquisición es una de las más exitosas en la historia de Intel, *Mobileye* es un motor de crecimiento para *INTC*. Los ingresos se han más que duplicado desde la adquisición y financian completamente el trabajo de desarrollo de *Mobileye* sobre el futuro autónomo. Único en la industria, el modelo comercial de *Mobileye* abarca toda la cadena de valor de la conducción automatizada, incluida la cámara frontal que impulsa la mayoría de los sistemas avanzados de asistencia al conductor (ADAS) actuales, la autonomía condicional, también conocida como nivel 2+, y el autocontrol. Sistema de conducción (SDS) para robotaxis y vehículos autónomos de consumo (AV). *Mobileye* es líder en cada una de estas categorías con la tecnología de detección de visión más avanzada de la industria, la capacidad de mapeo de fuentes múltiples y la política de conducción de seguridad sensible a la responsabilidad (RSS) [36].

## 2.3 VEHÍCULOS AUTÓNOMOS Y SIMULADORES DE CÓDIGO ABIERTO.

### 2.3.1 F1/TENT

F1TENTH es una comunidad internacional que está conformada por investigadores, ingenieros y entusiastas de los sistemas autónomos. Fue fundada originalmente en la Universidad de Pensilvania en 2016, pero desde entonces se ha extendido a muchas otras instituciones en todo el mundo como: Universidad de Toronto, Universidad de Virginia, Instituto de Tecnología de Massachusetts, Universidad Politécnica de Hong Kong por mencionar algunas.

Esta comunidad fue creada principalmente porque los sistemas autónomos requieren de plataformas sólidas en las que los investigadores puedan realizar pruebas y experimentos para saber cómo se van a comportar los sistemas en el mundo real. Por otro lado, la gran mayoría de los experimentos se realizan de forma aislada, ya sea completamente en algún software de simulación o en un hardware patentado. Las limitantes vienen cuando los investigadores no pueden emplear sus metodologías en un entorno realista sin resolver una serie de problemas. Por el contrario, las plataformas de código abierto son de gran ayuda para diferentes campos y múltiples grupos de investigadores, esto puede ser un impulso para permitir la enseñanza de alto impacto. En [37] se presenta F1/10: un banco de pruebas de vehículos autónomos a escala 1/10 de código abierto, asequible y de alto rendimiento. El banco de pruebas F1/10 incluye un conjunto completo de pilas de software de sensores, percepción, planificación, control y redes que son similares a las soluciones a gran escala.

En la Figura 2.3.1 se muestra la página principal del curso que imparte la Universidad de Virginia, en el cual los estudiantes trabajan en equipos para construir, conducir y correr autos de carrera autónomos a escala 1/10, mientras aprenden sobre los principios de percepción, planificación y control de vehículos autónomos. El curso culmina con una carrera de "batalla de algoritmos" de F1/10 entre los equipos.

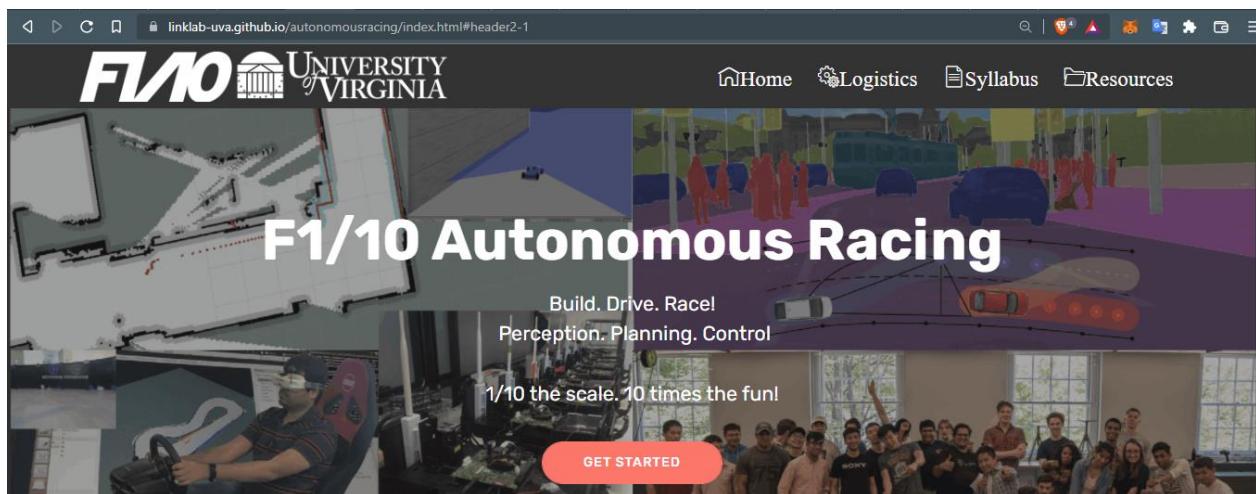


Figura 2.3.1.- Sitio web del curso para carreras autónomas F1/10 por la Universidad de Virginia [38].

### 2.3.2 CARLA

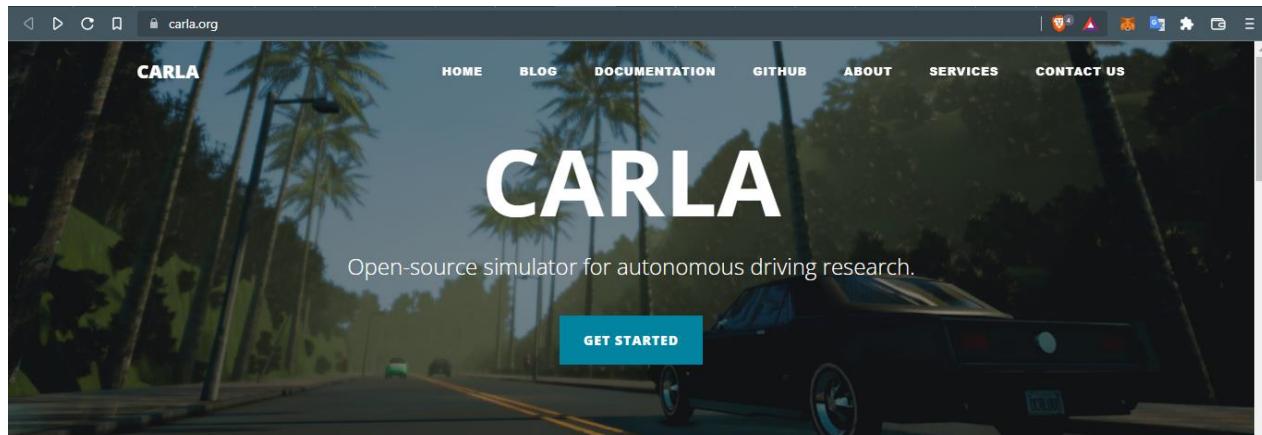
CARLA es un simulador de código abierto para la investigación de conducción autónoma fundado por *Intel Labs* y *Computer Vision Center* y dentro de sus patrocinadores está *Toyota Research Institute* y *Futurewei Technologies*. CARLA se ha desarrollado desde cero para apoyar el desarrollo, la formación y la validación de los sistemas de conducción autónoma. Además del código y los protocolos de fuente abierta, CARLA proporciona activos digitales abiertos (trazados urbanos, edificios, vehículos) que se crearon para este fin y se pueden utilizar libremente. La plataforma de simulación admite la especificación flexible de conjuntos de sensores, condiciones ambientales, control total de todos los actores estáticos y dinámicos, generación de mapas y mucho más.

En [39] se presenta la plataforma de simulación que admite la especificación flexible de conjuntos de sensores y condiciones ambientales. También menciona que se puede utilizar CARLA para estudiar el rendimiento de tres enfoques de la conducción autónoma: un canal modular clásico, un modelo de extremo a extremo entrenado a través del aprendizaje por imitación y un modelo de extremo a extremo entrenado a través del aprendizaje por refuerzo. Los enfoques se evalúan en escenarios controlados de dificultad creciente y su rendimiento se examina a través de métricas proporcionadas por CARLA, lo que ilustra la utilidad de la plataforma para la investigación de conducción autónoma. En la Figura 2.3.2 se pude ver la página principal del simulador CARLA y en la Tabla 2.2 se describen las funciones más destacadas por el desarrollador.

**TABLA 2.2.- Funciones destacadas de CARLA. Elaboración Propia.**

FUNCIONES DESTACADAS	
<b>Escalabilidad a través de una arquitectura servidor multicliente</b>	Múltiples clientes en el mismo o en diferentes nodos pueden controlar diferentes actores.
<b>API flexible</b>	CARLA expone una potente API que permite a los usuarios controlar todos los aspectos relacionados con la simulación, incluida la generación de tráfico, el comportamiento de los peatones, el clima, los sensores y mucho más.
<b>Conjunto de sensores de conducción autónoma</b>	Los usuarios pueden configurar diversos conjuntos de sensores, incluidos LIDAR, cámaras múltiples, sensores de profundidad y GPS, entre otros.
<b>Simulación rápida para planificación y control</b>	Este modo desactiva el renderizado para ofrecer una ejecución rápida de simulación de tráfico y comportamientos viales para los que no se requieren gráficos.

<b>Generación de mapas</b>	Los usuarios pueden crear fácilmente sus propios mapas siguiendo el estándar <i>OpenDrive</i> a través de herramientas como <i>RoadRunner</i>
<b>Simulación de escenarios de tráfico</b>	El motor <i>ScenarioRunner</i> permite a los usuarios definir y ejecutar diferentes situaciones de tráfico basadas en comportamientos modulares.
<b>Integración de ROS</b>	CARLA cuenta con integración con <i>ROS</i> a través de nuestro puente <i>ROS</i>
<b>Líneas base de conducción autónoma</b>	Proporcionan líneas de base de conducción autónoma como agentes ejecutables en CARLA, incluido un agente de <i>AutoWare</i> y un agente de aprendizaje de imitación condicional.



**Figura 2.3.2.- Sitio web del Simulador de código abierto para la investigación de conducción autónoma: CARLA [40].**

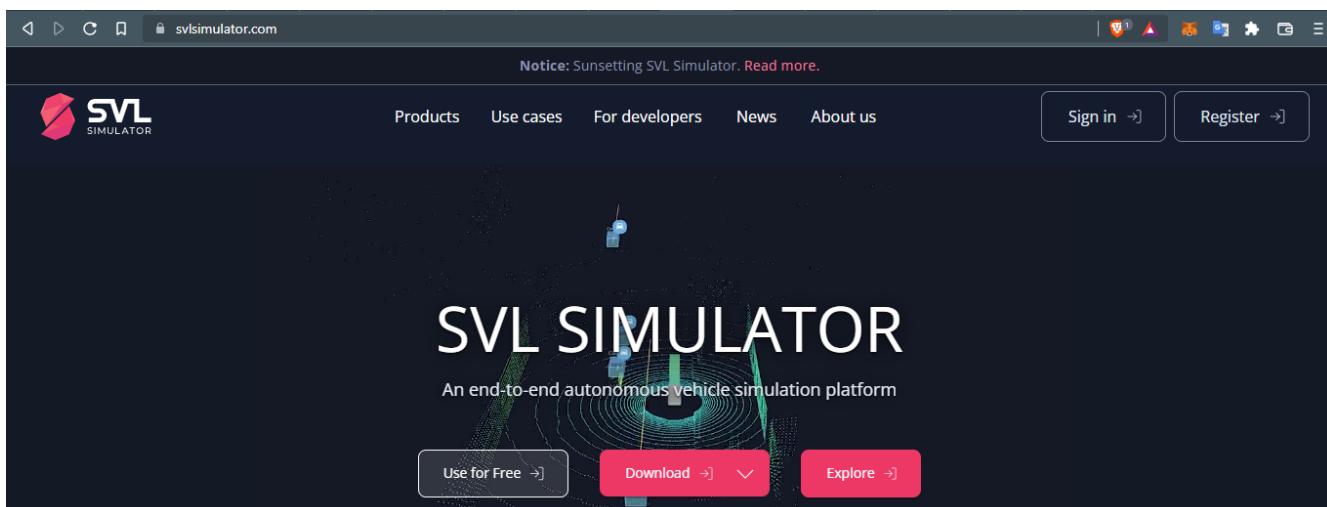
### 2.3.3 SVL Simulator

SVL lanzó su primera versión de simulador como un proyecto abierto para *Github* en noviembre de 2018 con el objetivo de permitir a los desarrolladores construir vehículos y robots autónomos a través de una simulación 3D de alto rendimiento de extremo a extremo. Desde entonces, se ha convertido *SVL Simulator* en un producto comercial que ayuda a los *OEM* automotrices, empresas de robótica y universidades de todo el mundo. El equipo está formado por ingenieros y especialistas altamente calificados, con años de experiencia en industrias que incluyen la automotriz, la robótica, el software, la electrónica de consumo y la tecnología móvil. “Creemos en la simulación como la clave para acelerar el desarrollo de máquinas autónomas. Nuestra visión es crear una plataforma de simulación estandarizada utilizada en el desarrollo de vehículos autónomos y robóticos, y creemos que el código abierto es clave para que esto suceda” (*LG Electronics*) [41].

En la Figura 2.3.2 se aprecia la página de inicio y los enlaces de descarga del simulador en su versión gratis y otra en versión de paga (Tabla 2.3). La versión de código abierto de SVL Simulator para uso interno de I+D, no está permitida para implementación comercial. La versión Premium del SVL Simulator de próxima generación incluye herramientas de simulación de código abierto, nube y contenido.

**TABLA 2.3.- Versiones de SVL Simulator.**

Simulación Básica	Free	Premium
<b>Simulación local</b>	Si	Si
<b>Simulación distribuida</b>	Si	Si
<b>Asistencia de Baidu Apolo</b>	Si	Si
<b>Soporte de Autoware.AI</b>	Si	Si
<b>Soporte de Autoware.Auto</b>	Si	Si
<b>API de Python</b>	Si	Si
<b>Uso comercial</b>	No	Si



**Figura 2.3.3.- Sitio web de SVL Simulator [41].**

## 2.3.4 QCar

QCar es el vehículo de Quanser Self-Driving Car Research Studio: una plataforma poderosa y altamente expandible diseñada específicamente para la investigación académica. Quanser se fundó en 1989 como respuesta a la necesidad de plataformas de hardware de escritorio optimizadas para la enseñanza y la investigación en la enseñanza de la ingeniería.

Dentro de los productos que ofrece la empresa, hay una categoría de robótica móvil donde se puede encontrar un modelo de vehículo a escala de arquitectura abierta diseñado para la investigación académica. Trabajando individualmente o en una flota, QCar es el vehículo ideal para validar la generación de conjuntos de datos, mapeo, navegación, aprendizaje automático, inteligencia artificial y otros conceptos avanzados de conducción autónoma.

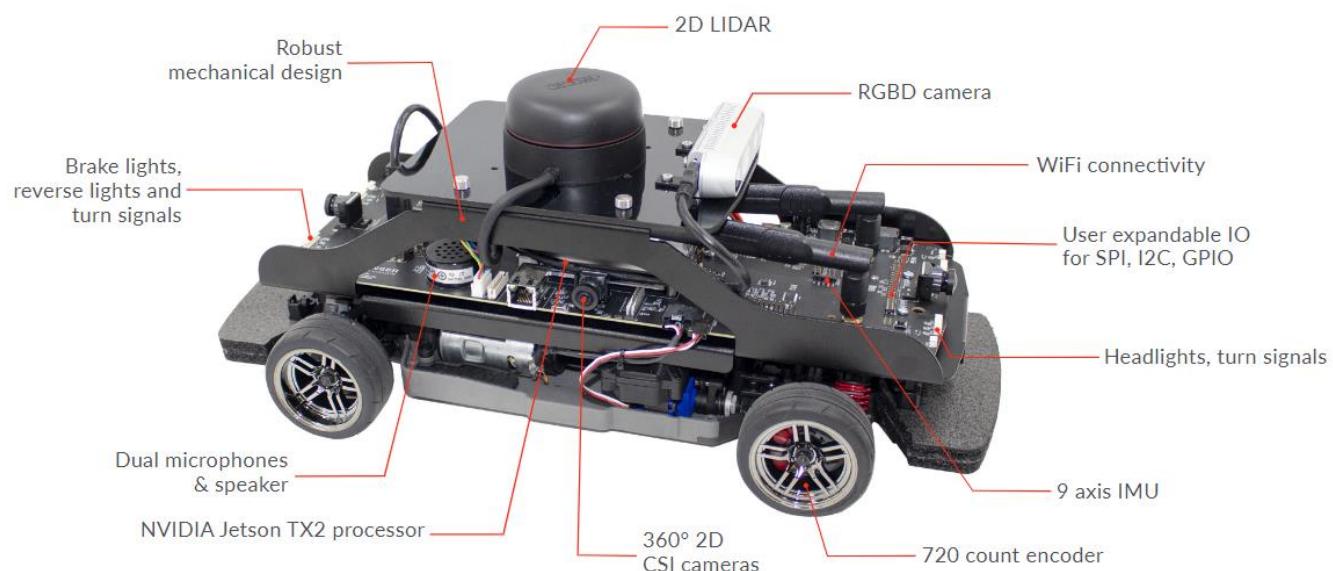


Figura 2.3.4.- Componentes principales de QCar [42].

La empresa trabaja en estrecha colaboración con un amplio espectro de organizaciones e individuos dentro de las comunidades educativa, de investigación, industrial y comercial. Entre estas organizaciones se encuentran instituciones académicas líderes como el Tecnológico de Monterrey, grupos de investigación industrial, empresas de hardware y software, organizaciones profesionales y grupos de divulgación. En la figura 2.3.5 se aprecia la página principal de Quanser Innovate Educate.

The screenshot shows a web browser displaying the Quanser website at [https://www.quanser.com/products\\_category/mobile-robotics/](https://www.quanser.com/products_category/mobile-robotics/). The page has a red header bar with the Quanser logo and navigation links for Request A Quote, REQUEST INFORMATION, PRODUCTS, COURSEWARE & RESOURCES, SUPPORT, COMMUNITY, BLOG, and ABOUT. Below the header, there's a search bar and a sidebar with the text "MOBILE ROB". The main content area features a heading "Mobile Robotics" and a "Browse By:" section with tabs for Lab Solution, Product Category (which is selected and highlighted in red), and Digital Experience. Under "Product Category", there are several sub-links: NI ELVIS III Top Boards, Rotary Motion Platform, Linear Motion Platform, Aerospace, Mobile Robotics (selected and highlighted in red), Manipulator Robotics, Telerobotics & Haptics, Structural Dynamics, Industrial Applications & Process Control, DAOs, Amplifiers, and Software. To the right of the menu, there's a large image of a black robotic arm or manipulator mounted on a base. At the bottom left, the URL [https://www.quanser.com/products\\_category/mobile-robotics/](https://www.quanser.com/products_category/mobile-robotics/) is visible.

**Figura 2.3.5.- Sitio web y productos de Quanser [43].**

### **3 ANTECEDENTES: VEHÍCULOS DE GUIADO AUTÓNOMO EN LA INDUSTRIA Y SUS APLICACIONES.**

#### **3.1 Introducción**

Debido al gran incremento que de la manufactura en servicios y productos la cual es beneficiada de acuerdo a la demanda existente en el mercado, es necesario incrementar el nivel de tecnología para una mejora en la automatización que permita hacer más eficiente el trabajo en los procesos industriales.

En este mismo contexto, es necesario implementar tecnologías específicas para la modernización de aspectos elementales dentro de las cadenas productivas. Generalmente existen diversas soluciones para dicha integración de forma automatizada como las bandas transportadoras que se hacen muy eficientes cuando se requiere un traslado de un punto "A" a un punto "B", sin embargo, cuando las distancias son muy prolongadas y además se requiere una flexibilidad o configuración en un punto intermedio en el transporte, una de las alternativas que tiene una buena relación en costo-beneficio son los vehículos de guiado autónomo que ayuda a aquellos procesos que requieran flexibilidad en el traslado de materiales y recursos, con el fin de realizar las tareas repetitivas y peligrosas, pero muy necesarias en la industria.

#### **3.2 Requerimientos de un sistema AGV**

El diseño e implementación de un sistema AGV es una tarea compleja, ya que se requieren tomar en cuenta muchas variables que impactan en el funcionamiento de dicho sistema y que son difíciles de predecir. Cada una de las decisiones que se tomen deben estar ligadas a otras variables y todas estas decisiones son muy importantes en el sistema. De acuerdo con (*Garber 2013*) y (*Beinschob 2017*), los principales aspectos que se deben de tener en cuenta en el diseño de un sistema AGV para la planificación de rutas, son las siguientes:

##### **3.2.1 Caracterización de instalaciones industriales y planificación de rutas.**

De acuerdo a unas características definidas del sitio donde se desea operar el vehículo se considera importantes realizar una ejecución de este tipo de tecnología.

En primer lugar la identificación del sitio, segundo, la aplicación del AGV en distintos entornos, en el que hay que tener en cuenta temas logísticos, constructivos e infraestructura, en relación a cableado eléctrico, movimientos y adecuación del lugar para disposición de maquinaria.

En tercer lugar, control del AGV, en donde es de suma importancia mencionar algunas técnicas de control empleadas en el avance de este tipo de tecnología y como último punto, el entorno de software para la programación de AGV (*Garber 2013*) [7].

Para cualquier tipo de proyecto que se quiera ejecutar en un entorno industrial, generalmente es muy importante hacer un reconocimiento de las instalaciones para así determinar si es viable o no la implementación de la solución planteada.

(Beinschob 2017) [8] presenta un enfoque para el despliegue de un AGV que tiene como objetivo reducir el tiempo, costo y personal involucrado. En primer lugar, propone una novedosa tecnología de escaneo láser SICK 3D competente para la industria con el fin de edificar una representación virtual precisa y relacionada de todo el del almacén. Mediante la utilización de una segmentación y un procesamiento adecuado, se crea un mapa semántico que contiene posiciones georreferenciadas en 3D, así como un mapa de localización 2D que elimina la necesidad de una solución dedicada al mapeo 2D. En la Figura 2.4.1 se muestra un mapa 3D con gráfico de características de un almacén industrial.

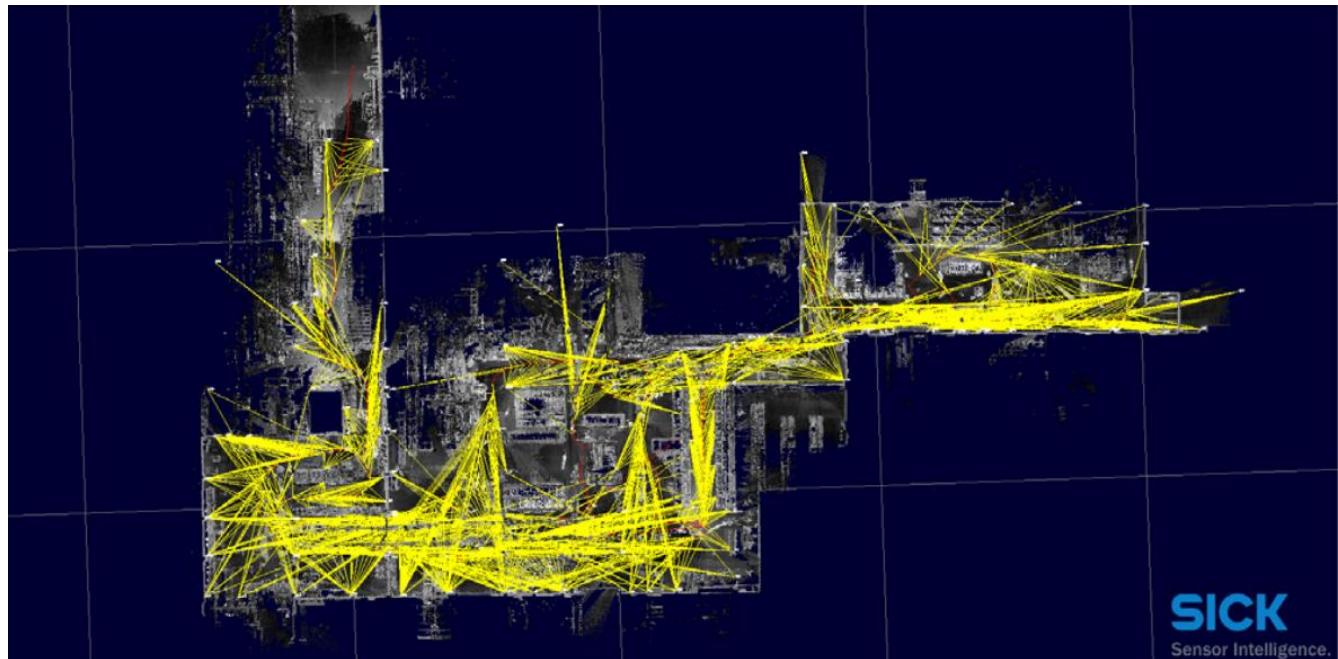


Figura 3.2.1.- Mapa 3D (Beinschob 2017). Extraído de [8].

