



UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
CARRERA DE ELECTROMECAÁNICA

**PROPUESTA DE EVALUACIÓN AUTOMÁTICA DE
CONDICIONES ERGONÓMICAS DE UN TRABAJADOR
USANDO EL MÉTODO RULA.**

PROPUESTA TECNOLÓGICA PREVIO A LA OBTENCIÓN DEL TÍTULO
DE
INGENIERO ELECTROMECAÁNICO

AUTOR:

Marlon Santiago Udeo Chuchuca

TUTOR:

Ing. MSC. Ángel Guillermo Hidalgo Oñate

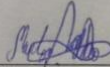
LATACUNGA, AGOSTO 2025

Latacunga, julio 2025

DECLARACIÓN DE AUTORÍA

Yo, Udeo Chuchuca Marlon Santiago declaro ser autor del proyecto de titulación **“PROPUESTA DE EVALUACIÓN AUTOMÁTICA DE CONDICIONES ERGONÓMICAS DE UN TRABAJADOR USANDO EL MÉTODO RULA”**, siendo el Ing. Ms.C. Ángel Guillermo Hidalgo Oñate tutor del presente trabajo de titulación; y eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo de titulación, son de mi exclusiva responsabilidad.

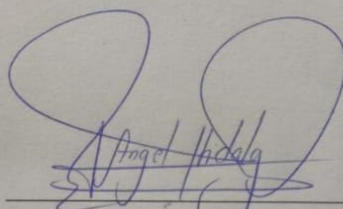


Udeo Chuchuca Marlon Santiago
CC. 0504374406

Latacunga, julio 2025

AVAL DEL TUTOR DEL PROYECTO DE INVESTIGACIÓN

En calidad de Tutor del Trabajo de Investigación sobre el título: **“PROPUESTA DE EVALUACIÓN AUTOMÁTICA DE CONDICIONES ERGONÓMICAS DE UN TRABAJADOR USANDO EL MÉTODO RULA”**, propuesto por el estudiante Udeo Chuchuca Marlon Santiago de la Carrera de Ingeniería Electromecánica, considero que dicho proyecto de titulación cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos al tribunal de lectores.



Ing. Ms.C. Ángel Guillermo Hidalgo Oñate
C.C. 0503257404
TUTOR

Latacunga, julio 2025

AVAL DE APROBACIÓN DE LECTORES

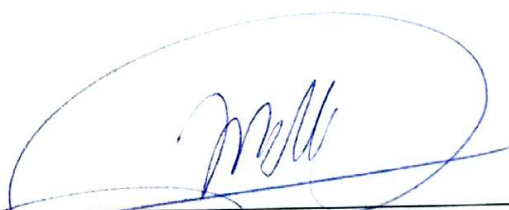
Cumpliendo con el Reglamento de Titulación de la Universidad Técnica de Cotopaxi, en calidad de Lectores de Tribunal de Propuesta Tecnológica con el Título “PROPUESTA DE EVALUACIÓN AUTOMÁTICA DE CONDICIONES ERGONÓMICAS DE UN TRABAJADOR USANDO EL MÉTODO RULA”, propuesto por el estudiante Udeo Chuchuca Marlon Santiago de la Carrera de Ingeniería Electromecánica, nos permitimos indicar que el estudiante ha concluido todas las observaciones y realizado las correcciones señaladas por el Tribunal de Lectores, por lo cual presentamos el Aval de Aprobación del Proyecto de Titulación correspondiente a la modalidad de Propuesta Tecnológica en virtud de lo cual el postulante puede presentarse a la Defensa de su Proyecto de Titulación.

Particular que ponemos en conocimiento para los fines legales pertinentes.

Atentamente,



LECTOR 1 (PRESIDENTE)
ING. Albarracín Álvarez Mauro Darío
CC: 050311373-0



LECTOR 2
ING. Moreano Martínez Edwin Homero
CC: 050260750-0



LECTOR 3
ING. Luigi Orlando Freire Martínez
CC: 050252958-9

AGRADECIMIENTO

Agradezco profundamente a mi padre Udeo Vinicio, y a Chuchuca Carmen, mi madre, por haber sido mi pilar fundamental durante todo este proceso. Su apoyo incondicional, esfuerzo y confianza han sido el motor que me ha impulsado a seguir adelante incluso en los momentos más difíciles.

Extiendo un especial agradecimiento al Ing. Hidalgo Ángel, tutor de este trabajo de titulación, por su guía, compromiso y valiosas observaciones que permitieron el correcto desarrollo de este proyecto.