### 3.2.2 Técnica de control del AGV en espacios cerrados.

Existen numerosas técnicas de control que auxilan a los AGV a cumplir con las tareas proyectadas dentro de una planta de producción o cualquier otro entorno en donde se requieran. Muchas de estas técnicas son aplicadas de acuerdo a la especificidad del proyecto o la necesidad de resolver una situación con una técnica específica. Este es el caso de (Silviranti 2017) [10] el cual busca la estabilidad del movimiento de un AGV debido a que este afecta directamente la seguridad de los productos entregados. (Silviranti 2017) argumenta que la variación de la carga durante el proceso de entrega es una de las causas del movimiento de inestabilidad de los vehículos. Por tal motivo, implementa un control de velocidad sin importar la variación de la carga utilizando un sistema de inferencia difusa integrado con el método P/D. La entrada del sistema proviene de los sensores de velocidad e inclinación cerca de las ruedas del AGV. Esta información es luego procesada por el sistema de inferencia difusa, cuyo

resultado son datos sobre la velocidad del AGV y los valores de los parámetros para el controlador PID. Adicionalmente utiliza sensores de ultrasonido para evasión de los obstáculos, *encoders* en las ruedas que retroalimentan hacia el sistema de control y *RFID* para la navegación del vehículo en sitio. En definitiva, (*Silviranti 2017*) [10] logra una velocidad constante de 10 m/min independiente de la variación de la carga en el AGV.

### **3.2.3 Software para la conducción automatizada del AGV.**

Existen diferentes lenguajes de programación y dependiendo el tipo de proyecto que se quiera realizar se debe de elegir una metodología para la programación de robots. En proyectos tan complejos como lo es un AGV, se requiere un nivel de programación para cada campo que lo va a integrar, tanto electrónico, mecánico, eléctrico, etc. Actualmente existen una variedad de sistemas que satisfacen todos estos campos de la ingeniería que puede ir desde la programación de robots que no requieran un alto grado de complejidad hasta una programación compleja como la de los vehículos autónomos.

En [44] se hace una comparación de software modular para la conducción automatizada, buscando un marco que sea modular y extensible con una sobrecarga baja, para apoyar el desarrollo tolerante a fallas y para proporcionar un rico ecosistema de herramientas de apoyo.

#### **3.2.3.1 KogMo-RTDB**

Uno de los marcos de software existentes es la base de datos en tiempo real para automóviles cognitivos (*KogMo-RTDB*) que se utilizó como proveedor de intercambio de datos dentro de los vehículos de investigación, pero carece de una comunidad poco sólida.

#### **3.2.3.2 ADTF**

El segundo marco de software consta de datos automotrices y marco activado por tiempo, *Assist Automotive Data and Time Triggered (ADTF)* es el entorno de desarrollo y prueba más utilizado para los sistemas avanzados de asistencia al conductor (ADAS) en la actualidad, pero lo que lo hace poco común es que es un producto de licencia profesional utilizado por empresas industriales y esto hace que su comunidad sea limitada.

#### **3.2.3.3 ROS**

El sistema operativo de robot (*ROS*) es un marco de software de código abierto que admite el desarrollo de sistemas complejos pero modulares en un entorno informático distribuido. Si bien los componentes centrales de *ROS* son muy genéricos, el enfoque principal de *ROS* y su ecosistema se establece en el desarrollo y la investigación de robots. Las partes críticas de rendimiento del marco están escritas en *C++*, pero las aplicaciones que operan sobre el marco pueden actualmente estar escrito en *C++*, *Python* o *Lisp*.

### **3.2.4 Aplicaciones del AGV en otros entornos de trabajo.**

Los AGV no solo se pueden implementar en entornos industriales sino también en otras áreas específicas que son repetitivas y requieren una agilización en el proceso (*Acosta 2015*) [9],

menciona la importancia de mejorar el flujo de trabajo del personal de salud. Por lo tanto propone una plataforma móvil hospitalaria para tareas de logística la cual consta de la operación de un vehículo automatizado que ayude a la productividad de las tareas logísticas dentro de un hospital, como la recolección y entrega de productos, para que el personal puede enfocarse en las tareas necesarias para servir mejor a la comunidad. Para detectar y evitar obstáculos, se ha equipado la base móvil con un láser de escaneo colocado en la parte delantera de la base móvil, que cubre 180° (Figura 3.2.2). Los datos recopilados del láser combinados con los datos de edometría de la base móvil se utilizan para generar un mapa de los establecimientos de salud donde opera la base móvil.

Además, los datos del buscador de rango láser ayudan al sistema a comprender el entorno y tratar de hacer coincidir eso con una ubicación en el mapa para mantener un seguimiento de la base móvil mientras navega.



Figura 3.2.1.- Prototipo de plataforma móvil hospitalaria para tareas de logística [9].

### 3.2.5 Beneficios de un sistema AGV.

De acuerdo con la empresa SS SCHAEFER, la perfecta integración de las soluciones AGV en el sistema de control de gestión de la empresa y en el área de producción, permite, que muchos procesos de producción y almacenamiento se sincronicen de manera más estrecha y eficiente [45]. Se puede combinar una gran cantidad de fuentes y orígenes.

El uso de AGV en almacenes no ha dejado de aumentar en los últimos años. Esto se debe al hecho de que los AGV mejoran la eficiencia de la manipulación las mercancías y, por consiguiente, la productividad [46]. Entre las ventajas de un sistema AGV se encuentran:

- Automatización flexible y escalable de los flujos internos de material
- Mayor eficiencia y más flexibilidad
- Reducción sostenible de los tiempos de ciclo
- Bajas inversiones y costes operativos
- Tiempos de implementación cortos
- Reducción de los daños en el transporte y entregas erróneas
- Fácil adaptación a los nuevos requerimientos
- Composición individual de la flota interna de AGV
- Máxima flexibilidad para la selección interna del almacén debido a la entrega secuenciada
- Transporte cuidadoso y seguro de los artículos y transportadores de carga
- Fácil integración y bajos requisitos de infraestructura (*IT*)
- Integración eficiente en el hardware y software existentes gracias a la interfaz de estándar del sistema
- Mejora en la ergonomía y flujos de materiales óptimos
- Mayor seguridad en las operaciones y mejores condiciones de trabajo
- Se puede usar incluso en entornos hostiles (almacenamiento a temperatura controlada o áreas peligrosas)
- Concepto innovador de energía (operaciones de carga automáticas y rápidas)
- Máxima flexibilidad en cuanto a modificaciones y ampliaciones del sistema
- Servicio personalizado y opciones de mantenimiento
- Conexión en red de diferentes áreas del almacén y trabajo a largas distancias
- Gestión compartida del AGV y personas en las áreas de producción y logística
- Posibilidad de evitar obstáculos o rutas alternativas
- Optimización y control de rutas de transporte mediante controlador de flota integrado
- Uso flexible de la sala y del espacio

## **4 SISTEMA DE MANUFACTURA FLEXIBLE.**

### **4.1 Introducción**

En la industria existen operaciones de manufactura que requieren un proceso de almacenamiento, transporte y mecanizado de diferentes materiales. Esto trae como consecuencia la preparación de los dispositivos de sujeción, la programación de los diferentes elementos automáticos (robots, actuadores, centros de maquinado, sistemas de inspección, etc.) y al registro de información. Para ello se cuenta con una estación para cada proceso y aunado a eso un sistema de transporte con vehículos transportadores de piezas y una estación central para programación, dirección y control de las demás estaciones y del flujo de materiales. Un Sistema de Manufactura Flexible (SMF) está dinámicamente interactuando con su medio ambiente, lo cual afecta en la operación general [47], [48], [49].

En general, se da el nombre de fabricación o manufactura flexible a un sistema de fabricación conformado por máquinas y subsistemas enlazados por un sistema de transporte y control común, con la posibilidad de realizar diversas tareas, dentro de un margen razonable. En este apartado del trabajo se comprenderá el esquema general del Sistema de Manufactura Flexible su diseño, niveles, un ejemplo de aplicación y la coordinación de robots multiagentes AGV's.

### **4.2 ¿Qué es un Sistema de Manufactura Flexible?**

De acuerdo con Un Sistema de Manufactura Flexible que en inglés se le conoce como *Flexible Manufacturing System (FMS)*, es una celda de maquinado con tecnología de grupos con un nivel de automatización alto que se conforma de un grupo de estaciones de procesamiento que generalmente son máquinas de Control Numérico por Computadora (CNC) interconectadas entre sí por medio de un sistema.

La razón por la que el *FMS* se llama flexible es que es capaz de procesar una variedad de diferentes estilos de piezas simultáneamente en varias estaciones de trabajo, y la combinación de estilos de piezas y cantidades de producción se pueden ajustar en respuesta a patrones de demanda cambiantes. Un *FMS* se basa en los principios de la tecnología de grupo. Ningún sistema de fabricación puede ser completamente flexible. Hay límites en la gama de piezas o productos que se pueden fabricar en un *FMS*. En consecuencia, un sistema de fabricación flexible está diseñado para producir piezas (o productos) dentro de una gama definida de estilos, tamaños y procesos. En otras palabras, un *FMS* es capaz de producir una sola familia de piezas o una gama limitada de familias de piezas.

Un término más apropiado para *FMS* sería sistema de fabricación automatizado flexible. El uso de la palabra "automatizado" distinguiría este tipo de tecnología de producción de otros sistemas de fabricación que son flexibles, pero no automatizado, como una celda de máquina *GT* (Tecnología de Grupo) tripulada. La palabra "*flexible*" lo distinguiría de otros sistemas de fabricación altamente automatizados pero no flexibles, como una línea de transferencia convencional [50].

### 4.3 Flexibilidad del sistema.

Para que un sistema pueda ser flexible, un sistema de fabricación debe ser capaz de realizar las siguientes funciones en cada ciclo de trabajo:

- Identificación de diferentes unidades de trabajo. Diferentes estilos de piezas o productos requieren diferentes operaciones. El sistema de fabricación debe identificar la unidad de trabajo para poder realizar las operaciones correctas.
- Cambio rápido de instrucciones de funcionamiento. Las instrucciones, o el programa de piezas en el caso de máquinas de producción controladas por computadora, deben corresponder a la operación correcta para la pieza dada.
- Cambio rápido de configuración física. La flexibilidad en la fabricación hace que las diferentes unidades de trabajo no se produzcan en lotes.

### 4.4 Clasificación de los Sistemas de Manufactura.

Tomando en cuenta los tres puntos anteriores, se pueden identificar tres tipos básicos de sistemas de fabricación: celdas de una sola estación, sistemas de múltiples estaciones con enrutamiento fijo y sistemas de múltiples estaciones con enrutamiento variable. Cada tipo se puede implementar como un sistema tripulado o un sistema automatizado, como se muestra en la Figura 4.3.0. En el caso de sistemas multiestación, también son posibles híbridos que consisten en estaciones tripuladas y automatizadas.

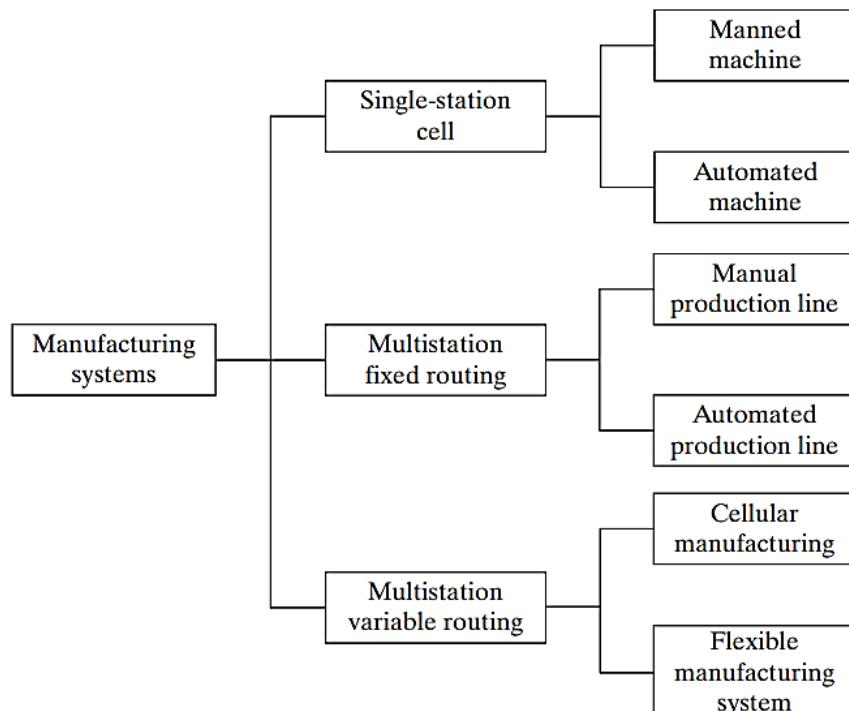


Figura 4.4.- Clasificación de los Sistemas de Manufactura [50].

## 4.5 Esquema general del FMS.

En el mismo contexto de 4.3 y 4.4, se puede decir que la flexibilidad es un atributo que se aplica tanto a los sistemas manuales como a los automatizados. En los sistemas manuales, los humanos suelen ser los habilitadores para darle la flexibilidad al sistema.

Tomando en cuenta la Figura 4.5, que consta de dos máquinas CNC y que a su vez un robot industrial carga y descarga desde un sistema de almacenamiento. La celda funciona sin supervisión durante largos períodos de tiempo. Periódicamente, un trabajador debe descargar piezas terminadas del sistema de almacenamiento y reemplazarlas con piezas de trabajo nuevas. Por definición, esta es una celda de fabricación automatizada, y se podría argumentar que sí, es flexible porque la celda consta de máquinas herramienta CNC, y dado que las máquinas CNC son flexibles porque pueden programarse para mecanizar diferentes configuraciones de piezas.

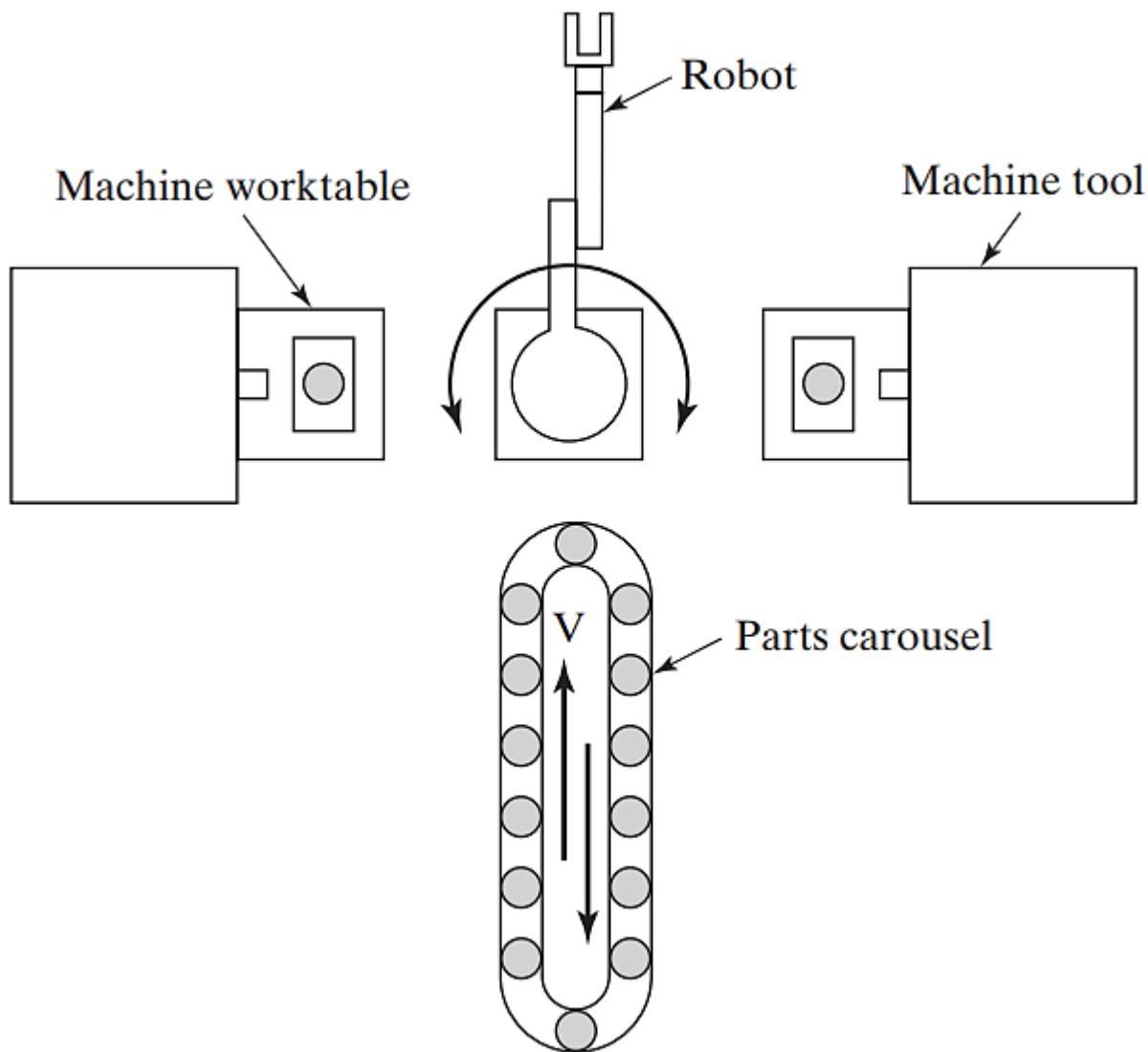


Figura 4.5.- Celda de fabricación automatizada con dos máquinas CNC [50].

Sin embargo, como se mencionó en 4.3, si la celda solo opera en un modo por lotes, en el que ambas máquinas producen el mismo estilo de pieza en lotes de varios cientos de unidades, entonces esto no califica como fabricación flexible. Para calificar como flexible, un sistema de fabricación automatizado debe satisfacer las siguientes cuatro pruebas de flexibilidad:

1. Prueba de variedad parcial. ¿Puede el sistema procesar diferentes estilos de piezas o productos en un modo de modelo mixto (no por lotes)?
2. Prueba de cambio de horario. ¿Puede el sistema aceptar fácilmente cambios en el programa de producción, es decir, cambios en la mezcla de partes y/o cantidades de producción?
3. Prueba de recuperación de errores. ¿Puede el sistema recuperarse correctamente de fallas y averías del equipo, de modo que la producción no se interrumpa por completo?

Sí la respuesta a estas preguntas es “SI”, entonces el sistema es flexible.

## 4.6 Diseño del FMS.

De acuerdo con el autor de [47], el diseño de un sistema productivo a ejecutarse en el *FMS* debe ser planeado con anterioridad para definir la secuencia de cada una de las actividades productivas. De acuerdo con lo anterior y lo definido en el diseño de producto que previamente debe estar diseñado, se puede representar de la siguiente manera:

- Selección de materia prima.
- Elección del producto a fabricar.
- Transporte al centro de mecanizado.
- Fabricación del producto.
- Almacenamiento.
- Monitoreo y control de datos.
- Visualización de procesos de producción.
- Visualización y control de las estaciones.
- Sincronización de las máquinas.
- Intercambio de información de las estaciones.
- Regulación de sistemas de transporte y de los robots.
- Creación de alarmas de emergencia.
- Planes de procesos.
- Dispositivos: corresponden a los equipos controlados en el *FMS*, entre los que se encuentran para este caso el torno, la banda transportadora, la fresadora y el robot. Estos dispositivos son configurados por el usuario, es decir, si al momento de utilizar el *FMS* no se tiene una máquina adicional, por ejemplo, puede agregarse para que pase a formar parte del sistema.

- Controladores: para que los equipos del *FMS* puedan funcionar correctamente es necesario agregarle a cada uno un controlador especificado que permita la comunicación y transferencia de tareas desde un software al dispositivo y viceversa.
- Tareas: función definida previamente en un software o que es desarrollada por el programador para llevar a cabo una actividad.

El autor (Sergio Foyo 2015) presenta en [51] un esquema general de un *FMS* en el cual se diseña una arquitectura de control que sea capaz de coordinar AGV's y las tareas de los procesos obteniendo su mayor concurrencia y obedeciendo a las restricciones de un proceso en un *FMS* general.

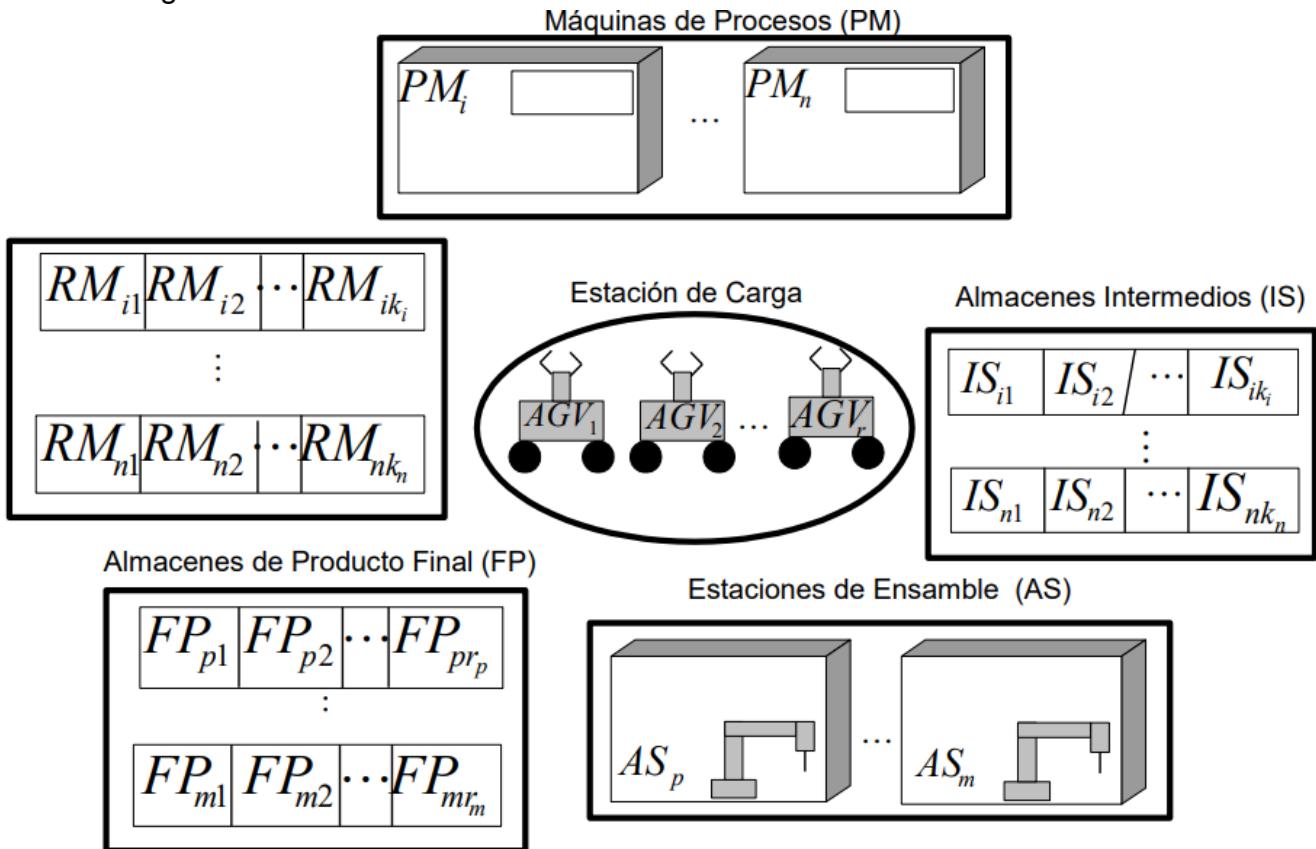


Figura 4.6.- Esquema general de un FMS [51].

## 4.7 Niveles de Flexibilidad.

Cada *FMS* está diseñado para una aplicación específica, es decir, una familia específica de piezas y procesos. Por lo tanto, cada *FMS* está diseñado a la medida y es único. Dadas estas circunstancias, uno esperaría encontrar una gran variedad de diseños de sistemas para satisfacer una amplia variedad de requisitos de aplicación.

Los sistemas de fabricación flexible se pueden distinguir según los tipos de operaciones que realizan:

- Operaciones de procesamiento
- Operaciones de ensamblaje.

Un *FMS* generalmente está diseñado para realizar uno u otro, pero rara vez ambos.

Los sistemas de fabricación flexibles tienen un cierto número de máquinas de procesamiento. Las siguientes son categorías típicas:

- **Celda de una sola máquina:** consta de un centro de mecanizado CNC combinado con un sistema de almacenamiento de piezas para operación desatendida (Figura 4.7).
- **Celda de Manufactura Flexible (Por sus siglas en inglés FMC):** consta de dos o tres estaciones de trabajo de procesamiento (normalmente centros de mecanizado CNC o centros de torneado) más un sistema de manipulación de piezas. El sistema de manipulación de piezas está conectado a una estación de carga/descarga. El sistema de manipulación suele incluir una capacidad limitada de almacenamiento de piezas (Figura 4.7.1).
- **Sistema de Manufactura Flexible (Por sus siglas en inglés FMS)** tiene cuatro o más estaciones de procesamiento conectadas mecánicamente por un sistema común de manejo de piezas y electrónicamente por un sistema informático distribuido.

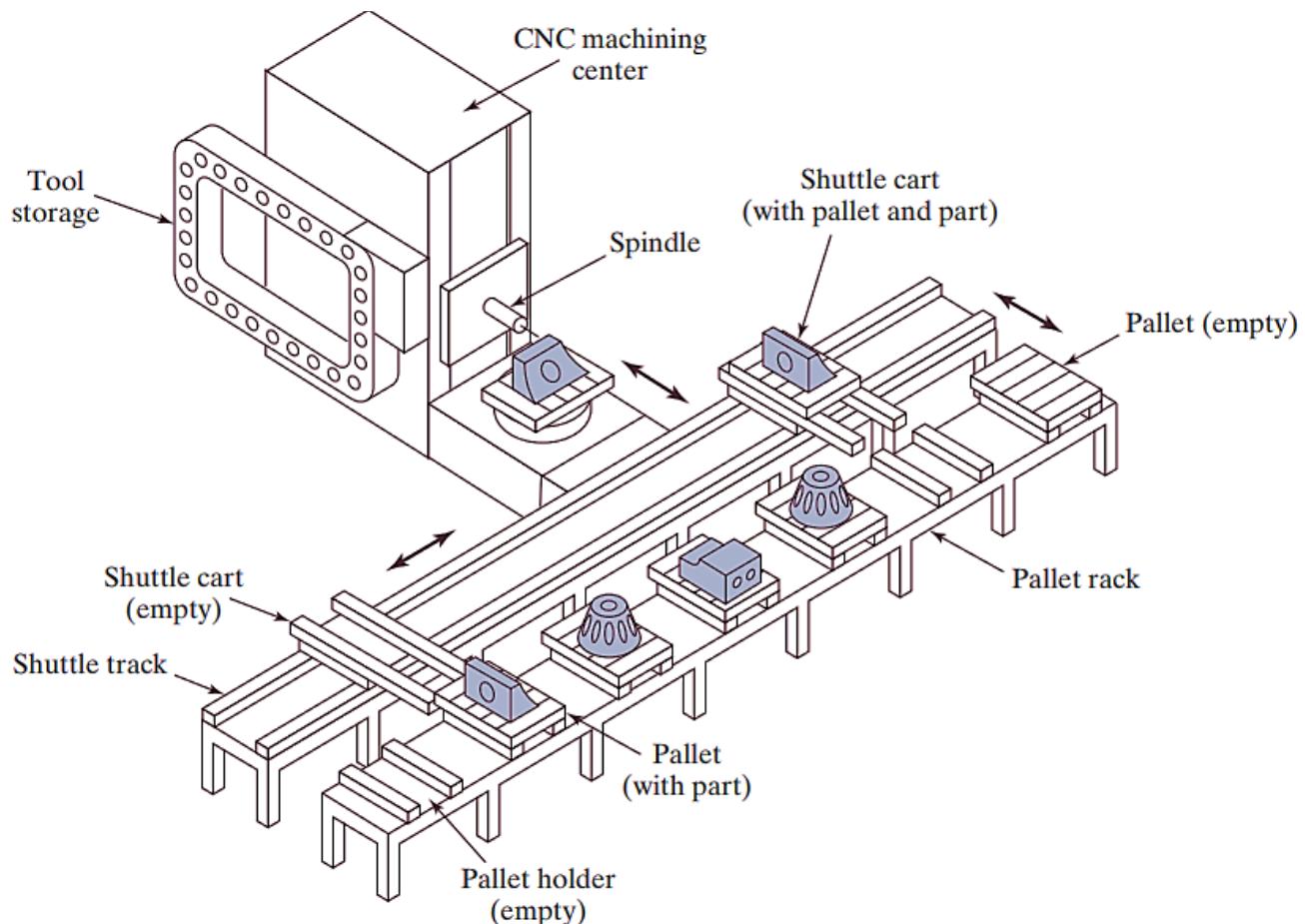
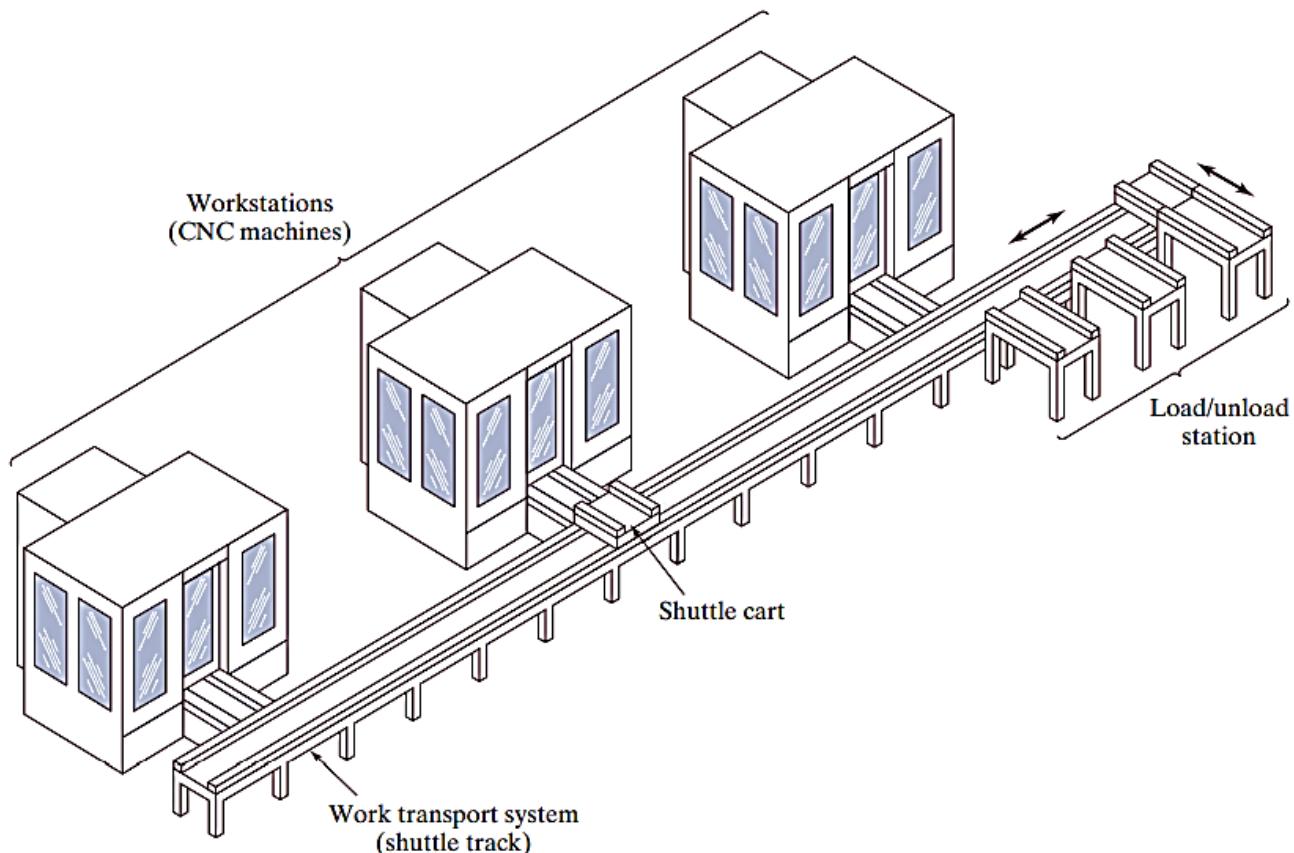


Figura 4.7.- Celda de una sola máquina [50].



**Figura 4.7.1.- Celda de Manufactura Flexible (FMC) [50].**

Siguiendo con la metodología de (Groover 2008), presenta un testeo de cuatro pruebas de flexibilidad en [50]:

1. Variedad de partes.
2. Cambio de horario.
3. Error de recuperación.
4. Parte nueva.

**TABLA 4.7.- Pruebas de Flexibilidad aplicadas a: Celda de una sola máquina, FMC y FMS [50].**

PRUEBAS DE FLEXIBILIDAD				
Tipo de Sistema	Variedad de Partes	Cambio de horario	Error de recuperación	Parte nueva
Celda de una sola maquina	Sí, pero el procesamiento es secuencial, no simultáneo.	✓	Recuperación limitada debido a una sola máquina.	✓
Celda de Manufactura flexible (FMC)	Sí, producción simultánea de diferentes piezas.	✓	Recuperación de errores limitada por menos máquinas que FMS.	✓

Sistema de Manufactura Flexible ( <i>FMS</i> )	Sí, producción simultánea de diferentes piezas.	✓	La redundancia de la máquina minimiza el efecto de las averías de la máquina.	✓
--	---	---	---	---

#### 4.8 Consideraciones para el manejo de materiales.

El manejo y transporte de materiales es una de las actividades en el sistema de distribución más grande mediante el cual los materiales, las piezas y los productos se mueven, almacenan y rastrean en la infraestructura. El término comúnmente utilizado para el sistema más grande es logística, que se ocupa de la adquisición, movimiento, almacenamiento y distribución de materiales y productos, así como la planificación y control de estas operaciones para satisfacer la demanda de los clientes.

Las operaciones logísticas se pueden dividir en dos categorías básicas: logística externa y logística interna, siendo esta última la utilizada en un *FMS* ya que permite el manejo de materiales, implica el movimiento y almacenamiento de materiales dentro de una instalación determinada.

El equipo de manejo de materiales generalmente se ensambla en un sistema, el cual debe especificarse y configurarse para satisfacer los requisitos de una aplicación en particular como se mencionó en (4.6). En la Tabla 4.8 se consideran los factores que deben tomarse en cuenta según (Muther y Haganas) [52], a la hora del diseño del sistema de manejo de materiales clasificándolos por las características físicas presentadas.

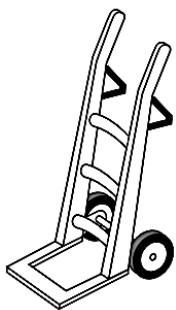
**TABLA 4.8.- Características de los materiales para su manejo [52].**

Categoría	Descripción o Medida
<b>Estado físico</b>	Sólido, líquido o gas
<b>Tamaño</b>	Volumen, largo, ancho, alto
<b>Peso</b>	Peso por pieza, peso por unidad de volumen
<b>Forma</b>	Largo y plano, redondo, cuadrado, etc.
<b>Condición</b>	Caliente, frío, húmedo, sucio, pegajoso
<b>Riesgo de daño</b>	Frágil, quebradizo, resistente
<b>Riesgo para la seguridad</b>	Explosivos, inflamables, tóxicos, corrosivos, etc.

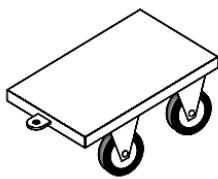
#### **4.8.1 Vehículos y transportes industriales para el manejo de materiales.**

Los vehículos y transportes se dividen en dos categorías: sin motor y con motor. Los tipos sin motor a menudo se denominan carretillas de mano porque son empujadas o jaladas por trabajadores humanos que son capaces de mover estas cargas, ya que las cantidades de material movido y las distancias recorridas son relativamente bajas. En la Figura 8.1.1 se muestran las herramientas sin motor que son operadas por el ser humano.

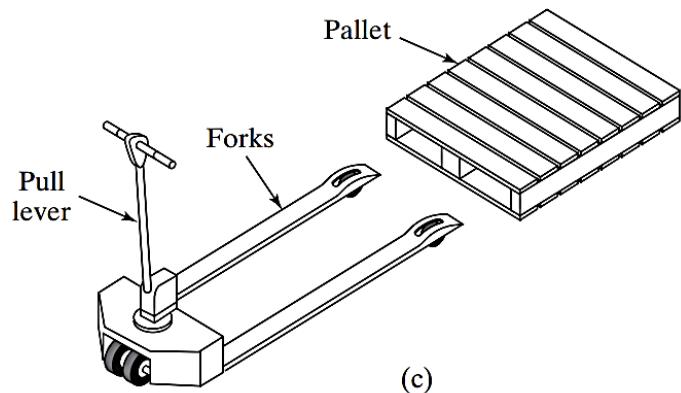
- (a) Son generalmente más fáciles de manipular por el trabajador pero están limitados a cargas más livianas.
- (b) Son posibles varias configuraciones de ruedas, incluidas ruedas fijas y ruedas giratorias.
- (c) Son carretillas para tarimas, tienen dos horquillas que se pueden insertar a través de las aberturas de una tarima.



(a)



(b)

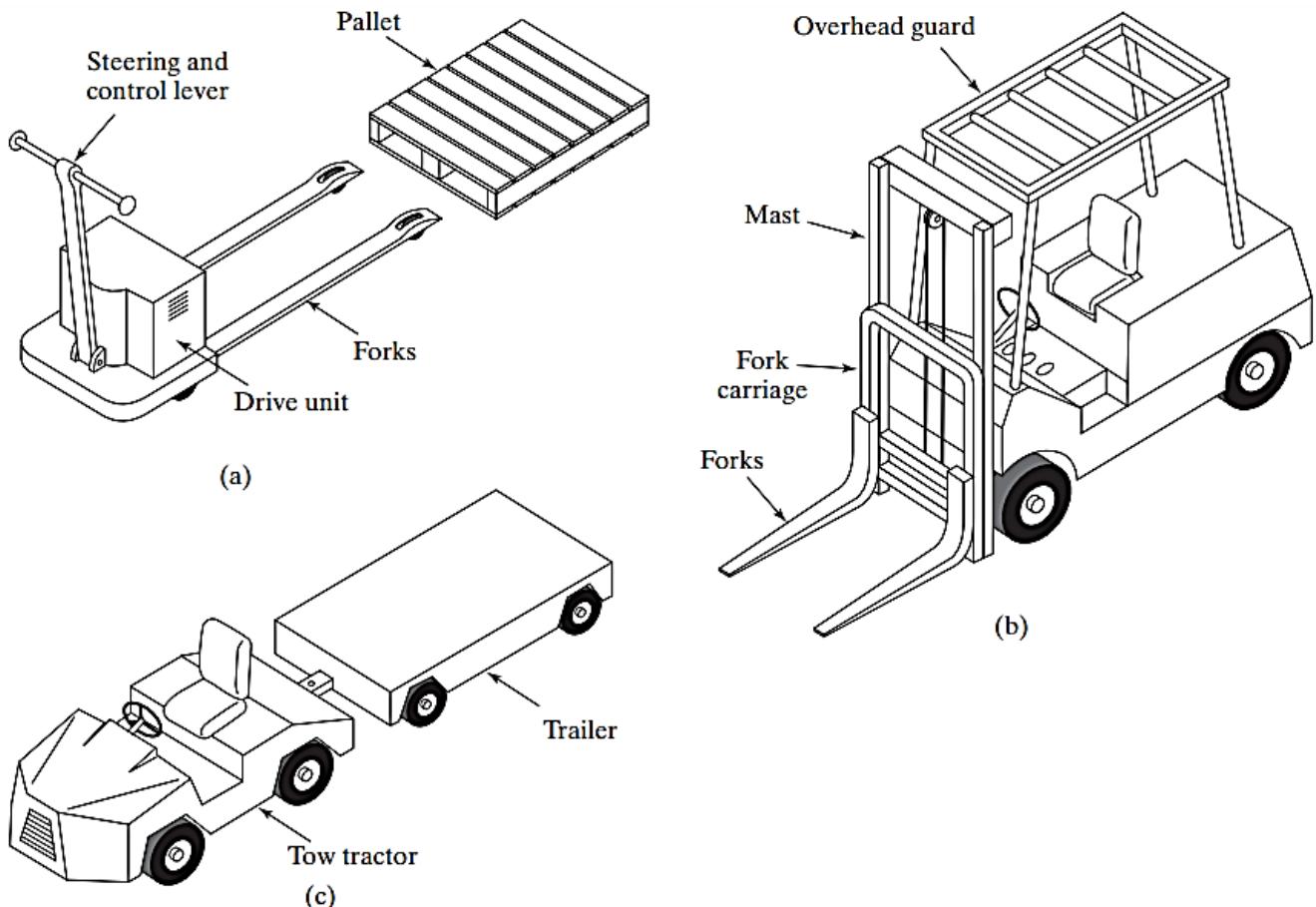


(c)

**Figura 4.8.1.- Transporte NO motorizado operado por el humano [50].**

En la Figura 8.1.2 se aprecian tres tipos principales de camiones motorizados:

- (a) camión con operador a pie.
- (b) montacargas.
- (c) tractor de remolque.



**Figura 4.8.2.- Transporte motorizado operado por el humano [50].**

#### 4.9 Transporte de materiales con AGV's.

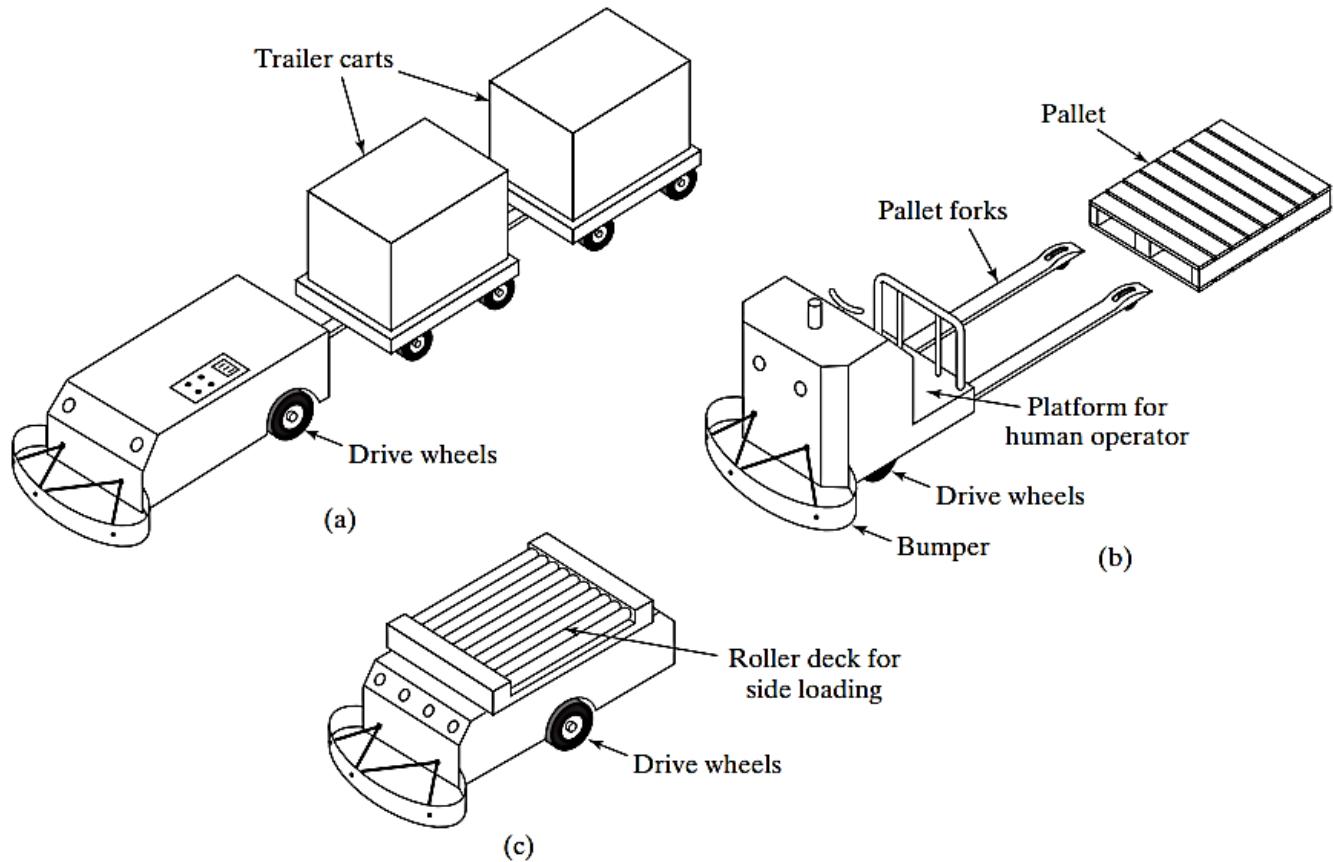
El sistema de AGV's se abordó en el capítulo 3 de este trabajo y ahora en este apartado se mencionan tres categorías distintivas en un AGV en un sistema más específico.

Los vehículos guiados automatizados se pueden dividir en las siguientes categorías:

- Vehículos de remolque para trenes sin conductor.
- Transporte de Pallets.
- Transportadores de carga unitaria.

Estos tres tipos de sistemas de AGV se ilustran en la Figura 4.9 donde:

- (a) Es Un tren sin conductor consta de un vehículo un AGV que tira de uno o más remolques para formar un tren.
- (b) se utilizan para mover cargas paletizadas a lo largo de rutas predeterminadas.
- (c) Los transportadores de carga unitaria AGV se utilizan para mover cargas unitarias de una estación a otra. A menudo están equipados para la carga y descarga automática de tarimas o bandejas mediante rodillos motorizados, cintas móviles, plataformas elevadoras mecanizadas u otros dispositivos construidos en la cubierta del vehículo.



**Figura 4.9.- Tres tipos de sistemas de AGV [50].**

## **5 INDUSTRIA 4.0.**

Las empresas deberán establecer una red de colaboración para poder llevar a cabo la producción global. La red debe estar compuesta por proveedores de tecnologías, empresas de logística y fabricantes. La cooperación con empresas de servicios altamente especializados y nuevas empresas creativas es especialmente importante para la innovación en la Industria.

En la Industria 4.0, la producción se basa en el uso de tecnologías digitales. Esto permite la producción flexible de una variedad de productos. Además, el uso de tecnologías digitales permite una mayor colaboración entre las empresas para mayor producción en masa e individualizada.

En este capítulo se aborda desde el concepto de la industria 4.0, sus inicios, las tecnologías que están relacionadas y los países que están involucrados en este hito diferenciándolos entre potencias a nivel mundial y a nivel Latinoamérica. Finalmente se mencionan algunas aplicaciones de la industria 4.0 en diferentes campos.

### **5.1 Definición**

Cuando se habla de la industria 4.0, resulta complejo dar con certeza una definición debido a que el tema es muy complejo, sin embargo en este trabajo se define como:

Industria 4.0 (*i4.0*): Es Impacto crucial en la información y la comunicación tecnológica, teniendo impacto en el sector industrial con mayor flexibilidad en las necesidades de producción, eficiencia de recursos e integración de procesos; que va desde el monitoreo del equipo hasta la entrega final del producto, permitiendo que los sistemas ciber físicos tengan una interacción en tiempo real entre maquinaria, software y humanos haciendo uso de la tecnología IoT (internet of things). Esta es la cuarta etapa de la revolución industrial siendo la primera el uso de la energía a vapor y mecanizado de los procesos.

*A lo largo de este capítulo se hará referencia a la industria 4.0 como: i4.0.*

### **5.2 INICIOS**

Para tener en claro donde surgió la i4.0 se tiene que mencionar a los países tradicionalmente industrializados como Francia, Italia, Gran Bretaña y Estados Unidos, etc. Sin embargo, el país en dar a conocer este concepto es Alemania que en tan pocos años ya se ha extendido cada vez más por todo el mundo, especialmente en Europa.

Este concepto nació con la finalidad de proporcionar a la industria alemana el necesario apoyo político para que el país fuera el líder mundial en el ámbito de la oferta y la demanda de tecnologías de producción digitalizadas [53].

## 5.3 TECNOLOGÍAS

Estas son algunas de las tecnologías presentes en la i4.0 y están definidas de acuerdo a (*Cambridge Dictionary*). En la Figura 5.1 se pueden apreciar otras tecnologías.

**5.3.1 Internet de las Cosas:** Son todos aquellos dispositivos, incluidas computadoras, teléfonos, tecnología portátil y sistemas inteligentes, que pueden conectarse entre sí a través de Internet.

**5.3.2 Big Data:** Conjuntos muy grandes de datos producidos por personas que utilizan Internet y que solo pueden almacenarse, comprenderse y utilizarse con la ayuda de herramientas y métodos especiales

**5.3.3 Inteligencia Artificial:** El estudio de cómo producir máquinas que tengan algunas de las cualidades que tiene la mente humana, como la capacidad de comprender el lenguaje, reconocer imágenes, resolver problemas y aprender.

**5.3.4 Cloud Computing:** el uso de servicios, programas informáticos, etc. que están en Internet en lugar de los productos físicos que están en una computadora personal.

**5.3.5 Realidad Aumentada:** imágenes producidas por una computadora y utilizadas junto con una vista del mundo real.

**5.3.6 Ciberseguridad:** Cosas que se hacen para proteger a una persona, organización o país y su información informática contra delitos o ataques realizados a través de Internet

**5.3.7 Simulación:** Sn modelo de un conjunto de problemas o eventos que se pueden usar para enseñar a alguien cómo hacer algo, o el proceso de hacer tal modelo.



**Figura 5.1. Tecnologías presentes en la i4.0 [54].**

## **5.4 Industria 4.0 a nivel Mundial.**

Existen ya varios países que están desarrollando, o cuentan ya con un programa para esta revolución industrial, ya que hay muchas posibilidades de innovación y diferenciación (como se vio en 5.3) por empresas nacionales, así como la mejora en la competitividad que pueda producirse, contribuyendo así a un crecimiento del país.

(Cardenas 2020) [55], Menciona que "*A lo largo del tiempo diferentes países han profundizado en la implementación de las diferentes tecnologías que acoge la industria 4.0 al sector manufacturero.*" Siguiendo al mismo autor, los países que más sobresalen en la i4.0 son los siguientes:

**5.4.1 Japón:** Desde la segunda revolución industrial no ha tenido ningún problema en avanzar y experimentar nuevas cosas. Por mencionar una de ellas son sus fascinantes inspiraciones que causan sus robots. No en vano, lo bautizan como "el paraíso de los robots".