A la Universidad Técnica de Cotopaxi, mi alma máter, por brindarme una formación integral, tanto académica como humanística, que me ha permitido crecer como profesional y como ser humano.

Finalmente, a mis amigos y compañeros de camino, quienes me ofrecieron su apoyo constante, palabras de ánimo y colaboración durante las distintas etapas de mi carrera y en la ejecución de este trabajo de titulación.

Marlon Udeo

DEDICATORIA

Dedico este trabajo, en primer lugar, a mis padres, Udeo Vinicio y Chuchuca Carmen, por su amor, sacrificio y constante apoyo. Su ejemplo de esfuerzo y perseverancia ha sido la inspiración que me ha guiado a lo largo de mi formación.

A mis amigos y seres queridos, por estar presentes en cada paso del camino, brindándome palabras de aliento y acompañándome con paciencia y comprensión.

Y a mí mismo, por no rendirme, por creer en este proyecto y por haber recorrido este camino con determinación y compromiso.

Marlon Udeo

UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TÍTULO: “PROPUESTA DE EVALUACIÓN AUTOMÁTICA DE CONDICIONES ERGONÓMICAS DE UN TRABAJADOR USANDO EL MÉTODO RULA”

Autor:

Udeo Chuchuca Marlon Santiago

RESUMEN

El presente proyecto tiene como finalidad el desarrollo de un sistema automático para la evaluación de condiciones ergonómicas mediante la aplicación del método RULA (Rapid Upper Limb Assessment). La propuesta responde a la necesidad de optimizar los procesos de análisis postural, reduciendo la subjetividad y el tiempo requerido en las evaluaciones manuales tradicionales.

Para ello, se diseñó un prototipo basado en sensores inerciales distribuidos estratégicamente en el cuerpo humano, controlados por una unidad central que gestiona la lectura de datos y su transmisión en tiempo real. Esta información es procesada por un entorno gráfico en Unity, donde se visualiza un modelo tridimensional del usuario. A través de la interpretación de cuaterniones y su conversión a ángulos de Euler, se obtienen los ángulos relativos entre extremidades, fundamentales para aplicar los criterios del método RULA de forma automatizada.

El sistema fue sometido a pruebas en un entorno controlado para validar su funcionamiento, realizando una comparación cualitativa con valores de referencia conocidos. Se identificaron factores determinantes en la precisión del sistema, como la correcta colocación y calibración de los sensores. Finalmente, se efectuó un análisis económico y funcional para evaluar la viabilidad de implementación en contextos reales.

Palabras clave: evaluación ergonómica, automatización, método RULA, sensores inerciales, Unity.

TECHNICAL UNIVERSITY OF COTOPAXI
FACULTY OF ENGINEERING AND APPLIED SCIENCES

TITLE: " Proposal for the Automatic Evaluation of a Worker's Ergonomic Conditions Using
the RULA Method"

Author:

Udeo Chuchuca Marlon Santiago

ABSTRACT

This project aims to develop an automated system for the ergonomic assessment of workers through the application of the RULA (Rapid Upper Limb Assessment) method. The proposal addresses the need to optimize posture analysis processes by reducing subjectivity and the time required in traditional manual evaluations.

To achieve this, a prototype was designed based on inertial sensors strategically placed on the human body, managed by a central unit that handles data acquisition and real-time transmission. This information is processed within a graphical environment in Unity, where a 3D model of the user is visualized. By interpreting quaternions and converting them into Euler angles, the system calculates relative angles between body segments, which are essential for automatically applying the RULA criteria.

The system underwent testing in a controlled environment to validate its performance, using qualitative comparisons with known reference values. Key factors affecting accuracy were identified, particularly the proper placement and calibration of sensors. Finally, an economic and functional analysis was conducted to evaluate the feasibility of its implementation in real-world contexts.

Keywords: ergonomic assessment, automation, RULA method, inertial sensors, Unity.