**5.4.2 Serbia:** Analiza el progreso en la implementación de Industria 4.0 en empresas manufactureras puesto que mediante encuestas se evidencia que las organizaciones cuentan con una madurez digital. Basados en aspectos como el apoyo del gobierno, universidades e instituciones de investigación, soporte para usuarios de industria 4.0 y proveedores de soporte para la misma.

**5.4.3 Alemania:** Ha sido uno de los países con mayor progreso y alcance en la industria prueba de esto es que ha sido merecedor de ser un pilar en la implementación de estas tecnologías suscita que crea nuevos valores que no se han visto antes, construye nuevos modelos de negocio y resuelve diversos problemas al vincular las cosas dentro y fuera de una fábrica y los servicios a través de las redes de comunicación basadas en CPS, IoT e IoT (Internet de los Servicios).

**5.4.4 Estados Unidos:** La investigación y los programas de desarrollo relacionados con la fabricación en EE. UU, se centran en las asignaciones de tecnología clave, que incluyen IoT, big data, análisis de datos, CPS, integración de sistemas, fabricación sostenible y aditivos fabricación para responder agresivamente a la fabricación innovadora cambio ambiental llamado la cuarta revolución industrial.

**5.4.5 Corea:** Este país se encuentra impulsando el desarrollo de la fabricación inteligente relacionada con tecnologías similares a la Industria 4.0 de Alemania y a las estrategias nacionales relacionadas con la Fabricación Inteligente en los EE.UU. Los departamentos que apoyan los principales proyectos de desarrollo tecnológico están operando hojas de ruta con características ligeramente diferentes, pero se están moviendo gradualmente hacia el establecimiento e impulso de estrategias integradas que van más allá de las barreras departamentales.

## **5.5 Industria 4.0 en Latinoamérica.**

En [53] (Martinez 2020), que a su vez cita a (Schmitz , 1993), menciona a América Latina en el ambiente en que comenzaron a desarrollarse las Micro, Pequeñas y Medianas Empresas (MIPYMES) fue muy particular ya que por un lado en esos momentos “*las economías latinoamericanas estaban cerradas al mundo, con una baja competencia y una alta incertidumbre económica*” y por otro se mantenía un esquema de “Industrialización por Sustitución de Importaciones (modelo ISI). Muchas se originaron como empresas propiamente dichas, donde existe una gestión empresarial y el trabajo remunerado y en muchos de los casos fueron creadas por los gobiernos municipales y nacionales con el fin de producir pequeñas producciones que no podían ser elaboradas por las grandes empresas. Este tipo de empresas es considerado como la estructura y grupo empresarial más complejo de Latinoamérica y del Caribe, y algunos se arriesgarían a asegurar que del mundo. Esto es debido a su amplia diversidad en cuanto a sus dimensiones económicas, número de trabajadores, capacidad empresarial, sectores de la economía en que intervienen etc; lo que hace por momentos muy difícil comprender su real magnitud y alcance tanto nacional como internacional.

Por otro lado en [55] menciona que en Latinoamérica la implementación de la industria 4.0 se reduce considerablemente en comparación con países más desarrollados, debido a diferentes factores como la cultura tradicional de las organizaciones, desconocimiento de las tecnologías, dificultad de inversión, apoyo del gobierno en el sector de investigación e innovación esto impide la implementación de las diferentes tecnologías emergentes de esta revolución. Las políticas creadas por países desarrollados y en desarrollo para permitir el avance de la revolución digital de la Industria 4.0 son diferentes.

## **6 Industria 4.0 en México.**

### **6.1 Características del sector Manufacturero en México.**

En [56] hace una comparación entre México y Canadá. México es manufacturero de clase mundial, exportando más de mil millones de dólares por día. El cincuenta por ciento de estas exportaciones son productos manufacturados, de los cuales una gran parte son tecnologías altamente sofisticadas. De hecho, más del 80% de las exportaciones de alta tecnología en América Latina se producen en México, y el país exporta productos aún más sofisticados que Canadá.

También se menciona que los acuerdos comerciales internacionales, junto con la correlación entre el peso mexicano y el dólar estadounidense, más la experiencia acumulada de las industrias electrónica y automotriz, han creado una zona altamente competitiva para el desarrollo de una industria manufacturera de exportación. Esta situación, ha atraído a un gran número de empresas interesadas en México como plataforma exportadora. Las empresas europeas y asiáticas se encuentran entre las más atraídas por las ventajas competitivas de México, aun siendo Estados Unidos el mercado final.

## 6.2 México hacia la Industria 4.0.

Una publicación en el portal de la (SEECO 2018) dice que “Nuestro país sabe que para poder avanzar hacia la llamada Cuarta Revolución Industrial y la economía del conocimiento y del valor agregado, es necesario trabajar en cuatro pilares fundamentales” [57].

- **Desarrollo de capital humano:** México debe desarrollar capital humano de primera generación, con las habilidades más recientes que esta cuarta revolución exige. Estas incluyen, habilidades en información de datos, minería de datos, y ciberseguridad, entre otras.
- **Innovación:** hacer que las empresas innoven a través del uso de herramientas específicas.
- **Clústers:** generar sinergias positivas entre los clústers para detonar acciones coordinadas.
- **Adopción de tecnología:** para que las pequeñas y medianas empresas la adopten, la obtengan de inmediato y generen procesos productivos más competitivos.

La misma publicación menciona que mediante el desarrollo de esta industria, México genera una mayor productividad y competitividad en el ámbito de la innovación y la tecnología, sin embargo Según el Índice de Competitividad Global (GCI) 2015-2016 (Figura 6.3), México ocupa el puesto 57 de 140 naciones evaluadas. Si bien el país tiene calificaciones aceptables en algunos de los pilares que integran el índice, hay que reconocer que, en materia de innovación; preparación tecnológica; Educación y entrenamiento; las calificaciones no eran satisfactorias.

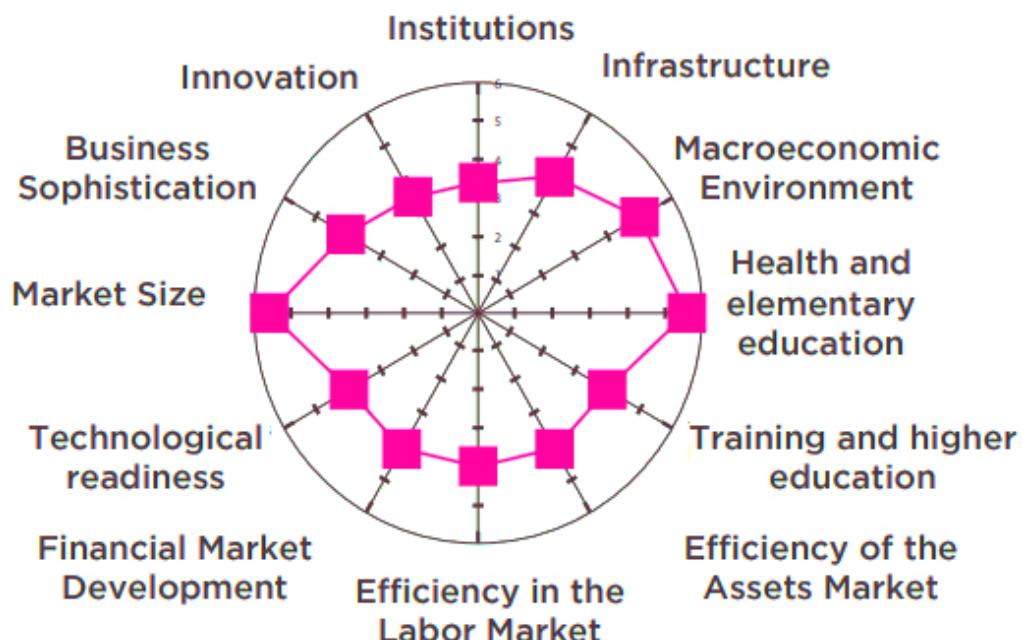


Figura 6.2.- Desempeño de México en el GCI, 2015-2016 [56].

### **6.3 Talento enfocado a la Industria 4.0 en México.**

Según datos del Anuario Estadístico de Educación Superior, para el ciclo escolar 2014-2015, 987.317 estudiantes se matricularon en programas educativos relacionados con ingeniería, manufactura y construcción, tanto a nivel de licenciatura como técnico. Los estados que reportaron un mayor número de estudiantes matriculados en dichas áreas fueron Ciudad de México, Estado de México y Veracruz. Cabe señalar que una gran parte de los estudiantes están matriculados en ingeniería industrial, mecánica, electrónica y tecnología, seguido de especializaciones en manufactura y procesos; estas áreas concentran el 75,6% y el 6,7% de los alumnos matriculados. Adicionalmente, el número de estudiantes matriculados en programas académicos relacionados con las ciencias de la computación, matemáticas y estadística y física fue de 85.590, 18.846 y 7.226, respectivamente.

En cuanto al número de egresados para el período académico 2014-2015, finalizaron sus estudios 128.427 estudiantes, de los cuales Ingeniería Industrial, Mecánica, Electrónica y Tecnología concentraron el mayor número de egresados (80,9% del total), seguidos de Ciencias de la Computación ( 10,5%), y manufactura y procesos (8,3%). En contraste, matemáticas y estadística, así como física, fueron las áreas con menor número de egresados registrado en el período (0,4% y 1,5%, respectivamente).

## 7 DESARROLLO E IMPLEMENTACIÓN

### 7.1 VirtualBox

Debido a que se va a utilizar la distribución de *ROS Melodic Morenia* lanzada en Mayo del 2018 soportada hasta Mayo del 2023, la versión más recomendada para su instalación en un Sistema Operativo (OS) es Ubuntu 18.04.06 LTS (Long Term Support) siendo este el más recomendado por los desarrolladores, sin embargo también está disponible para otros sistemas operativos como Windows y Debian y otros que aún están en su fase experimental.

Para instalar Ubuntu 18.04.06 LTS se utilizará VirtualBox que es un virtualizador de propósito general para hardware x86, dirigido a servidores, escritorios y uso integrado, además que es gratuito y admite los principales sistemas operativos.

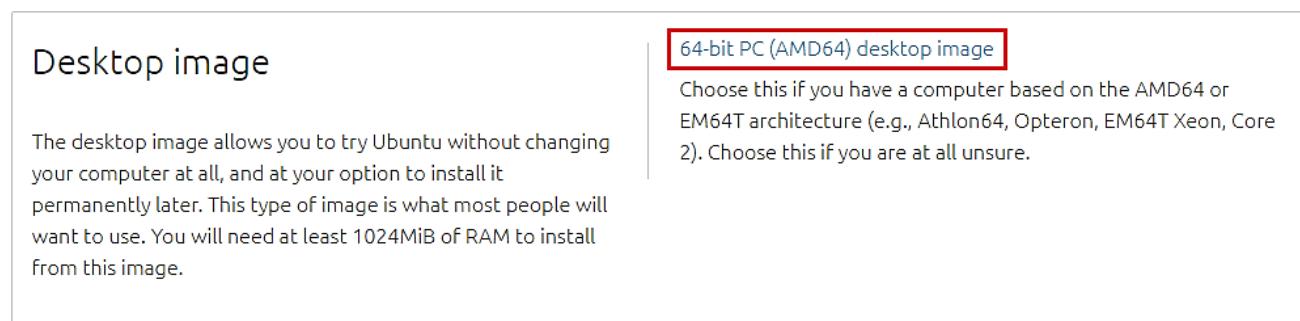
Se puede descargar y seguir los pasos desde <https://www.virtualbox.org/wiki/Downloads>.

### 7.2 Instalación de Ubuntu.

Para descargar Ubuntu se recomienda ir a la página oficial del desarrollador <https://releases.ubuntu.com/18.04/> y para este caso que será virtualizado, se elegirá la opción de imagen de escritorio. Hacer clic en 64-bit PC (AMD64) desktop image como se muestra en la figura 7.2.

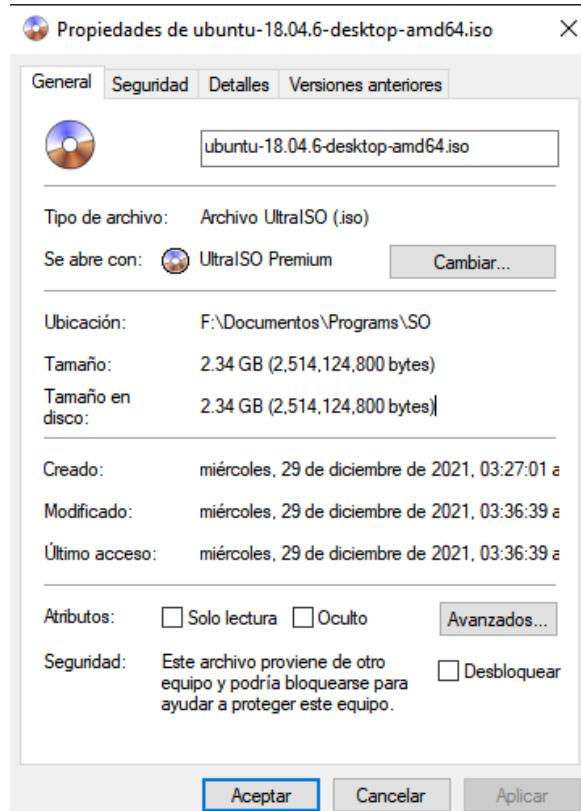
#### Select an image

Ubuntu is distributed on three types of images described below.



**Figura 7.2.- Imagen de escritorio de Ubuntu 18.04.06 LTS.**

En seguida se va a descargar un archivo .ISO (Figura 7.2.1) que se rige por la norma ISO 9660 estandarizada por la Organización Internacional de Normalización (Por sus siglas en inglés ISO) Este formato de medio estándar del sector originalmente se diseñó para especificar las estructuras de volumen y archivo de discos ópticos compactos de memoria sólo de lectura (CD-ROM), y es un formato de medio sólo de lectura, en el cual viene el Sistema operativo.



**Figura 7.2.1.- Propiedades de la imagen de Ubuntu 18.04.06 LTS.**

Para la instalación de Ubuntu 18.04, primero se ejecuta VirtualBox y después hacer clic en el botón “Nueva”



**Figura 7.2.2.- Creación de una Nueva Máquina Virtual.**

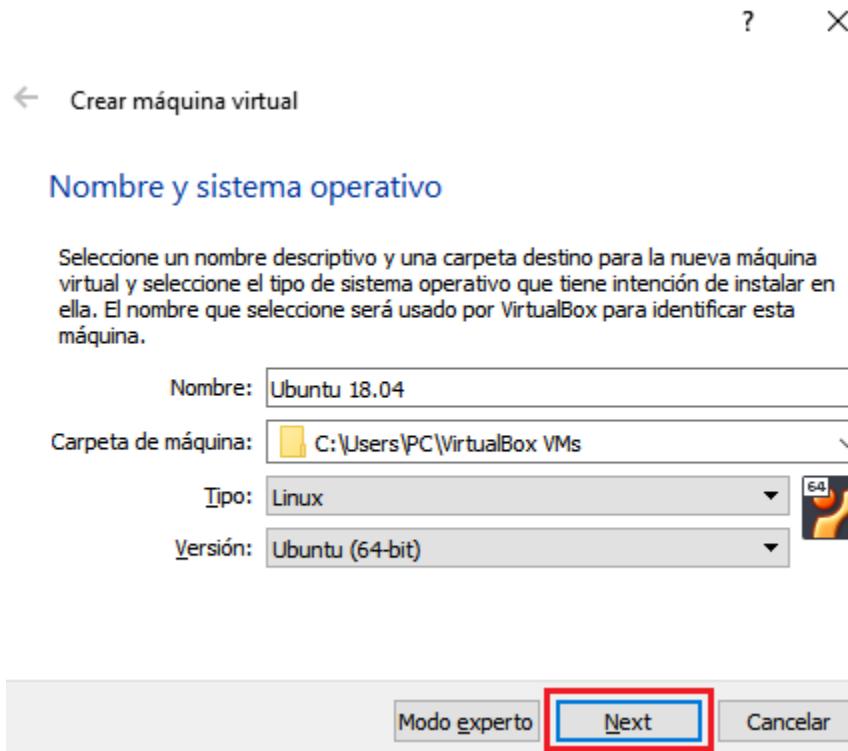
Después poner los siguientes datos de configuración:

**Nombre:** Ubuntu 18.04

**Carpeta de maquina:** \*Seleccionar la dirección de destino para almacenar la Máquina Virtual\*

**Tipo:** Linux

**Versión:** Ubuntu 64-bit



**Figura 7.2.3.-Configuracion de Máquina Virtual.**

Los siguientes pasos van a depender mucho del equipo físico que se tenga y de lo que se vaya a realizar en la máquina virtual. Para este trabajo se tomarán las siguientes configuraciones:

The image contains two side-by-side screenshots of the Oracle VM VirtualBox Manager interface during the 'Create New Virtual Machine' process.

**Left Screenshot (Memory Configuration):**

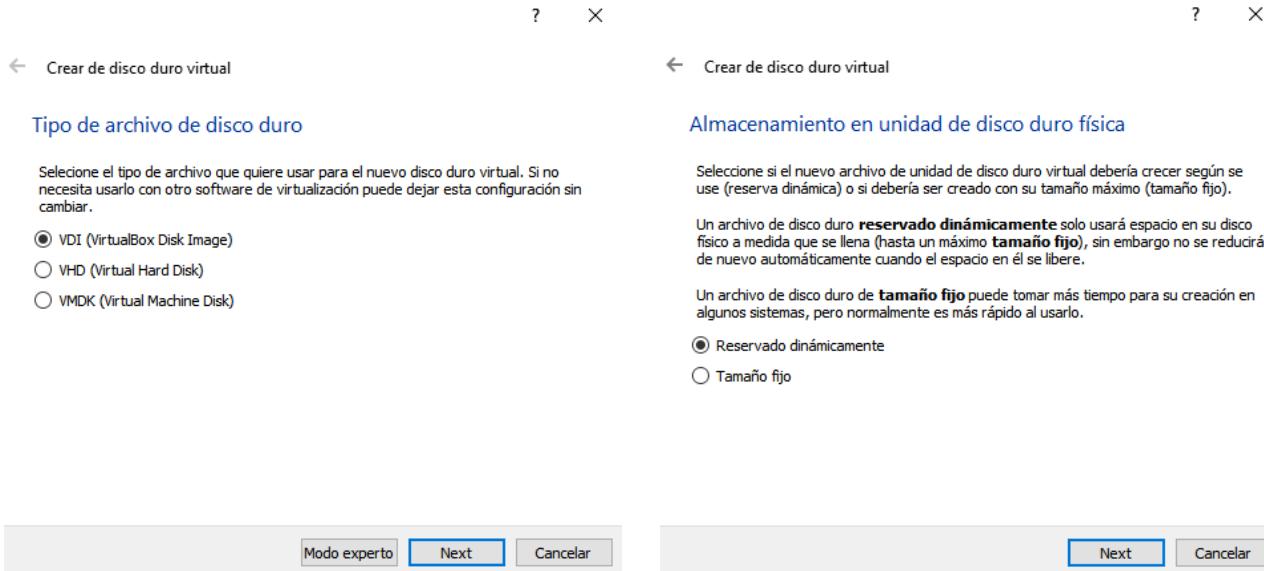
- Header: 'Crear máquina virtual'.
- Section: 'Tamaño de memoria'.
- Description: 'Seleccione la cantidad de memoria (RAM) en megabytes a ser reservada para la máquina virtual.'
- Text: 'El tamaño de memoria recomendado es **1024 MB**'.
- Slider: A horizontal slider with a blue arrow pointing right, currently set to 2048 MB. The scale ranges from 4 MB to 4096 MB.
- Buttons: 'Next' (highlighted with a red box) and 'Cancelar'.

**Right Screenshot (Hard Disk Configuration):**

- Header: 'Crear máquina virtual'.
- Section: 'Disco duro'.
- Description: 'Si desea puede añadir un disco duro virtual a la nueva máquina. Puede crear un nuevo archivo de disco duro o seleccionar uno de la lista o de otra ubicación usando el icono de la carpeta.'
- Text: 'Si necesita una configuración de almacenamiento más compleja puede omitir este paso y hacer los cambios a las preferencias de la máquina virtual una vez creada. El tamaño recomendado del disco duro es **32.00 GB**'.
- Options: Three radio buttons for disk creation:
  - No añadir un disco duro virtual
  - Crear un disco duro virtual ahora
  - Usar un archivo de disco duro virtual existente
- Dropdown: 'Ubuntu 18.04.vdi (Normal, 80.00 GB)'.
- Buttons: 'Crear' (highlighted with a red box) and 'Cancelar'.

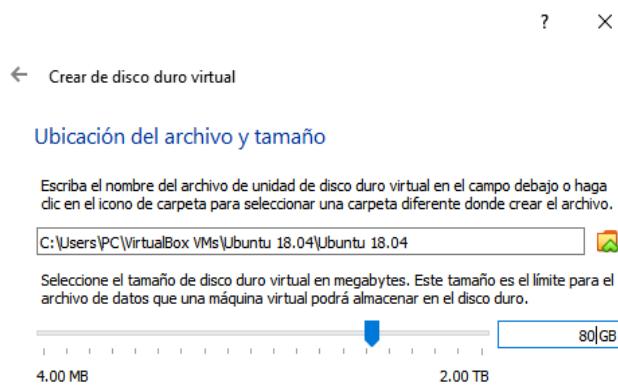
**Figura 7.2.4.- Tamaño de Memoria RAM.**

**Figura 7.2.5.- Creación de Disco Duro Virtual.**



**Figura 7.2.6.- Tipo de archivo del Disco Duro Virtual.**

**Figura 7.2.7.- Almacenamiento en unidad de disco duro física.**



**Figura 7.2.8.- Ubicación del archivo y tamaño del disco duro virtual.**

En este punto ya estará creada la máquina virtual con las configuraciones hechas anteriormente. En la pantalla principal de VirtualBox aparecen todos los detalles de la máquina (Figura 7.2.9), sin embargo aún falta instalar el sistema operativo.

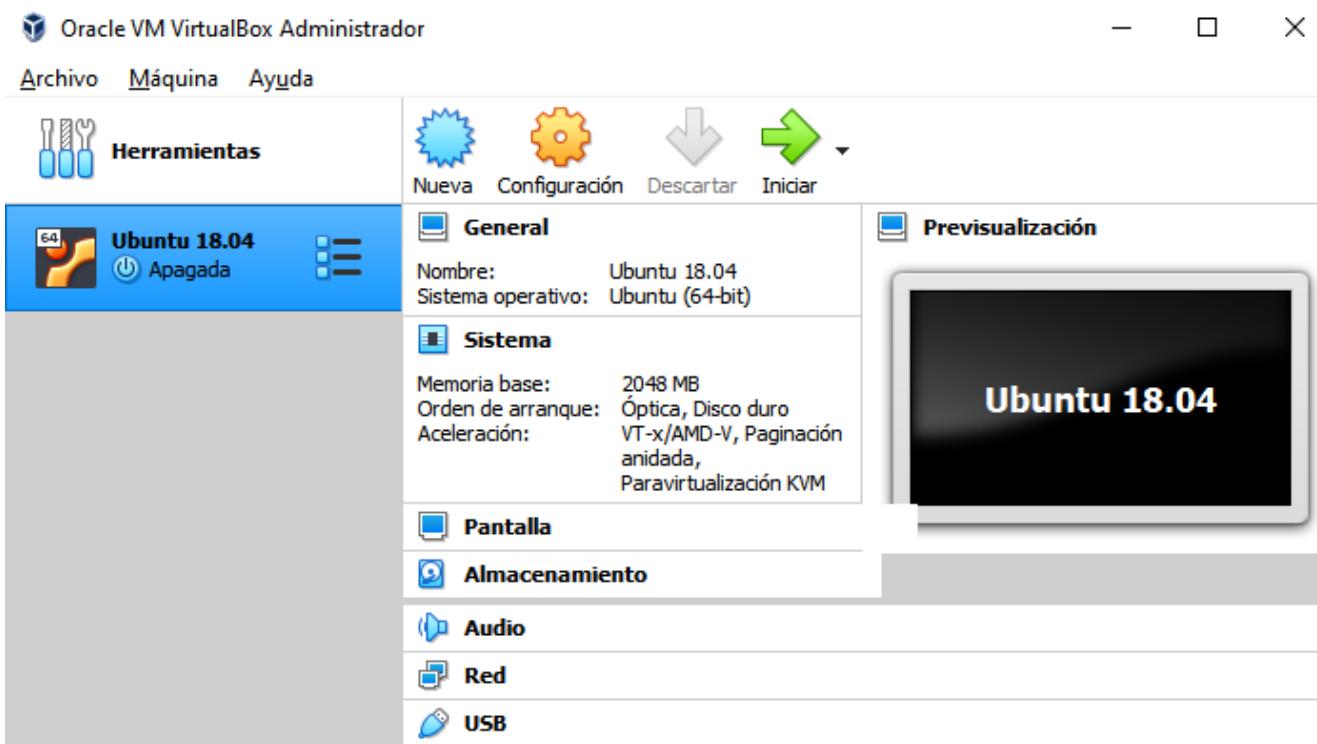


Figura 7.2.9.- Previsualización de las configuraciones para Ubuntu 18.04 en VirtualBox.

Después se va a insertar la imagen del Sistema Operativo. Para esto hacer clic en el engrane de “Configuración>Almacenamiento>Vacío” y se habilitará en los atributos la unidad óptica, después hacer clic en el disco azul (Figura 7.2.10) en la cual se tiene que adjuntar el archivo .ISO que se descargó desde la página oficial de Ubuntu.

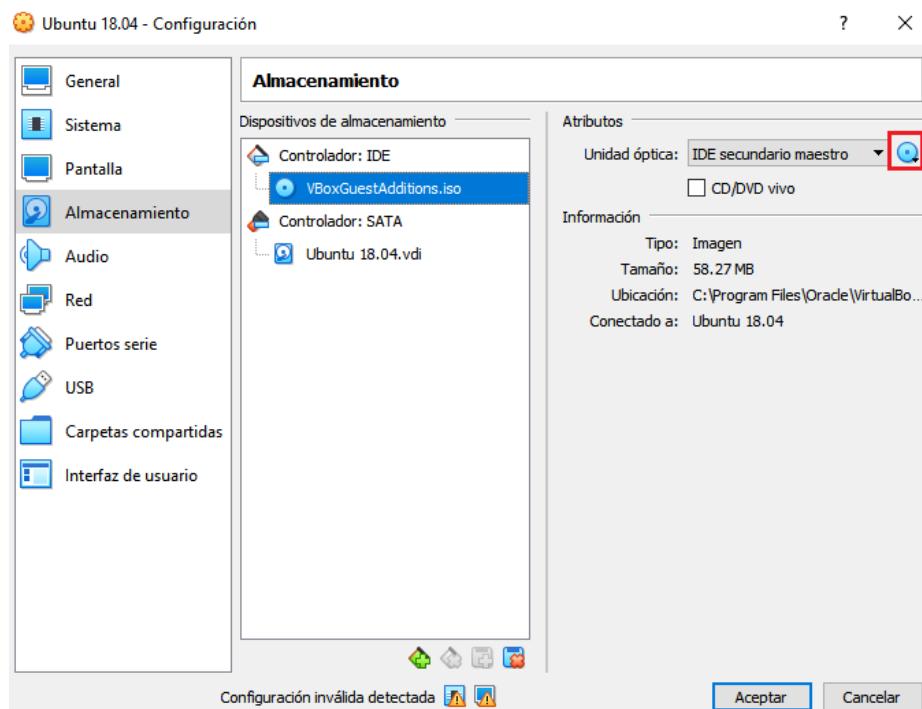
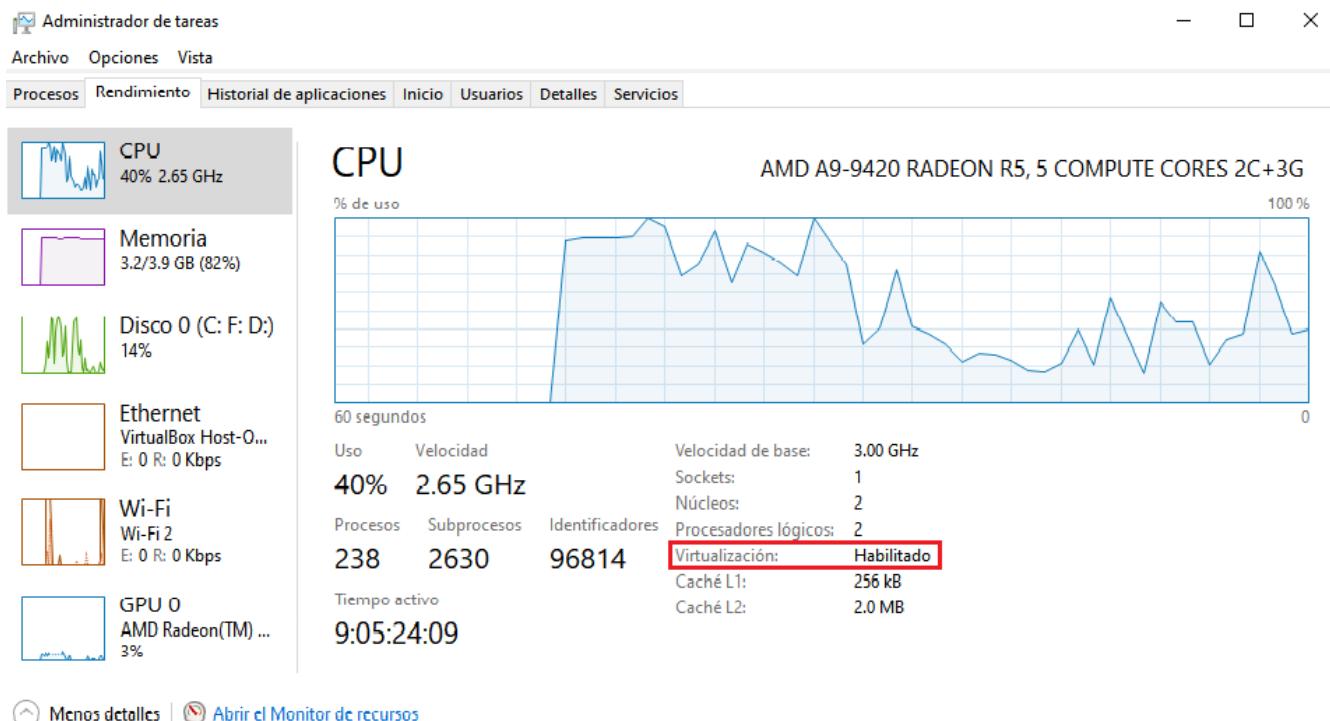


Figura 7.2.10.- Almacenamiento de unidad óptica.

**NOTA:** Para poder usar una máquina virtual se tiene que activar la virtualización en el ordenador. Normalmente esto ya viene activado por defecto, pero no siempre es el caso. Para saber si está habilitada esta opción, ir al administrador de tareas y hacer clic en “rendimiento” y en la esquina inferior derecha viene el estado de la virtualización (Figura 7.2.11). En el caso que no estuviera habilitada, buscar en la BIOS dependiendo el Procesador que se tenga. Para Intel, ir a “Intel Virtualization Technology” y para AMD ir a “SVM Mode” (Se recomienda leer el manual de su equipo).



**Figura 7.2.11.- Estado de Virtualización del equipo.**

Hasta este punto ya se tiene todo listo y configurado para instalar y configurar el sistema operativo. Para eso se deberá correr la Máquina virtual con el botón de “Iniciar” (Figura 7.2.12).



**Figura 7.2.12.- Inicialización de Máquina Virtual.**

Al cargar el sistema, deberá de aparecer la imagen de bienvenida de Ubuntu y dará la opción de elegir idioma y aquí mismo se debe de elegir que se desea hacer con el sistema operativo: “Probar Ubuntu” o “Instalar Ubuntu”, elegir esta última (Figura 7.2.13).



**Figura 7.2.13.- Pantalla de Bienvenida de Ubuntu.**

En este punto se deberá elegir una configuración personal como la zona horaria, idioma, teclado, red, nombre de usuario, contraseña, etc. Despues de esto ya estará listo el sistema operativo para su uso.

## 7.3 Instalación de ROS en Ubuntu.

### 7.3.1 Configuración de Repositorios en Ubuntu.

Antes de ejecutar la instalación de ROS, es necesario saber que muchos de estos programas se almacenan en archivos de software comúnmente denominados *repositorios*. Los repositorios facilitan la instalación de nuevo software, al mismo tiempo que brindan un alto nivel de seguridad, ya que el software se prueba exhaustivamente y se construye específicamente para cada versión de Ubuntu el cual también distingue entre software que es "gratuito" y software que no lo es [58].

Los cuatro repositorios principales son:

**Principal**: software gratuito y de código abierto compatible con Canonical.

**Universe**: software gratuito y de código abierto mantenido por la comunidad.

**Restringido**: controladores patentados para dispositivos.

**Multiverse**: software restringido por derechos de autor o cuestiones legales.

Para esto tiene que asegurarse de tener habilitadas las opciones para instalar software de terceros y elegir desde que servidor se desean hacer las descargas, como se muestra en la Figura 7.3.

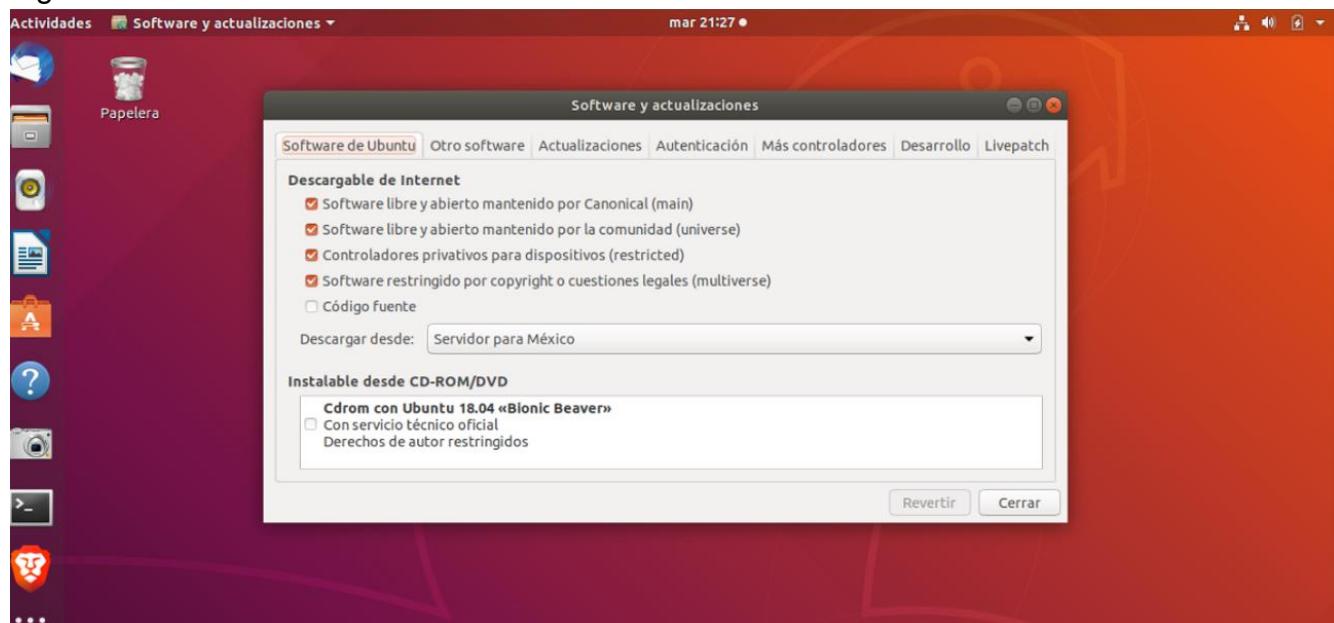


Figura 7.3.- Pestaña de Software de Ubuntu.

Teniendo en cuenta este punto, ya se puede hacer la instalación de ROS Melodic Morenia en Ubuntu. Para esto deben de ejecutarse ciertos comandos por terminal que a continuación se muestran (7.3.2).

**NOTA:** Los comandos fueron obtenidos de la Wiki de ROS:

<http://wiki.ros.org/melodic/Installation/Ubuntu>.

### 7.3.2 Configuración de llaves.

```
$ sudo apt install curl
```

```
$ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
```

### 7.3.3 Instalación.

Primero, asegúrese de que su índice de paquetes Debian esté actualizado:

```
$ sudo apt update
```

Sí todo ha salido bien, la terminal debería verse como la imagen 7.3.1

```
brasil@pc:~$ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
OK
brasil@pc:~$ sudo apt update
Des:1 http://packages.ros.org/ros/ubuntu bionic InRelease [4 680 B]
Des:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Des:3 http://packages.ros.org/ros/ubuntu bionic/main i386 Packages [26.7 kB]
Des:4 http://packages.ros.org/ros/ubuntu bionic/main amd64 Packages [783 kB]
Obj:5 http://mx.archive.ubuntu.com/ubuntu bionic InRelease
Des:6 http://mx.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Des:7 http://mx.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Descargados 1 066 kB en 13s (84.5 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 120 paquetes. Ejecute «apt list --upgradable» para verlos.
brasil@pc:~$ █
```

**Figura 7.3.1.- Comandos ejecutados de Configuración de llaves e índice de paquetes actualizado.**

Hay muchas bibliotecas y herramientas diferentes en ROS. Se proporcionan cuatro configuraciones predeterminadas para que pueda comenzar. También puede instalar paquetes ROS individualmente. En este caso se elegirá la versión completa de ROS, en la cual incluye ROS, rqt, rviz, librerías de robot genéricas, simuladores 2D/3D y percepción 2D/3D. Por lo tanto es el comando más tardado y va a depender mucho de la velocidad de internet.

```
$ sudo apt install ros-melodic-desktop-full
```

En este mismo comando, lo primero que hará es descargar todos los paquetes, después actualizará 12 de ellos y se instalarán 1014 nuevos y 108 se actualizarán, que equivalen a 2259MB (Figura7.3.2). Para hacer esto tenemos que presionar la letra "S" del teclado y dar ENTER.

```

brasil@pc: ~
Archivo Editar Ver Buscar Terminal Ayuda
ros-melodic-std-msgs ros-melodic-std-srvs ros-melodic-stereo-image-proc
ros-melodic-stereo-msgs ros-melodic-tf ros-melodic-tf-conversions
ros-melodic-tf2 ros-melodic-tf2-eigen ros-melodic-tf2-geometry-msgs
ros-melodic-tf2-kdl ros-melodic-tf2-msgs ros-melodic-tf2-py
ros-melodic-tf2-ros ros-melodic-theora-image-transport
ros-melodic-topic-tools ros-melodic-trajectory-msgs
ros-melodic-transmission-interface ros-melodic-turtle-actionlib
ros-melodic-turtle-tf ros-melodic-turtle-tf2 ros-melodic-turtlesim
ros-melodic-urdf ros-melodic-urdf-parser-plugin
ros-melodic-urdf-sim-tutorial ros-melodic-urdf-tutorial
ros-melodic-urdfdom-py ros-melodic-vision-opencv
ros-melodic-visualization-marker-tutorials ros-melodic-visualization-msgs
ros-melodic-visualization-tutorials ros-melodic-viz
ros-melodic-webkit-dependency ros-melodic-xacro ros-melodic-xmldcpp ruby
ruby-did-you-mean ruby-minitest ruby-net-telnet ruby-power-assert
ruby-test-unit ruby2.5 rubygems-integration sbcl sdformat-sdf sgml-base
sip-dev tango-icon-theme tcl tcl-dev tcl-vtk6 tcl8.6 tcl8.6-dev tk tk-dev
tk8.6 tk8.6-blitz2.5 tk8.6-dev ttf-bitstream-vera ttf-dejavu-core unixodbc-dev
uuid-dev vtk6 x11proto-composite-dev x11proto-core-dev x11proto-damage-dev
x11proto-dev x11proto-fixes-dev x11proto-input-dev x11proto-randr-dev
x11proto-scrnsaver-dev x11proto-xext-dev x11proto-xf86vidmode-dev
x11proto-xinerama-dev xml-core xorg-sgml-doctools xtrans-dev zlib1g-dev
Se actualizarán los siguientes paquetes:
  libcurl3-gnutls libexpat1 libglib2.0-0 libglib2.0-bin libicu60 libopenexr22
  libpython3.6 libpython3.6-minimal libpython3.6-stdlib libssl1.1 python3.6
  python3.6-minimal
12 actualizados, 1014 nuevos se instalarán, 0 para eliminar y 108 no actualizado
s.
Se necesita descargar 496 MB/521 MB de archivos.
Se utilizarán 2 259 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S

```

**Figura 7.3.1.- Lista de paquetes actualizados, por actualizar y descargar.**

Cuando la barra verde haya llegado a 100%, ya estará instalado ROS, pero aun faltaría la configuración del entorno.

```

brasil@pc: ~
Archivo Editar Ver Buscar Terminal Ayuda
Preparando para desempaquetar .../48-libxcb-shape0-dev_1.13-2~ubuntu18.04_amd64.
deb ...
Desempaquetando libxcb-shape0-dev:amd64 (1.13-2~ubuntu18.04) ...
Seleccionando el paquete libxcb-xfixes0-dev:amd64 previamente no seleccionado.
Preparando para desempaquetar .../49-libxcb-xfixes0-dev_1.13-2~ubuntu18.04_amd64
.deb ...
Desempaquetando libxcb-xfixes0-dev:amd64 (1.13-2~ubuntu18.04) ...
Seleccionando el paquete libxcb-sync-dev:amd64 previamente no seleccionado.
Preparando para desempaquetar .../50-libxcb-sync-dev_1.13-2~ubuntu18.04_amd64.de
b ...
Desempaquetando libxcb-sync-dev:amd64 (1.13-2~ubuntu18.04) ...
Seleccionando el paquete libxcb-present-dev:amd64 previamente no seleccionado.
Preparando para desempaquetar .../51-libxcb-present-dev_1.13-2~ubuntu18.04_amd64
.deb ...
Desempaquetando libxcb-present-dev:amd64 (1.13-2~ubuntu18.04) ...
Seleccionando el paquete libxshmfence-dev:amd64 previamente no seleccionado.
Preparando para desempaquetar .../52-libxshmfence-dev_1.3-1_amd64.deb ...
Desempaquetando libxshmfence-dev:amd64 (1.3-1) ...
Seleccionando el paquete libxcb-dri2-0-dev:amd64 previamente no seleccionado.
Preparando para desempaquetar .../53-libxcb-dri2-0-dev_1.13-2~ubuntu18.04_amd64.
deb ...
Desempaquetando libxcb-dri2-0-dev:amd64 (1.13-2~ubuntu18.04) ...
Seleccionando el paquete libxcb-glx0-dev:amd64 previamente no seleccionado.
Preparando para desempaquetar .../54-libxcb-glx0-dev_1.13-2~ubuntu18.04_amd64.de
b ...
Desempaquetando libxcb-glx0-dev:amd64 (1.13-2~ubuntu18.04) ...
Seleccionando el paquete x11proto-fixes-dev previamente no seleccionado.
Preparando para desempaquetar .../55-x11proto-fixes-dev_1%3a2018.4-4_all.deb ...
Desempaquetando x11proto-fixes-dev (1:2018.4-4) ...

Progreso: [ 3% ] [#.....]

```

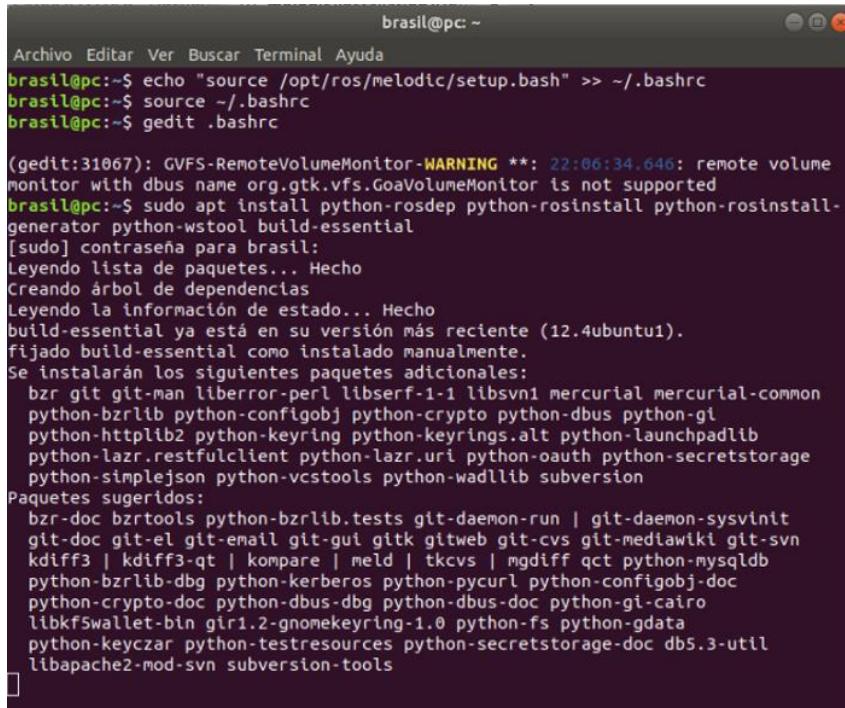
**Figura 7.3.2. Desempaquetado de librerías.**

### 7.3.4 Configuración del entorno.

```
$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc  
$ source ~/.bashrc
```

### 7.3.5 Dependencias para construir paquetes.

```
$ sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-wstool build-essential
```



The screenshot shows a terminal window titled 'brasil@pc: ~'. The user has run several commands to set up ROS dependencies:

```
brasil@pc:~$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc  
brasil@pc:~$ source ~/.bashrc  
brasil@pc:~$ gedit .bashrc  
  
(gedit:31067): GVFS-RemoteVolumeMonitor-WARNING **: 22:06:34.646: remote volume monitor with dbus name org.gtk.vfs.GoaVolumeMonitor is not supported  
brasil@pc:~$ sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-wstool build-essential  
[sudo] contraseña para brasil:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
build-essential ya está en su versión más reciente (12.4ubuntu1).  
fijado build-essential como instalado manualmente.  
Se instalarán los siguientes paquetes adicionales:  
  bazaar git git-man liberror-perl libserf-1-1 libsvn1 mercurial mercurial-common  
  python-bzrlib python-configobj python-crypto python-dbus python-gi  
  python-httplib2 python-keyring python-keyrings.alt python-launchpadlib  
  python-lazr.restfulclient python-lazr.uri python-oauth python-secretstorage  
  python-simplejson python-vcstools python-wadllib subversion  
Paquetes sugeridos:  
  bzr-doc bzrtools python-bzrlib.tests git-daemon-run | git-daemon-sysvinit  
  git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn  
  kdiff3 | kdiff3-qt | kompare | meld | tkcvs | mgdiff qct python-mysqldb  
  python-bzrlib-dbg python-kerberos python-pycurl python-configobj-doc  
  python-crypto-doc python-dbus-dbg python-dbus-doc python-gi-cairo  
  libkf5wallet-bin gir1.2-gnomekeyring-1.0 python-fs python-gdata  
  python-keyczar python-testresources python-secretstorage-doc db5.3-util  
  libapache2-mod-svn subversion-tools
```

Figura.- 7.3.3.- Dependencias para construir paquetes.

### 7.3.6 Inicializar rosdep.

Antes de poder usar muchas herramientas de ROS, se tiene que inicializar rosdep para instalar fácilmente las dependencias del sistema para la fuente que se desea compilar y es necesario para ejecutar algunos componentes principales en ROS.

Para instalar rosdep se ejecutan los siguientes comandos:

```
$ sudo apt install python-rosdep  
$ sudo rosdep init  
$ rosdep update
```

Al finalizar la ejecución de estos comandos, ya estará todo listo y ahora solo queda verificar por terminal que ROS ya está instalado correctamente. Para eso ejecutamos el siguiente comando:

```
roscore
```

```
roscore http://pc:11311/
Archivo Editar Ver Buscar Terminal Ayuda
Add distro "noetic"
Add distro "rolling"
updated cache in /home/brasil/.ros/rosdep/sources.cache
brasil@pc:~$ roscore
... logging to /home/brasil/.ros/log/3d6b7564-945f-11ec-81c2-080027e4f4f1/roslau
nch-pc-1953.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://pc:39505/
ros_comm version 1.14.12

SUMMARY
=====
PARAMETERS
* /rosdistro: melodic
* /rosversion: 1.14.12

NODES

auto-starting new master
process[master]: started with pid [1971]
ROS_MASTER_URI=http://pc:11311/

setting /run_id to 3d6b7564-945f-11ec-81c2-080027e4f4f1
process[rosout-1]: started with pid [1982]
started core service [/rosout]
```

Figura.- 7.3.4.- Resumen de Instalación de ROS.

En la figura 7.3.4 se muestra el resumen de la instalación de ros, Mostrando la distribución ROS Melodic en su versión 1.14.12 y el nodo maestro: ROS\_MASTER. Para terminar el proceso de roscore, se debe presionar la combinación de teclas “Ctrl+C” y cada vez que se requiera de la ejecución de un programa se vuelve a ejecutar el comando, ya que es la base de ROS.

Antes de empezar con ROS de lleno, se tienen que conocer algunos conceptos base que se estarán referenciando en este trabajo. Estos niveles y conceptos se resumen a continuación y se analizan cada uno de ellos con mayor detalle.

- Nivel del sistema de archivos.
- Nivel del gráfico de computación.
- Nivel de la comunidad.