ÍNDICE GENERAL

1	INFORMACION GENERAL	1
2	INTRODUCCIÓN.....	2
2.1	SITUACIÓN PROBLEMÁTICA.....	2
2.2	FORMULACIÓN DEL PROBLEMA	3
2.3	OBJETO Y CAMPO DE ACCIÓN	3
2.3.1	Objeto de investigación	3
2.3.2	Campo de acción	3
2.4	BENEFICIARIOS	3
2.5	JUSTIFICACIÓN.....	3
2.6	OBJETIVOS.....	4
2.6.1	Objetivo General.....	4
2.6.2	Objetivos Específicos	4
2.7	SISTEMA DE TAREAS.....	4
3	FUNDAMENTACIÓN TEÓRICA	6
3.1	AUTOMATIZACIÓN DE LA EVALUACIÓN ERGONÓMICA	6
3.1.1	Introducción a sistemas de monitoreo en tiempo real	6
3.1.2	Relevancia de la automatización en ergonomía.	6
3.1.3	Antecedentes o trabajos similares a nivel internacional.....	7
3.1.4	Sistemas de referencia biomecánica.	8
3.1.5	Tipos de representación de orientación.	9
3.2	IMU (UNIDAD DE MEDICIÓN INERCIAL).	10
3.2.1	Principio de funcionamiento:	10
3.2.2	MPU9250.	10
3.3	ALGORITMOS DE ORIENTACIÓN Y FILTRADO.....	11
3.3.1	Algoritmo Madgwick	11
3.3.2	Algoritmo Mahony	11
3.3.3	Filtro mediante DMP (hardware).....	12
3.3.4	Filtro complementario.	12
3.4	ESP32	12
3.4.1	Arquitectura del ESP32	12
3.5	INTERNET DE LAS COSAS (IoT)	13
3.6	REPRESENTACIÓN 3D Y UNITY	14

3.6.1	Unity.....	14
3.6.2	Modelos humanoides.....	14
3.6.3	Aplicación de rotaciones en modelos 3D.....	14
3.7	ERGONOMÍA Y RIESGOS MÚSCULO-ESQUELÉTICOS.....	15
3.7.1	Definición de ergonomía.....	15
3.7.2	Importancia de la ergonomía en la prevención de TME (Trastornos Músculo-Esqueléticos).....	15
3.7.3	Método RULA.....	16
3.7.4	Legislación aplicable en Ecuador.....	17
4	MÉTODOS Y PROCEDIMIENTOS.....	19
4.1	METODOLOGÍA DE INVESTIGACIÓN.....	19
4.1.1	Metodología Waterfall (Cascada).....	19
4.2	ETAPA DE ANÁLISIS.....	20
4.3	ETAPA DE DISEÑO.....	22
4.3.1	Arquitectura del sistema.....	22
4.3.2	Fundamentos matemáticos y representación de orientación.....	23
4.3.3	Selección de algoritmos de orientación.....	27
4.3.4	Justificación de selección de componentes.....	29
4.3.5	Distribución y lógica de sensores.....	30
4.3.6	Estructura trabajo con microcontrolador.....	31
4.3.7	Diseño del sistema en Unity.....	33
4.4	ETAPA DE IMPLEMENTACIÓN.....	36
4.4.1	Diagramas de flujo para la ESP32.....	39
4.4.2	Diagramas de flujo para Unity.....	41
4.5	ETAPA DE PRUEBAS.....	43
4.6	ETAPA DE AJUSTES.....	45
5	ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS.....	48
5.1	EVALUACIÓN DEL FUNCIONAMIENTO DEL PROTOTIPO.....	48
5.1.1	Lectura de sensores y estabilidad.....	48
5.2	DISCUSIÓN GENERAL DE RESULTADOS.....	56
5.3	ANÁLISIS ECONÓMICO.....	58
6	CONCLUSIONES Y RECOMENDACIONES.....	60
6.1	CONCLUSIONES.....	60

6.2	RECOMENDACIONES	61
7	REFERENCIAS	62

ÍNDICE DE FIGURAS

Figura 4.1 Componentes principales del sistema	23
Figura 4.2 Modelo de cuaternión.....	25
Figura 4.3 Proceso de lectura de los sensores	33
Figura 4.4 Esquema básico de conexión.	37
Figura 4.5 Esquema PCB	37
Figura 4.6 Montaje de sensor	38
Figura 4.7 Inicio de sensores, detección de errores y lectura de datos.....	38
Figura 4.8 Diagrama de flujo de macroprocesos de la ESP32	39
Figura 4.9 Diagrama de flujo del proceso de lectura y envío de datos de sensores.	40
Figura 4.10 Diagrama de flujo del funcionamiento del servidor WebSocket.....	41
Figura 4.11 Diagrama de flujo de conexión y recepción de datos por WebSocket.	41
Figura 4.12 Diagrama de flujo de actualización del modelo 3D y cálculo de puntaje RULA.	42
Figura 4.13 Dispositivo montado.	47
Figura 5.1 Ejes de rotación aplicados en el modelo.	50
Figura 5.2 Postura firme	50
Figura 5.3 Postura firme en Unity.	51
Figura 5.4 Posición en T.....	52
Figura 5.5 Posición en T en Unity.	52
Figura 5.6 Brazo levantado.....	53
Figura 5.7 Brazo levantado en Unity.	54
Figura 5.8 Movimiento de cuello.	55
Figura 5.9 Movimiento de cuello en Unity.....	55