**NOTA: Todos los conceptos y niveles están en inglés y así se estarán utilizando en cada comando que se escriba en la terminal. Todas las imágenes y conceptos fueron extraídas de <http://wiki.ros.org/es/ROS/Conceptos>.**

### 7.3.7 Nivel del sistema de archivos.

Los conceptos de nivel de sistema de archivos cubren principalmente los recursos de ROS que encuentra en el disco, como:

- **Packages**: Los paquetes son la unidad principal para organizar el software en ROS. Un paquete puede contener procesos de tiempo de ejecución de ROS (*nodos*), una biblioteca dependiente de ROS, conjuntos de datos, archivos de configuración o cualquier otra cosa que se organice de manera útil en conjunto. Los paquetes son el elemento de compilación y el elemento de lanzamiento más atómico en ROS. Lo que significa que lo más granular que puede crear y lanzar es un paquete.
- **Metapackages** : los metapackages son paquetes especializados que solo sirven para representar un grupo de otros paquetes relacionados. Más comúnmente, los metapackages se usan como un marcador de posición compatible con versiones anteriores para las pilas rosbuild convertidas.
- **Package Manifests** : los manifiestos (package.xml) proporcionan metadatos sobre un paquete, incluido su nombre, versión, descripción, información de licencia, dependencias y otra metainformación, como paquetes exportados.
- **Repositorios**: una colección de paquetes que comparten un sistema VCS común. Los paquetes que comparten un VCS comparten la misma versión y se pueden lanzar juntos usando la herramienta de automatización de lanzamiento catkin bloom . A menudo, estos repositorios se asignarán a pilas rosbuild convertidas. Los repositorios también pueden contener un solo paquete.
- **Message(msg)types**: las descripciones de mensajes, almacenadas en my\_package/msg/MyMessageType.msg , definen las estructuras de datos para los mensajes enviados en ROS.
- **Service(srv)types**: Las descripciones de los servicios, almacenadas en my\_package/srv/MyServiceType.srv , definen las estructuras de datos de solicitud y respuesta para los servicios en ROS.

### 7.3.8 Nivel del gráfico de computación.

*Computation Graph* es la red peer-to-peer de procesos ROS que procesan datos juntos. Los conceptos básicos de *Computation Graph* de ROS son *nodos*, *maestro*, *servidor de parámetros*, *mensajes*, *servicios*, *temas* y *paquetes*, todos los cuales proporcionan datos al gráfico de diferentes maneras. Estos conceptos se implementan en el repositorio ros\_comm.

- **Nodes**: los nodos son procesos que realizan cálculos. ROS está diseñado para ser modular en una escala de grano fino; un sistema de control de robot generalmente comprende muchos nodos. Por ejemplo, un nodo controla un telémetro láser, un nodo controla los motores de las ruedas, un nodo realiza la localización, un nodo realiza la planificación de rutas, un nodo proporciona una vista gráfica del sistema, etc. Un nodo ROS se escribe con el uso de una biblioteca de cliente ROS, como roscpp o rospy .
- **Master**: El ROS Master proporciona el registro de nombres y la búsqueda del resto del Gráfico de Cómputo. Sin el Maestro, los nodos no podrían encontrarse, intercambiar mensajes o invocar servicios.

- **Parameter server:** El Servidor de Parámetros permite que los datos sean almacenados por clave en una ubicación central. Actualmente forma parte del Máster.
- **Messages:** los nodos se comunican entre sí pasando mensajes. Un mensaje es simplemente una estructura de datos, que comprende campos escritos. Se admiten los tipos primitivos estándar (entero, punto flotante, booleano, etc.), al igual que las matrices de tipos primitivos. Los mensajes pueden incluir matrices y estructuras anidadas arbitrariamente (al igual que las estructuras C).
- **Topics:** los mensajes se enrutan a través de un sistema de transporte con semántica de publicación/suscripción. Un nodo envía un mensaje *publicándolo* en un tema determinado. El tema es un nombre que se utiliza para identificar el contenido del mensaje. Un nodo que esté interesado en cierto tipo de datos se *suscribirá* al tema correspondiente. Puede haber múltiples publicadores y suscriptores simultáneos para un solo tema, y un solo nodo puede publicar y/o suscribirse a múltiples temas. En general, los editores y los suscriptores no conocen la existencia de los demás. La idea es desvincular la producción de información de su consumo. Lógicamente, uno puede pensar en un tema como un bus de mensajes fuertemente tipificado. Cada bus tiene un nombre y cualquiera puede conectarse al bus para enviar o recibir mensajes, siempre que sean del tipo correcto.
- **Services:** el modelo de publicación/suscripción es un paradigma de comunicación muy flexible, pero su transporte unidireccional de muchos a muchos no es apropiado para las interacciones de solicitud/respuesta, que a menudo se requieren en un sistema distribuido. La solicitud/respuesta se realiza a través de servicios, que están definidos por un par de estructuras de mensajes: una para la solicitud y otra para la respuesta. Un nodo proveedor ofrece un servicio bajo un nombre y un cliente utiliza el servicio enviando el mensaje de solicitud y esperando la respuesta. Las bibliotecas de clientes de ROS generalmente presentan esta interacción al programador como si fuera una llamada a un procedimiento remoto.
- **Bags:** las bolsas son un formato para guardar y reproducir datos de mensajes ROS. Las bolsas son un mecanismo importante para almacenar datos, como datos de sensores, que pueden ser difíciles de recopilar pero que son necesarios para desarrollar y probar algoritmos.

### 7.3.9 Nivel de la comunidad.

Los conceptos de ROS Community Level son recursos de ROS que permiten a comunidades separadas intercambiar software y conocimientos. Estos recursos incluyen:

- **Distribuciones:** Las distribuciones ROS son colecciones de pilas versionadas que puede instalar. Las distribuciones desempeñan un papel similar al de las distribuciones de Linux: facilitan la instalación de una colección de software y también mantienen versiones coherentes en un conjunto de software.
- **Repositorios:** ROS se basa en una red federada de repositorios de código, donde diferentes instituciones pueden desarrollar y lanzar sus propios componentes de software de robot.
- **El Wiki de ROS:** El Wiki de la comunidad de ROS es el foro principal para documentar información sobre ROS. Cualquiera puede registrarse para obtener una cuenta y

contribuir con su propia documentación, proporcionar correcciones o actualizaciones, escribir tutoriales y más.

- **Sistema de tickets de error:** consulte Tickets para obtener información sobre los tickets de archivo.
- **Listas de correo:** La lista de correo de usuarios de ros es el principal canal de comunicación sobre nuevas actualizaciones de ROS, así como un foro para hacer preguntas sobre el software de ROS.
- **Respuestas de ROS:** Un sitio de preguntas y respuestas para responder a sus preguntas relacionadas con ROS.
- **Blog:** El blog de ros.org proporciona actualizaciones periódicas, incluidas fotos y videos.

## 7.4 Sistema Mecánico

### 7.4.1 Vehículo a Escala

Tomando en cuenta lo estudiado en el capítulo 2 y con los objetivos de este trabajo, se propone un vehículo a escala que posea características similares a las de un vehículo real, donde se puedan implementar los sistemas de navegación, percepción, creación de mapas, y seguimiento de trayectorias, en un entorno simulado.

Considerando las dimensiones de los componentes restantes del resto del sistema AGV, la escala de vehículo que se eligió fue la siguiente:

#### Escala 1/10

- Longitud: 438 mm
- ancho: 190 mm
- altura: 124 mm
- Distancia entre ejes: 257 mm
- diámetro del neumático: 27/67 mm (delantero y trasero)

Investigando en varias tiendas de modelismo los resultados no eran tan favorables, ya que estos vehículos muchas veces son de colección y el precio es muy elevado. Durante esta investigación se encontraron varios vehículos que cumplen la relación de bajo costo y duradero y al final se optó por un modelo de la empresa TAMIYA, la cual tiene un Kit de ensamblaje que recrea la forma de un coche real y el cuerpo está hecho de policarbonato, lo cual lo hace liviano y duradero

### 7.4.2 Suspensión.

Este sistema en los vehículos a escala está diseñado especialmente para movimientos bruscos o para ser manipulados en terrenos que demandan de dicho sistema, muchos otros simplemente están ahí para que luzca más estético o lo más cercano a una replica de un modelo real.

Debido a que el vehículo no va a cargar mucho peso y estará en entornos “controlados”, se tomó en cuenta que amortiguadores del vehículo propuesto no tuvieran mucha relación con la función estructural.

### **7.4.3 Dirección**

Hay una variedad de locomociones, las cuales algunas están en permanencia con la superficie donde se esté moviendo el robot, mientras otras simulan el comportamiento de pies o patas.

Para dar dirección a las maniobras y giros del AGV que va a realizar en el entorno controlado y además de darle una apariencia a un vehículo real, se eligió la locomoción Ackerman, que para la dirección se va a utilizar un servomotor, ya que este tiene un ángulo de 180°, aunque también los hay en 360°. La implementación de un servomotor en un vehículo a escala es fácil de controlar y de montar.

### **7.4.4 Tracción**

Dado el punto anterior y tomando en cuenta que el Kit del vehículo a escala incluye un Motor DC Brushless 540 (Motor sin escobillas), no se hará ningún cambio. A parte que no se requiere de un alto torque para el funcionamiento del AGV

## **7.5 Electrónica y Eléctrica del Vehículo**

### **7.5.1 Módulos.**

Debido a que el vehículo va a estar recibiendo señales de la computadora y de un Radio Control (por sus siglas en inglés RC), para la teleoperación, se tiene que hacer uso de un servo multiplexor que se va a encargar de comparar las señales y dependiendo de estas, va a permitir elegir entre el control autónomo y control manual.

Por otra parte, el servomotor no se puede conectar directamente al sistema de control, ya que no manejan la misma corriente. Se propone un servo multiplexor RC de 4 canales de la empresa Pololu por su tamaño compacto y también que hace una selección de la salida proveniente de la fuente maestra o esclava.

### **Otros**

Por otro lado, se estarán utilizando herramientas, cables, conectores, dispositivos de medición, etc. Con los que ya se cuenta.

## 7.6 Integración por Computadora de Placa Única.

Actualmente existen muchas computadoras de placa única (SBC por sus siglas en inglés) en el mercado y cada una ofrece una ventaja única, por este motivo se hace difícil elegir una SBC a la hora de hacer un proyecto debido a que también el mercado es muy cambiante. Una SBC cotidiana se distingue por tener un kit de desarrollo que incluye una plataforma completa de hardware y software necesario para que funcione. A continuación se hace una comparación de dos SBC con más popularidad en el mercado y se enlistan los requerimientos mínimos que se necesitan para este proyecto:

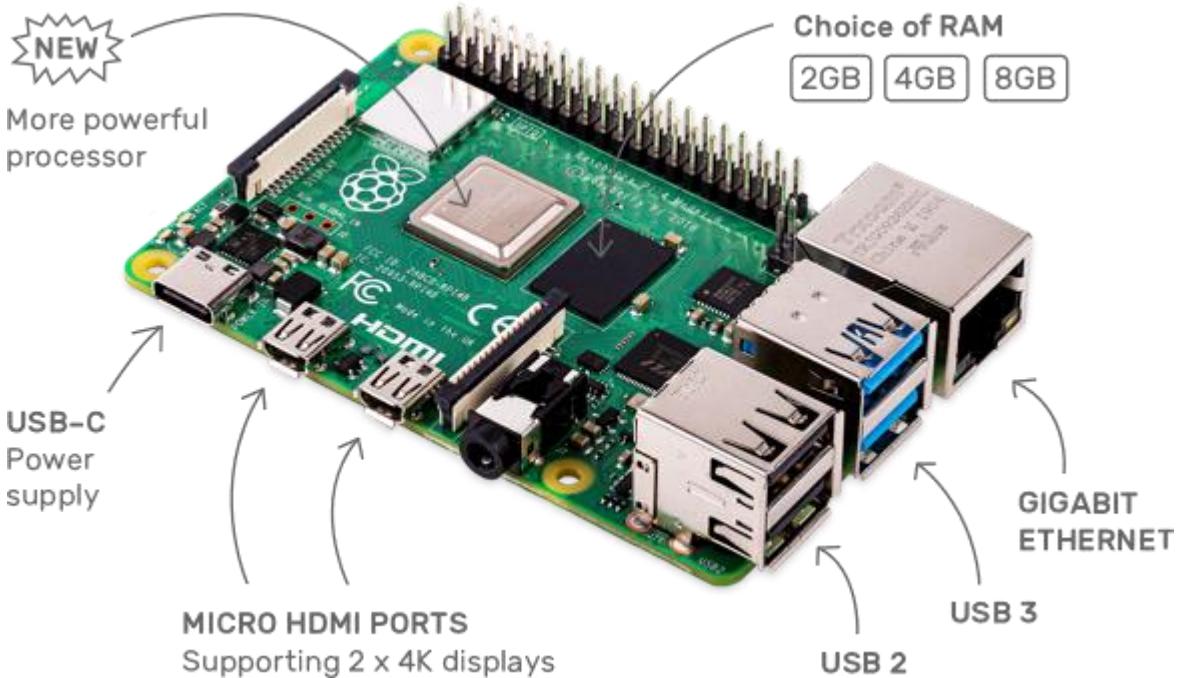
- Ejecución de un sistema operativo de alto nivel (Windows, Linux, Android, etc).
- Lista para usar.
- Comunicación inalámbrica y por cable.
- Ecosistema de desarrolladores.
- Soporte para ROS.
- Adaptable a robótica.
- Compacto y bajo consumo.
- Bajo costo.
- Aplicaciones básicas de AI (Inteligencia Artificial).

### 7.6.1 Raspberry Pi.

Raspberry Pi es una gran placa base de desarrollo para computadoras pequeñas y IoT. No solo es un dispositivo de IoT de bajo consumo, sino también una buena herramienta de creación de prototipos. Raspberry Pi 4 en la cual se centrará esta comparación, tiene las siguientes características.

- **PROCESADOR** ARM Cortex-A72
- **FRECUENCIA DE RELOJ** 1,5 GHz
- **GPU** VideoCore VI (con soporte para OpenGL ES 3.x)
- **MEMORIA** 1 GB / 2 GB / 4 GB LPDDR4 SDRAM
- **CONECTIVIDAD** Bluetooth 5.0, Wi-Fi 802.11ac, Gigabit Ethernet
- **PUERTOS** GPIO 40 pines 2 x micro HDMI 2 x USB 2.0 2 x USB 3.0 CSI (cámara Raspberry Pi) DSI (pantalla táctil) Micro SD Conector de audio jack USB-C (alimentación)

Dentro de la comunidad desarrolladora de raspberry pi, se encuentra un ambiente muy sólido, ya que existen proyectos que son excelentes para principiantes y no requieren conocimientos previos de programación, impresión 3D, electrónica, etc. También este modelo está disponible en tres versiones: 2GB, 4GB y 8GB. Los periféricos a destacar se pueden apreciar en la Figura 7.6.1.



**Figura 7.6.1. Periféricos de Raspberry Pi [59].**

## 7.6.2 Jetson Nano.

NVIDIA ha lanzado muchos tipos de placas base de IoT, entre las cuales NVIDIA Jetson Nano es la placa base más reciente como kit de desarrollo, esta SBC proporciona todas las entradas y conexiones necesarias para el prototipo de soluciones IoT. Jetson Nano tiene las siguientes características.

- CPU: procesador ARM Cortex-A57 MPCore de cuatro núcleos.
- GPU: NVIDIA Maxwell con 128 núcleos NVIDIA CUDA.
- Memoria: LPDDR4 de 4 GB de 64 bits.
- Tarjeta gráfica: salida HDMI y DisplayPort.
- USB: 4 puertos USB 3.
- E / S: I2C, SPI, UART y encabezado GPIO compatible con Raspberry Pi.

A diferencia de Rspberry Pi, su comunidad de desarrollo de proyectos es un poco escasa y la existente está en inglés, pero las comunidades hacen que sea fácil la implementación de los proyectos compartiendo la misma línea de proyectos que van desde principiantes que no requieren muchos conocimientos, hasta los más complejos.

Siguiendo con la comparativa de estas dos SBC, pareciera que no hay diferencia debido a que sus principales funciones son muy similares. Ambas tienen un procesador ARM y se pueden encontrar en versiones de 4GB, sin embargo, la mayor diferencia es que NVIDIA Jetson Nano incluye una Unidad de Procesamiento Gráfico (GPU por sus siglas en inglés) más potente y de mayor rendimiento, y aunque el precio es aún más alto, sigue siendo razonable. Incluso esta GPU puede estar algo cercana en comparación con la Nintendo Switch. Debido a esta última característica, se propone Jetson Nano Developer Kit como unidad principal de procesamiento. En la Figura 7.6.2 se puede apreciar la SBC.



**Figura 7.6.2.- Jetson Nano Developer Kit [34]**

### **Elementos para comenzar a utilizar Jetson Nano Developer Kit**

#### **Tarjeta micro SD:**

El Jetson Nano Developer Kit utiliza una tarjeta microSD como dispositivo de arranque y almacenamiento principal. Es importante tener una tarjeta que sea lo suficientemente rápida y grande para cualquier proyecto; el mínimo recomendado es una tarjeta de Velocidad Ultra Alta (UHS-1 por sus siglas en inglés) de 32 GB.

#### **Fuente de alimentación:**

Para alimentar el kit de desarrollador se deberá contar con una fuente de alimentación de buena calidad que pueda entregar 5V=2A en el puerto Micro-USB del kit de desarrollador.

Para continuar con la configuración de la Jetson, puede consultar toda la información y seguir los pasos en el siguiente enlace: <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#prepare>

## **7.7 Comunicación**

La comunicación para la telemetría y entrenamiento del robot que requiera seguir unos puntos establecidos en un entorno por el operador del vehículo o también para hacer el mapeo de algún recinto, se utilizará comunicación inalámbrica mediante un transmisor y receptor a radio control de 2.4GHz, que estará controlando al Motor DC Brushless 540 y al servomotor para la dirección. También puede servir un joystick o control de una consola como Play Station o Xbox que tenga soporte en ROS.

Po otro lado, la Jetson Nano no proporciona una placa de WI-FI integrada y se requiere instalar una placa WI-FI y sus antenas. Esto servirá también para hacer teleoperación, actualizaciones o comunicaciones con otros equipos.

De acuerdo al nivel grafico de cálculo de ROS, Los nodos se conectan a otros nodos directamente; el maestro solo proporciona información de búsqueda, como un servidor DNS. Los nodos que se suscriben a un tema solicitarán conexiones de los nodos que publican ese tema y establecerán esa conexión a través de un protocolo de conexión acordado. El protocolo más común usado en un ROS se llama TCPROS , que usa sockets TCP/IP estándar.

## 7.8 Fuente de Energía

Para alimentar al vehículo, se va a separar la alimentación de control de la Jetson Nano y la de potencia de los motores. Tomando en cuenta que es un vehículo a escala y que estos utilizan baterías de Niquel-Metalhidruro (NiMH) ya que son menos peligrosas para el medio ambiente y que no contienen cadmio (un metal toxicoo), se propone una batería NiMH de 7.2V a 30000 mAh junto con un cargador de 7.2V a 12V para baterías NiMH/NiCd que serán parte de la alimentación de los motores. Para la Jetson Nano se propone una power bank de al menos 5V a 2A mientras que el vehículo esté en movimiento. Cuando la Jetson Nano requiera una conexión de monitor, teclado y mouse, se tendrá que utilizar una fuente de alimentación recomendada por NVIDIA, en este caso recomienda una fuente conmutada de 5V 4A.

## 7.9 Percepción, localización y Mapeo

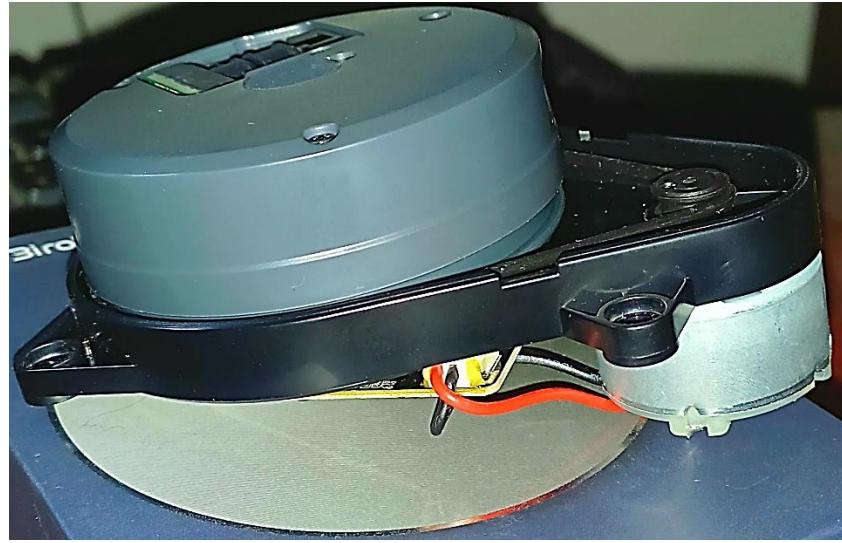
### 7.9.1 Cámara

Este apartado es muy importante para la autonomía del vehículo, ya que permite obtener un control e interacción de datos del entorno real donde se está moviendo el vehículo. Por cuestiones prácticas y económicas, se simulará un controlador de videojuegos desarrollado por la empresa Microsoft para la consola de videojuegos Xbox 360. Este dispositivo cuenta con una cámara RGB y un sensor de profundidad, que juntos proporcionan la captura de objetos en movimiento o estáticos en 3D.

Para la implementación física del vehículo y la percepción, se propone una cámara que se ajuste a un buen presupuesto y también que sea compatible con la Jetson Nano. El módulo de cámara de Raspberry Pi es una buena opción para este vehículo, ya que cuenta con 8 megapíxeles y es compacta para poder adaptarla al vehículo a escala.

### 7.9.2 LiDAR

Para el mapeo se va a utilizar un sensor *LiDAR* (por sus siglas en inglés, *Laser Imaging Detection and Ranging*), que va a permitir determinar la distancia entre un emisor laser a un objeto o superficie, calculando el retraso entre la emisión del pulso y detección a través de la señal reflejada. Se cuenta con el sensor LiDAR Delta-2A de la empresa 3irobotix (Figura 7.9.2) con un escaneo de 360° y un rango de 8m. Los datos de nubes de puntos 2D generados, se pueden utilizar para mapeo, localización, navegación y modelado de objetos y entornos.



**Figura 7.9.2.- Sensor LiDAR Delta-2A de la empresa 3irobotix.**

En la Figura 7.9.3 se aprecia el motor que hace girar al sensor, la frecuencia de escaneo típica de Delta-2A es de 4 ~ 10hz (360rpm), en la cual la resolución angular es de 0,3 ~ 0,8 °. Tiene un excelente rendimiento en todo tipo de ambientes interiores y exteriores sin luz solar directa.



**Figura 7.9.3.- Vista inferior de Delta-2A con motor.**

Este sensor viene con dos cables de alimentación: uno para el láser y otro para el motor. El controlador de este sensor es mediante un módulo que incorpora un microcontrolador STM32F042K6T6 y hace la comunicación con el sensor mediante RS232 lo que hace que solamente conectando los dos puertos USB al ordenador y eligiendo un puerto COM, el sensor podrá ser utilizado, sin embargo antes hay que tener un par de herramientas dadas por el

fabricante las cuales se pueden descargar del siguiente enlace: <Https://lidar.oss-cn-beijing.aliyuncs.com/Lidar.rar>

Al terminar la descarga deberá aparecer una carpeta llamada: Lidar.rar al abrirla se encontraran tres versiones del sensor Delta, en este caso se elegirá la carpeta Delta-2A (Figura 7.9.4).

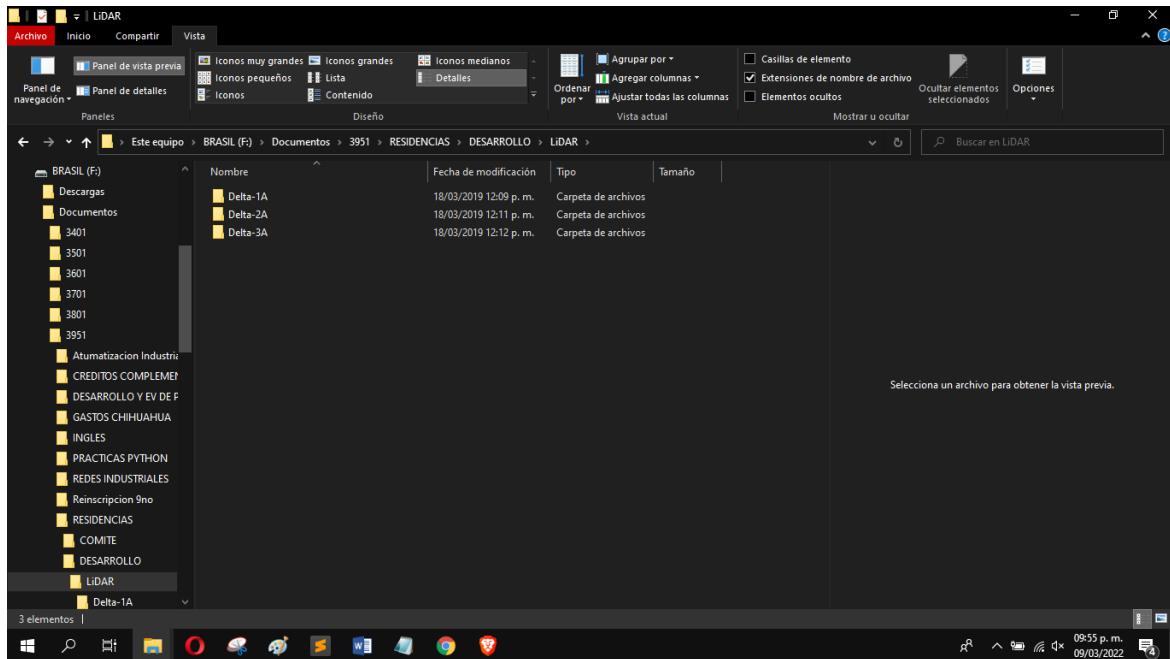


Figura 7.9.4.- Carpeta extraída de Delta-2A.

A continuación se podrá ejecutar el archivo .exe y dar clic en “Sí”

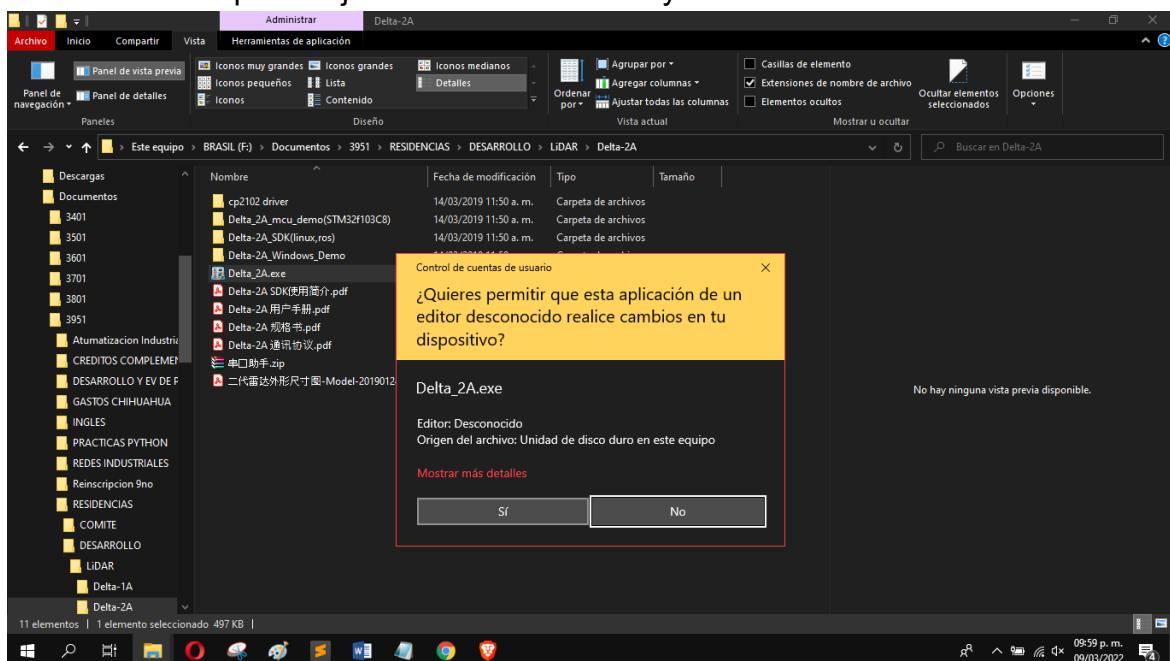
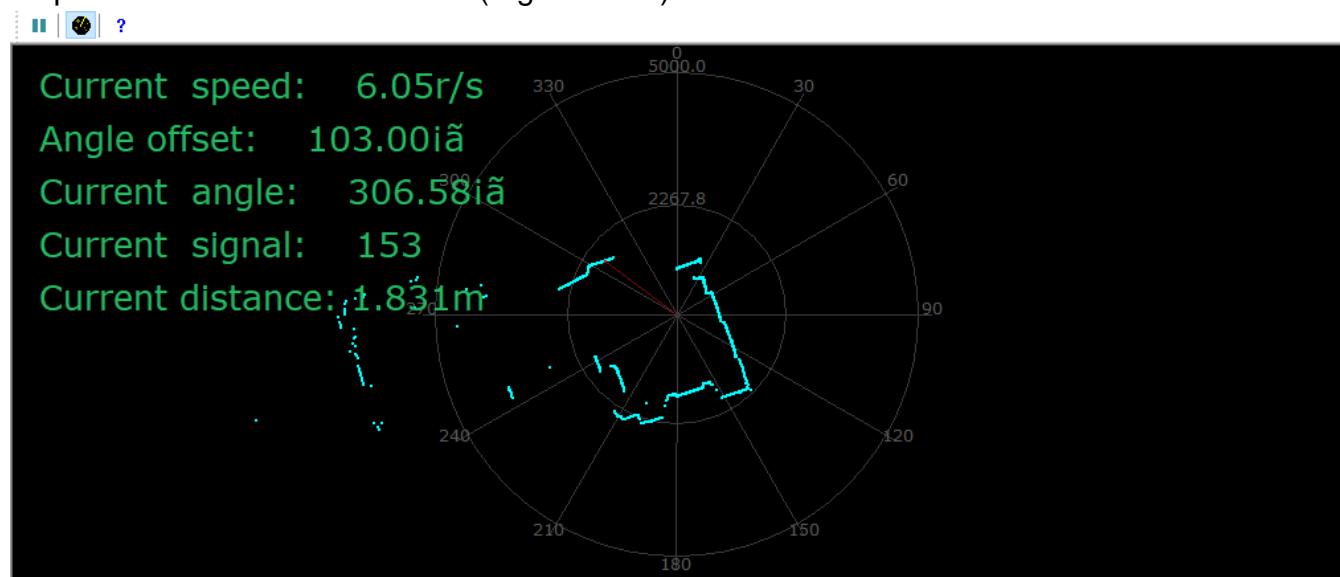


Figura 7.9.5.- Ejecución del archivo .exe.

Posteriormente pedirá elegir el puerto COM y despues dar clic en “aceptar” y el sensor empezará a escanear el entorno (Figura 7.9.6)

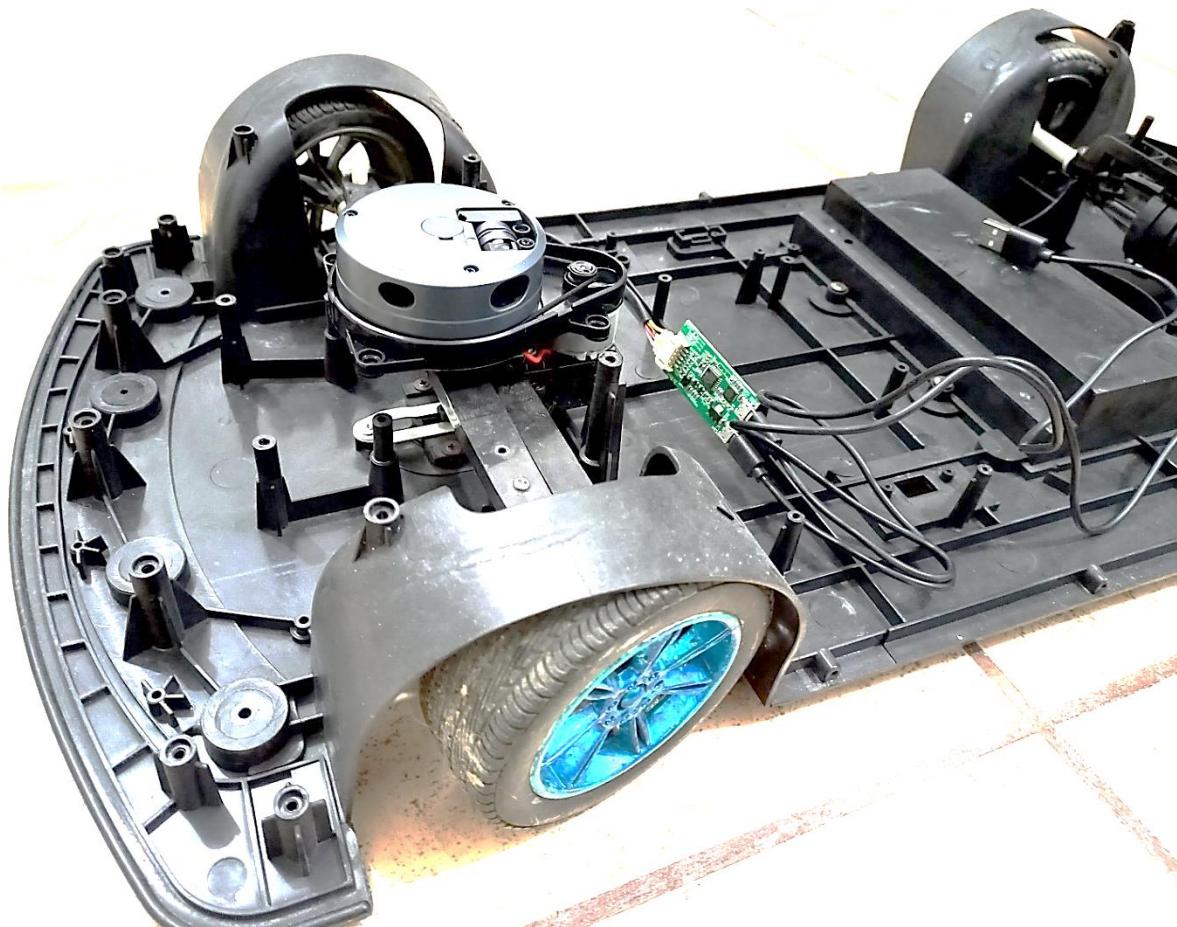


**Figura 7.6.6.- Primer escaneo del entorno con el sensor LiDAR Delta-2A.**

Para hacer el mapeo de un recinto, se procedió a montar el sensor en un carro RC a escala 1/7 el cual va a incorporar un ordenador personal portátil (ya que no se cuenta aún con la Jetson Nano). Posteriormente se empezará a desplazar el vehículo por una superficie de 3x2.60m (Figura 7.9.7 y 7.9.8)

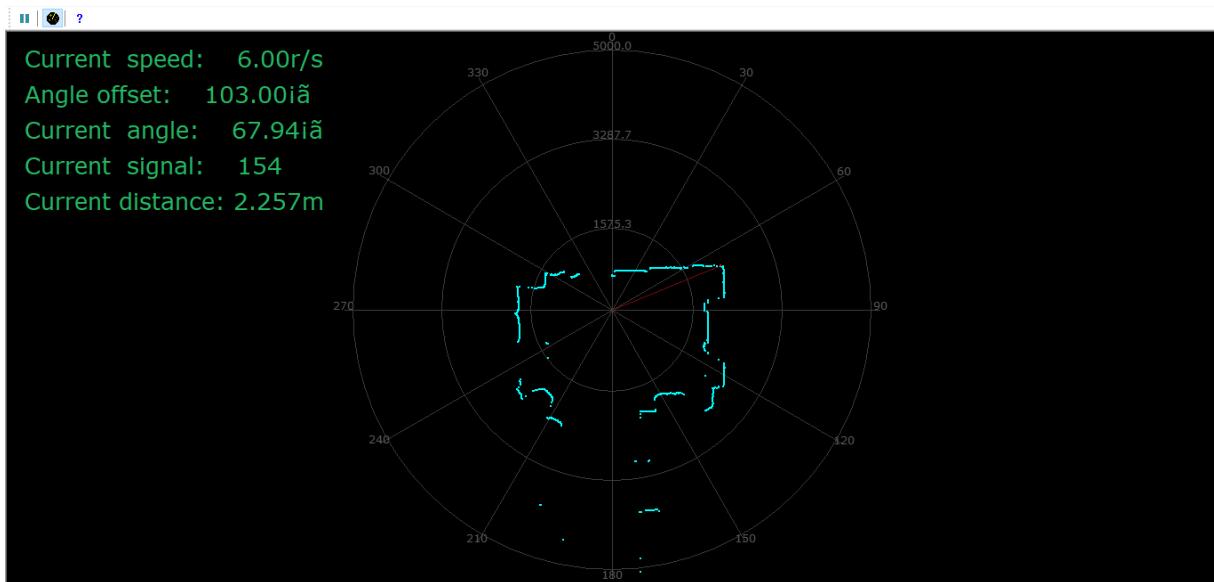


**Figura 7.9.7.- Vista isométrica del Sensor LiDAR y ordenador en proceso de mapeo.**



**Figura 7.9.8.- Vista isométrica de sensor LiDAR y módulo de control.**

El proceso de escaneo a su inicio logra tomar una lectura de 2.57m, que es lo ancho del entorno. Con esta lectura se puede ver el nivel de precisión del sensor sin el vehículo en movimiento (Figura 7.9.8).



**Figura 7.9.9.- Primera lectura del sensor sin el vehículo en movimiento.**

Para hacer el mapeo en 2D del entorno, se tiene que correr un nodo en ROS desde Ubuntu 18.04. Para evitar poner el código en este trabajo, se creó un repositorio en GitHub donde se podrán consultar todos los archivos y códigos para poder hacer este proyecto: <https://github.com/brasilmatt/agvTESI>

Para iniciar el nodo y empezar a hacer un mapeo del entorno con el vehículo en movimiento, tendrá que dirigirse a la consola y correr el nodo que se encuentra en la carpeta src/node.cpp y el client.cpp para que el sensor empiece a funcionar con ROS. Los resultados obtenidos son lo que se aprecian en la Figura 7.9.10 con una lectura de 2.60x7.50m con esta lectura y los datos que arroja el software, se puede crear un mapa 3D para la localización del vehículo.

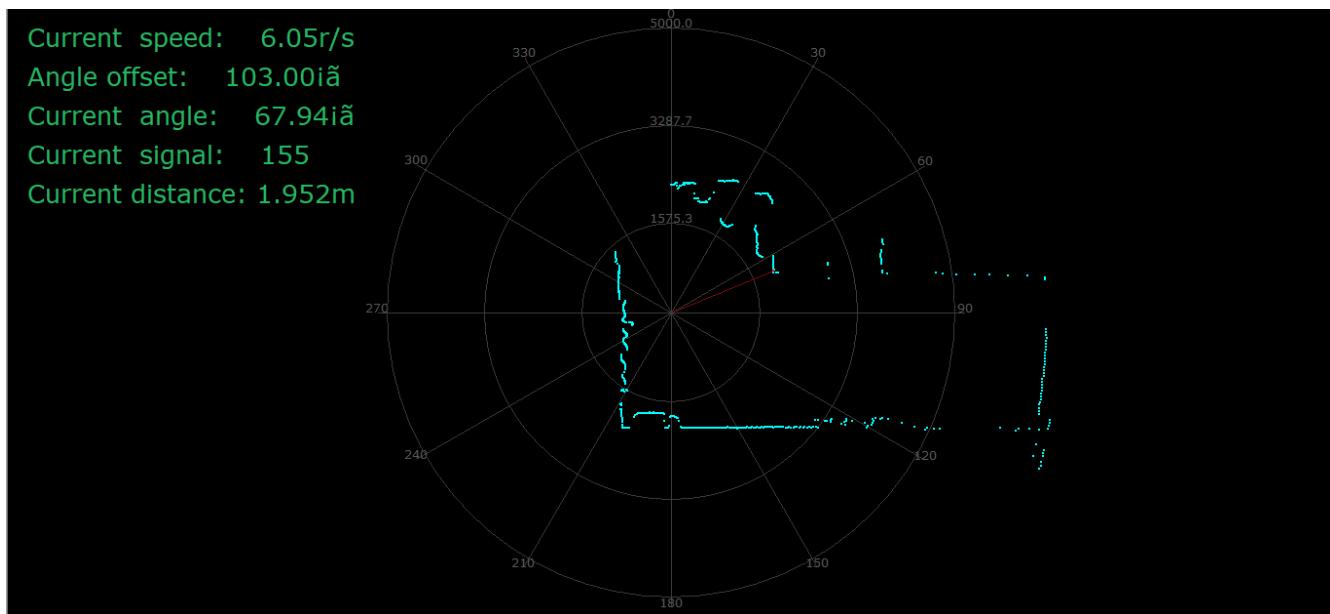


Figura 7.9.10.- Lectura para generación de mapa 3D.

### 7.9.3 SLAM

Se utilizará Hector SLAM que por sus siglas en inglés significa: *simultaneous localization and mapping*, primero se tiene que instalar hector-slam con el siguiente comando:

```
$ sudo apt-get install ros-kinetic-hector-slam
```

A continuación ejecutar el comando para reproducirlo en la maquina

```
$ roslaunch localization hector_slam.launch
```

Se abrirá una ventana con un mapa y na vez que este se haya generado por completo, en una nueva ventana de terminal, se tendrá que ejecutar lo siguiente para guardar el mapa como un archivo yaml. La última cadena después de "-f" es el nombre del mapa que desea guardar. Dado que en este caso estamos usando el archivo de bolsa del mapa precargado, llamamos apropiadamente al mapa "moore".

```
$ roslaunch map_server map_server -f moore
```

Con estos datos ya se puede generar el mapa 3D y probar los algoritmos antes de llevar el vehículo al mundo real, estos se encuentran en el repositorio de este proyecto. Con estos algoritmos se pueden generar y diseñar puntos de llegada, salida o estaciones de punto de carga y descarga. Para ello se tiene que generar un mundo como normalmente se conoce en el ámbito de generación de mapas 3D. Se utilizará el software SketchUp de la empresa Google y este lo podrá trabajar en el sistema operativo de su preferencia, también puede utilizar otro software de modelado 3D más avanzados como: Solid Works, AutoCAD, Fusion 360, etc.

## 7 Presupuesto

Para los componentes y herramientas propuestos durante el capítulo 7: DESARROLLO E IMPLEMENTACIÓN, se hizo un presupuesto en Amazon y se pueden consultar en el siguiente enlace:

[https://www.amazon.com.mx/gp/registry/wishlist/3ANDAMLK832PL/ref=cm\\_wl\\_huc\\_view](https://www.amazon.com.mx/gp/registry/wishlist/3ANDAMLK832PL/ref=cm_wl_huc_view)

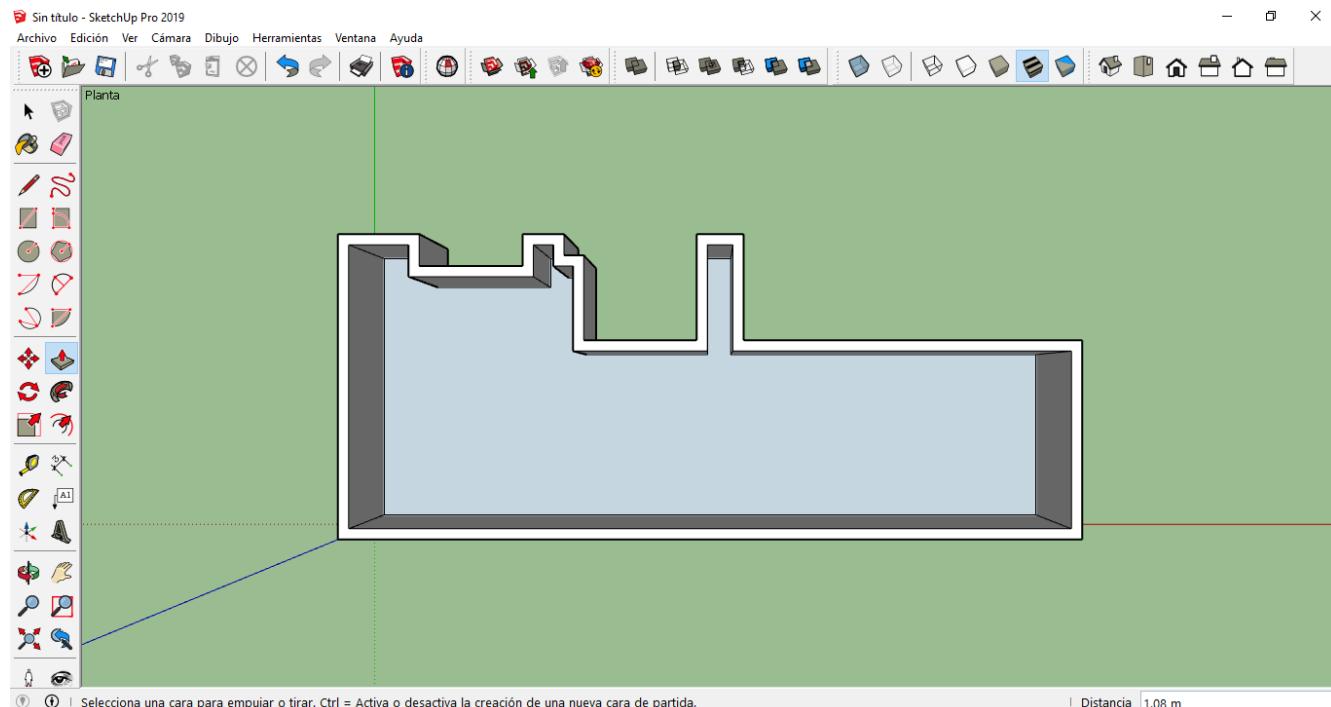
En la tabla 7.7 se resume la lista de la wishlist, omitiendo el costo del envío ya que puede variar en algunos productos y con una unidad por componente, por lo tanto el total es un estimado del precio final en pesos Mexicanos.

**TABLA 7.7 Presupuesto de Amazon sin envío.**

COMPONENTE	PRECIO
JETSON NANO DEVELOPER KIT 4GB	8,318.93
Micro SD Card	173
Batería USB 5V 2A	487
Cable USB a 2.1mm	150.52
Adaptador AC-DC 5V 4A	269.88
Chassis Tamiya Honda NSX Kit - TT02.	4,331.54
Servo futaba s3004	920.74
Radio Tx-Rx 3ch	1,084.79
Batería NiMH 7.2V 3000mAh	507.31
Cargador NiMH	774.02
DRIVER SERVO	209.61
Multiplexor servo	200
WiFi Trajeta	473.05
Antenas WiFi	184.10
Cámara Raspberry Pi	1,057
<b>SUB TOTAL</b>	<b>19,141.49</b>

## 8 RESULTADOS

Para cumplir con los objetivos del trabajo la simulación de pruebas del vehículo de guiado autónomo, se puede elegir cualquiera de los simuladores vistos en el apartado **2.3 VEHÍCULOS AUTONÓMOS Y SIMULADORES DE CODIGO ABIERTO**. Para esto ya se debe de contar con un mapa 3D generado en un software de modelado 3D como se muestra en la Figura 8.1.1, que previamente se realizó con la ayuda del sensor LiDAR Delta-2A mostrado en el apartado **7.6.2 LiDAR**.



**Figura 8.1.1.- Mapa 3D en SketchUp con los datos obtenidos del sensor LiDAR.**

La navegación en el entorno que se escaneó fue grabada y simulada en gazebo con velocidades moderadas tanto en el mapa 3D y 2D con ajustes lumínicos controlados y con leyes físicas configuradas de acuerdo al recinto y controlándolo con el teclado del ordenador. La Figura 8.1.2 muestra el trayecto en color azul con un control moderado de acuerdo a los parámetros que se le asignaron en el nodo y la Figura 8.1.3 muestra en los puntos rojos en donde se encuentra el vehículo

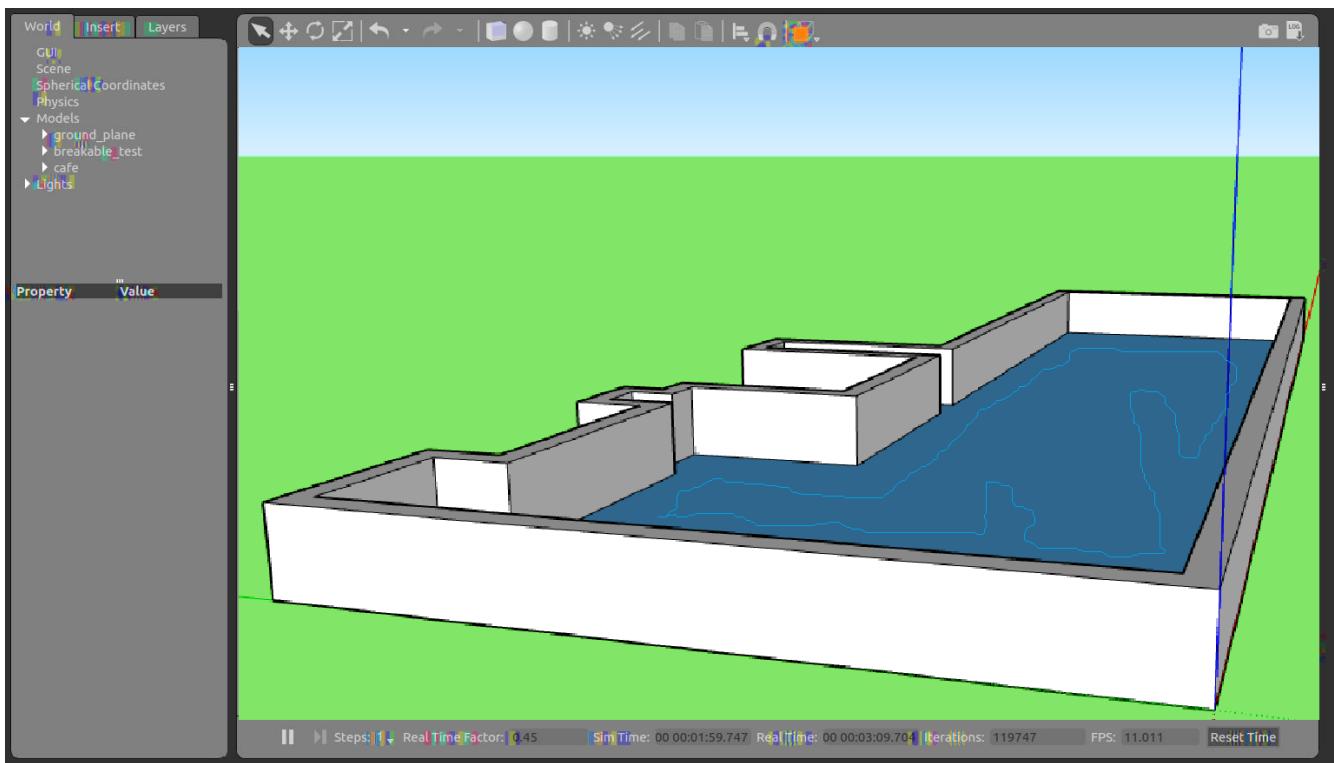


Figura 8.1.2.- Ruta de escaneo con sensor LiDAR en Gazebo.

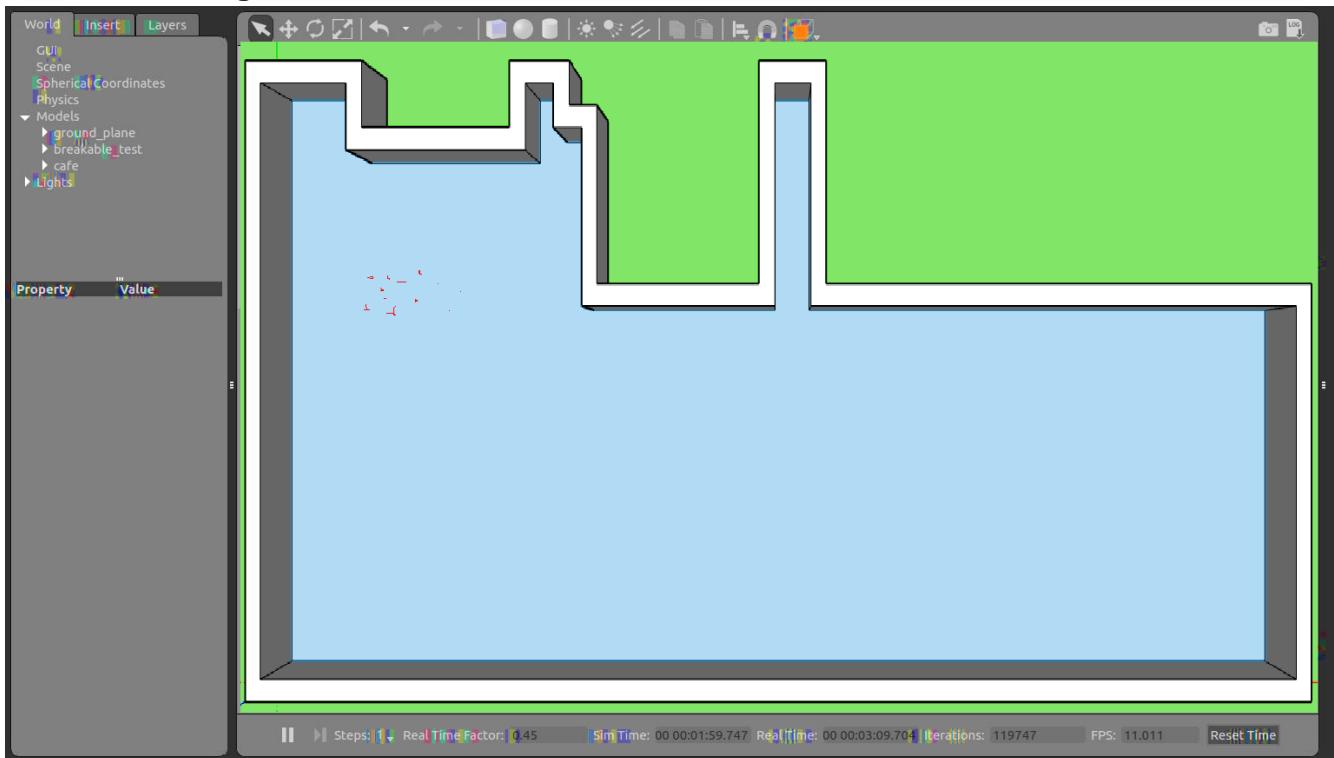


Figura 8.1.3.- Localización de vehículo en mapa 3D.

Para finalizar, los datos que se están ejecutando durante el escaneo y la comunicación que está haciendo ROS se muestra en el gráfico de la Figura 8.1.4 estas graficas se pueden obtener escribiendo en la consola el siguiente comando durante la ejecución del nodo:

```
$ rosrun rqt_graph rqt_graph
```

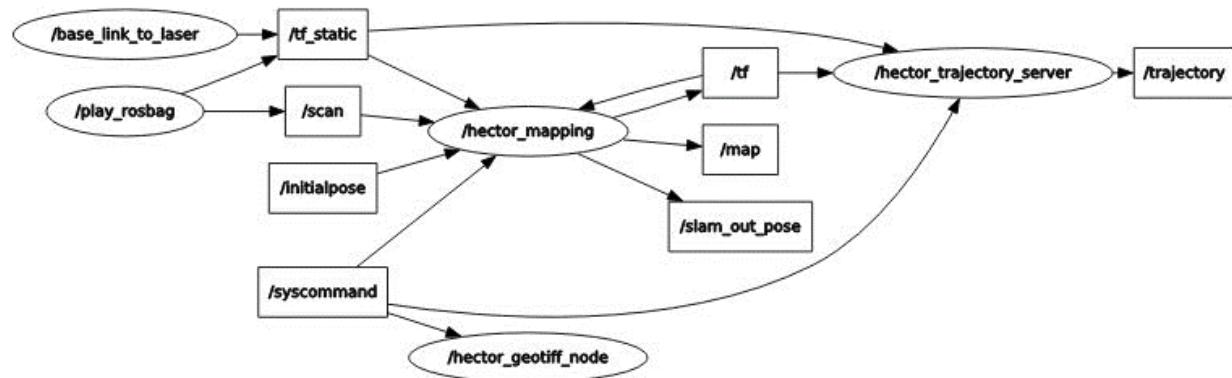


Figura 8.1.4.- Rqt\_graph generado de Hector SLAM.

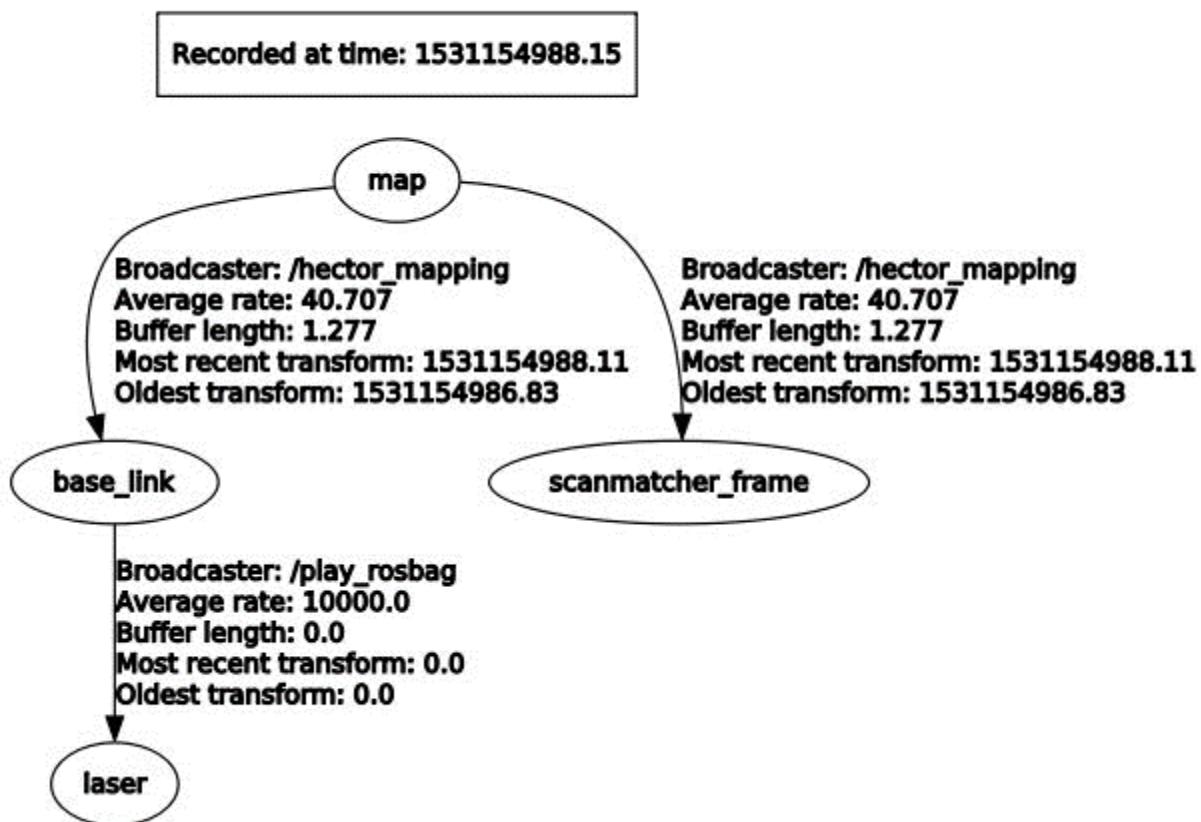


Figura 8.1.5.- Rqt\_tf\_tree generado para Hector SLAM ejecutando rosrun rqt\_tf\_tree rqt\_tf\_tree.

## **9 CONLUSIONES**

En este trabajo de investigación se cumplió el objetivo general de simular un vehículo capaz de brindar y obtener datos para generar un ambiente de coordinación en una línea de producción verificando mediante simulación la colaboración entre programas que nos son usados muy a menudo en la industria y que a parte son de código abierto.

También uno de los objetivos fue implementar un prototipo similar al que se va a utilizar en un trabajo futuro y se cumplió gracias a que se contaba con un chasis de un vehículo a escala con todos sus componentes, un sensor con el que ya se contaba y por los resultados obtenidos se seguirá utilizando y es por eso que no se incluyó dentro del presupuesto, por otra parte un componente fundamental era la Integración por Computadora de Placa Única, pero dado que esta implementación es necesaria solo para la integración de todos los componentes una vez que ya se tengan, se utilizó una computadora personal portátil.

Por otro lado, a menudo que se fue avanzando en la investigación se observó que actualmente los vehículos autónomos usan a menudo componentes muy similares, no importando el tamaño de este. Para este caso que es un vehículo a escala para usarse en la industria de manufactura flexible que tiene un diseño diferente a un vehículo comercial, su puede adaptar cada uno de los componentes, tanto de comunicación como cada uno de los sensores. También cabe resaltar que los materiales utilizados son de fácil acceso y con documentación necesaria para su implementación.

Con el tema de los sensores, se puede concluir que los más usados y seguros de implementar en sistemas autónomos son los sensores LiDAR, ya que hay algunos vehículos que incorporan sensores ultrasónicos, pero no son muy confiables a la hora de llevar a cabo un proceso de autonomía que requiera un nivel de seguridad alto.

Para finalizar, el tema de la industria 4.0 y dentro de las tecnologías que esta conlleva se logró utilizar la simulación como prueba de algoritmos y la inteligencia artificial se puede implementar en el vehículo para algoritmos del sistema autónomo. Por otro lado el internet de las cosas puede permitir en un sistema de manufactura flexible compartir datos de sus mismos sensores para un monitoreo y datos que el vehículo de guiado autónomo va recolectando, por ejemplo: se puede utilizar la Identificación por Radiofrecuencia y darle seguimiento a la materia prima de la empresa, otra aplicación de esta tecnología puede ayudar mucho al vehículo a saber dónde dar vuelta con etiquetas que se adhieran al piso previamente configuradas y al momento de ser leídas por el vehículo este sepa si tiene que dar vuelta, dirigirse a un centro de carga, etc.

## **10 TRABAJO FUTURO**

Dados los resultados obtenidos aún se pueden agregar varios complementos para el desarrollo del vehículo, entre ellas se encuentran:

- Ensamblar un vehículo con todas las características para que logre desplazarse de una manera más segura.
- Implementar nuevos algoritmos.
- Desarrollar un sistema para carga de batería del vehículo.
- Implementación del vehículo en un entorno simulado con las características de un Sistema de Manufactura Flexible.
- Implementar más tecnologías de la industria 4.0.

## 11 REFERENCIAS

- [1] M. Yakut Ali, S. G. M. Hossain, H. Jamil, and M. Z. Haq, "Development of automated guided vehicles for industrial logistics applications in developing countries using appropriate technology," *Int. J. Mech. Mech. Eng.*, vol. 10, no. 2, pp. 13–17, 2010.
- [2] P. Samaranayake, K. Ramanathan, and T. Laosirihongthong, "Implementing industry 4.0 - A technological readiness perspective," *IEEE Int. Conf. Ind. Eng. Eng. Manag.*, vol. 2017-Decem, pp. 529–533, 2018, doi: 10.1109/IEEM.2017.8289947.
- [3] R. Hill, J. Devitt, A. Anjum, and M. Ali, "Towards in-transit analytics for industry 4.0," *Proc. - 2017 IEEE Int. Conf. Internet Things, IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017*, vol. 2018-Janua, pp. 810–817, 2018, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.124.
- [4] G. Meneses Benavides, C. Velasquez Gómez, and R. Grisales Arboleda, "Prototipo De Vehículo De Guiado Autónomo Para Sistema De Almacenamiento Y Recuperación Automático De Libros," *Rev. Investig. Univ. del Quindío*, vol. 25, no. 1, pp. 73–79, 2014, doi: 10.33975/riuq.vol25n1.156.
- [5] I. F. A. Vis, "Survey of research in the design and control of automated guided vehicle systems," *Eur. J. Oper. Res.*, vol. 170, no. 3, pp. 677–709, 2006, doi: 10.1016/j.ejor.2004.09.020.
- [6] Hellmund A-M, Wirges S, Tas ÖS, Bandera C, Salscheider NO. Robot Operating System: A Modular Software Framework for Automated Driving. In: Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC). ; 2016:1564-1570. doi:10.1109/ITSC.2016.7795766
- [7] Garber L. Robot OS : A New Day for Robot Design. *IEEE Journals Mag.* 2013;46(12):16-20. doi:10.1109/MC.2013.434
- [8] Beinschob P, Meyer M, Reinke C, Digani V, Secchi C, Sabattini L. Semi-automated map creation for fast deployment of AGV fleets in modern logistics. *Rob Auton Syst.* 2017;87:281-295. doi:10.1016/j.robot.2016.10.018
- [9] Acosta Calderon CA, Mohan ER, Ng BS. Development of a hospital mobile platform for logistics tasks. *Digit Commun Networks.* 2015;1(2):102-111.
- [10] Shakya R, Rajanwal K, Patel S, Maurya RK. Optimization and Designing of Pid, Fuzzy & Pid-Fuzzy Controller. *Int J Sci Eng Res.* 2014;5(1):2040-2048.
- [11] Quigley M, Conley K, Gerkey B, et al. ROS: an open-source Robot Operating System. *Icra.* 2009;3(Figure 1):5. [http://pub1.willowgarage.com/~konolige/cs2\\_25B/docs/quigley-icra2009-ros.pdf](http://pub1.willowgarage.com/~konolige/cs2_25B/docs/quigley-icra2009-ros.pdf).
- [12] ESIME Azcapotzalco, "Mapa y Objetivos de Calidad," Disponible en: <https://www.esimeazc.ipn.mx/conocenos/sistema-de-gesti%C3%B3n-de-la-calidad.html>.

- [13] “ESIME - Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Azcapotzalco IPN · Av. de las Granjas 682, Santa Catarina, Azcapotzalco, 02550 Ciudad de México, CDMX, Mexico,” Disponible en: <https://www.google.com/maps/search/19.4896843,99.1742856?sa=X&ved=2ahUKEwjMh5evmpL0AhXZk2oFHQoXDu4Q8gF6BAgCEAE>
- [14] ESIME Azcapotzalco, “Organigrama,” Disponible en: <https://www.esimeazc.ipn.mx/conocenos/organigrama.pdf>.
- [15] P. Document, “J3016C: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International.” 30-Apr-2021.
- [16] WARRENDALE and Pa., “SAE International Releases Updated Visual Chart for Its ‘Levels of Driving Automation’ Standard for Self-Driving Vehicles.” 11-Dec-2018.
- [17] “¿Qué es el Frenado de Emergencia Autónoma? | Kia Motors Mexico,” *Kia México*.
- [18] A. L. Olana, *Seguridad Vial en vehículos altamente automatizados*. Centro Zaragoza, 2018.
- [19] GM Authority, “This Is Cruise Origin, GM’s First Driverless Car.” 21-Jan-2020.
- [20] Ford, “Quantum Signal: Compañía experta en Vehículos de Tecnología Autónoma,” [en línea] disponible en: <https://www.ford.mx/blog/innovacion/ford-vehiculos-autonomos-quantum-signal-201910/>.
- [21] Tesla Inc, “Autopilot,” [en línea] Disponible en: [https://www.tesla.com/es\\_MX/autopilot](https://www.tesla.com/es_MX/autopilot).
- [22] “Transforming transportation, without shortcuts,” *Cruise* [en línea] Disponible en: <https://www.getcruise.com/technology/>
- [23] B. Berman, “CICV, Marmara University and BigTRI Join the International Alliance for Mobility Testing & Standardization - Cruise Origin autonomous vehicle unveil.” 22-Jan-2020.
- [24] Ford (02-Dic-21), “Ford Anuncia La Adquisición De Quantum Signal Para Desarrollar Vehículos De Conducción Autónoma,” *FORD MEDIA CENTER* [en línea] Disponible en: <https://media.ford.com/content/fordmedia/fna/mx/es/news/2019/07/30/ford-anuncia-la-adquisicion-de-quantum-signal-para-desarrollar-v.html#>
- [25] TESLA, (07-Dic-21) “Support” [en línea] Disponible en: <https://www.tesla.com/support/autopilot>.
- [26] Waymo (09- Dic -21), “Most Experienced Driver” [en línea] [waymo.com/intl/es/](http://waymo.com/intl/es/)
- [27] Waymo, (09- Dic -21) “Waymo One” [en línea] [waymo.com/intl/es/waymo-one/](http://waymo.com/intl/es/waymo-one/)
- [28] Waymo (09- Dic -21) “Waymo Via”, [en línea] [waymo.com/intl/es/waymo-via/](http://waymo.com/intl/es/waymo-via/)
- [29] P. Sun o.fl., „Scalability in perception for autonomous driving: Waymo open dataset“, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., bls. 2443–2451, 2020, doi: 10.1109/CVPR42600.2020.00252.
- [30] “LQ, vínculo con el conductor, Toyota España,” Toyota ES.
- [31] Toyota (10-12-21) “Una nueva movilidad al servicio de todos” [en línea] Disponible en: <https://www.toyota.es/startyourimpossible/e-palette>
- [32] Toyota Research Institute (12-Dic-21) “Research”, 07-May-2021 [en línea] Disponible en: [tri.global/research/?page=2](http://tri.global/research/?page=2)

- [33] P. Tokmakov, J. Li, W. Burgard, og A. Gaidon, „Learning to Track with Object Permanence“, 2021, [Rafrænt]. Aðgengilegt á: <http://arxiv.org/abs/2103.14258>.
- [34] NVIDIA (18-Ene-22), “Plataforma de Hardware NVIDIA DRIVE,” *NVIDIA* [en línea] Disponible en: [nvidia.com/es-la/self-driving-cars/drive-platform/hardware/](https://nvidia.com/es-la/self-driving-cars/drive-platform/hardware/)
- [35] Samsung (18-Ene-22) , „Samsung amplía su compromiso con la tecnología de Conducción Autónoma ,” [en línea] Disponible en: [news.samsung.com/mx/samsung-amplia-su-compromiso-con-la-tecnologia-de-conduccion-autonomia](https://news.samsung.com/mx/samsung-amplia-su-compromiso-con-la-tecnologia-de-conduccion-autonomia)
- [36] Intel (18-Ene-22), “Autonomous Driving at Intel,” *Intel Newsroom 12-Apr-2021.* [en línea] Disponible en: [newsroom.intel.com/press-kits/autonomous-driving-intel/?-1986101841.1486645630#gs.ma5fth](https://newsroom.intel.com/press-kits/autonomous-driving-intel/?-1986101841.1486645630#gs.ma5fth)
- [37] M. O’Kelly o.fl. F1/10: An Open-Source Autonomous Cyber-Physical Platform“, b. 10, 2019, [Rafrænt]. Aðgengilegt á: <http://arxiv.org/abs/1901.08567>.
- [38] M. Behl, “F1/10 UVA Home,” (22-Dic-21), [en línea] Disponible en: <https://linklab-uva.github.io/autonomousracing/index.html#header2-1>
- [39] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, og V. Koltun, „CARLA: An Open Urban Driving Simulator“, tbl. CoRL, bls. 1–16, 2017, [Rafrænt]. Aðgengilegt á: <http://arxiv.org/abs/1711.03938>
- [40] A. Dosovitskiy, G. Ros , F. Codevilla, A. Lopez, and V. Koltun (22-Dic-21), “CARLA: An Open Urban Driving Simulator,” [en línea] Disponible en: <https://carla.org/>
- [41] LG Electronics ,(22-Dic-21), “SVL SIMULATOR: An end-to-end autonomous vehicle simulation platform,” [en línea] Disponible en: <https://www.svlsimulator.com/>
- [42] H. Performance og O. S. Architecture, „Sensor-rich autonomous vehicle for self-driving applications“.
- [43] Quanser, (22-Dic-21), “Quanser Innovate Educate,” *Quanser 12-Oct-2021* [en línea] disponible en: <https://www.quanser.com/>
- [44] A. M. Hellmund, S. Wirges, Ö. Ş. Taş, C. Bandera, og N. O. Salscheider, „Robot operating system: A modular software framework for automated driving“, *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, bls. 1564–1570, 2016, doi: 10.1109/ITSC.2016.7795766.
- [45] SS SCHAEFER (12-Feb-22), “AGV - VEHÍCULO DE GUIADO AUTOMÁTICO EN LA INTRALOGÍSTICA ,” [en línea] Disponible en: [ssi-schaefer.com/es-es/productos/transportadores/agv-veh%C3%ADculo-de-guiado-autom%C3%A1tico](https://ssi-schaefer.com/es-es/productos/transportadores/agv-veh%C3%ADculo-de-guiado-autom%C3%A1tico)
- [46] D. I. (12-Feb-22), “Qué AGV elegir - Guías de compra DirectIndustry,” [en línea] Disponible en: [guide.directindustry.com/es/que-agv-elegir/](https://guide.directindustry.com/es/que-agv-elegir/) 09-Sep-2019.
- [47] C. E. Fúquene Retamoso, S. Aguirre Mayorga, og N. B. Córdoba Pinzón, „Evolución de un sistema de manufactura flexible (FMS) a un sistema de manufactura integrada por computador (CIM)Itad de Ingeniería, Pontificia Universidad Javeriana., Ing. y Univ., b. 11, tbl. 1, bls. 57–69, 2007, [Rafrænt]. Aðgengilegt á: <http://www.redalyc.org/articulo.oa?id=47711104>.
- [48] J. Hernandez, Gabriel; Medel Juaréz, „Tiempo real en los sistemas de manufactura flexible automatizado Automated Flexible Manufacturing System in Real-Time“, *Comput. y Sist. Vol.*, b. 15, tbl. 1, bls. 129–132, 2011.
- [49] Eric Vallejo R, „Sistemas flexibles de manufactura“, *Rev. Científica Ing. y Desarro.*, bls. 43–49, 1998, [Rafrænt]. Aðgengilegt á: <http://rcientificas.uninorte.edu.co/index.php/ingenieria/article/viewArticle/2191>
- [50] M. P. Groover, *Automation, Production Systems, and Computer-integrated Manufacturing*, 4th ed. Prentice Hall, 2008, pp. 533.

- [51] S. A. Foyo "Coordinación de Robots Moviles en Sistemas Automatizados de Manufactura Utilizando Redes de Petri" M. en C. IPN ESIME, México, D.F, 2015.
- [52] Muther, R., and K. Haganas, Systematic Handling Analysis, Management and Industrial Research Publications, Kansas City, MO, 1969.
- [53] D. T. Martinez " Metodología Para La Implementación De Las Tecnologías De La Industria 4.0 En Las Mipymes En México. ", UAEM, Morelos, 2020.
- [54] M. Saturno, V. Moura Pertel, F. Deschamps, and E. Loures, "Proposal of an automation solutions architecture for Industry 4.0," *Destech Trans. Eng. Technol. Res.*, Mar. 2017. doi: 10.12783/dtetr/icpr2017/17675
- [55] A. P. Cardenas " Propuesta De Implementación De La Industrial 4.0 En El Sector Manufacturero De Bogotá " Universidad Católica de Colombia, Bogotá D.C 2020.
- [56] S.E.,Crafting the future: a roadmap for industry 4.0 in Mexico: report“, 2016.
- [57] S. de Economía, “El desarrollo de la Industria 4.0 en México,” gob.mx. 08-Apr-2018.
- [58] C. Imes (23-Dic-21), “Repositories Ubuntu,” *Ubuntu Documentation*, [en línea] Disponible en: <https://help.ubuntu.com/community.Repositories/Ubuntu>
- [59] Raspberry Pi, “Raspberry Pi 4,” *Raspberry Pi Foundation* [en línea] disponible en <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>