ÍNDICE DE TABLAS

Tabla 2.1 Beneficiarios directos e indirectos.....	3
Tabla 2.2 Lista de tareas a realizar por objetivo.....	5
Tabla 3.1 Estudios relacionados.....	8
Tabla 3.2 Formas de representar orientación.....	9
Tabla 3.3 Elementos para el sistema IoT.....	13
Tabla 3.4 Puntuación del brazo.....	16
Tabla 3.5 Combinación de puntuación de extremidades superiores.....	17
Tabla 4.1 Comparación de características entre filtros.....	29
Tabla 4.2 Distribución de sensores en el cuerpo con ID.....	31
Tabla 5.1 Comparativa de filtros.....	48
Tabla 5.2 Comparación de resultados en postura firme.....	51
Tabla 5.3 Comparación de resultados en posición en T.....	53
Tabla 5.4 Comparación de resultados con brazo levantado.....	54
Tabla 5.5 Comparación de resultados para el movimiento del cuello.....	56
Tabla 5.6 Error en las pruebas.....	56
Tabla 5.7 Costo componentes para fabricación del prototipo.....	58
Tabla 5.8 Tiempo necesario para aplicar la evaluación manual y automática.....	59

1 INFORMACION GENERAL

Tema del proyecto: Propuesta de evaluación automática de condiciones ergonómicas de un trabajador usando el método RULA.

Modalidad de Titulación:

Propuesta Tecnológica	<input checked="" type="checkbox"/>
Proyectos de Investigación	<input type="checkbox"/>

Carrera: Electromecánica

Trabajo de Titulación Vinculado al Proyecto: "Orientación formativa práctica-colaborativo en carreras de Ciencias, Tecnología, Ingeniería y Matemática (STEM) en la provincia de Cotopaxi"

Equipo de Trabajo del Trabajo de Titulación:

Udeo Chuchuca Marlon Santiago.

Área de Conocimiento: 07:

Ingeniería, Industria y Construcción /071: Ingeniería y Profesiones Afines; 0714: Electrónica y automatización.

Línea de investigación:

Desarrollo de las ciencias de la ingeniería, energías alternativas, renovables, micro eléctrica y TICS/ Gestión de la calidad y seguridad laboral.

Sublíneas de investigación de la Carrera:

Automatización y control avanzado de procesos industriales e inteligencia artificial aplicado a la robótica y la industria.

2 INTRODUCCIÓN

La automatización de procesos permite procesos más eficientes, con menor costo e intervención humana reducida. En el campo de la ergonomía laboral existen métodos de evaluación, sin embargo, en distintas áreas se continúan realizando de forma manual, limitando su aplicación.

Este proyecto propone el desarrollo de un sistema automatizado capaz de evaluar condiciones ergonómicas mediante datos de movimiento corporal. El sistema utiliza sensores inerciales distribuidos en puntos clave del cuerpo humano para capturar su orientación en tiempo real. La información es procesada por un microcontrolador ESP32 y transmitida vía Wifi hacia una plataforma de visualización 3D desarrollada en Unity, donde se reconstruye la postura y se aplica el método RULA de forma automática.

A través de esta solución se busca integrar tecnologías disponibles en el mercado para automatizar la evaluación postural permitiendo un uso rápido y de menor costo.

2.1 SITUACIÓN PROBLEMÁTICA

La automatización de procesos ha permitido el avance en diferentes sectores, sin embargo, en el campo de la ergonomía ocupacional aún existe una dependencia de métodos de evaluación manuales que son lentos. Esto revela una oportunidad para el desarrollo de un sistema automatizado para la evaluación de condiciones ergonómicas mediante sensores y software especializado.

Durante un proceso de evaluación convencional se realizan diferentes procesos para la toma de datos, análisis de condiciones y evaluación, este proceso es lento y requiere de personal para llevarlo a cabo, esto debido a la ausencia de un proceso automatizado que permita agilizar este proceso y realizar evaluación de condiciones ergonómicas periódicamente y reducir así los riesgos ergonómicos presentes en diferentes áreas de trabajo.

2.2 FORMULACIÓN DEL PROBLEMA

¿Cómo el diseño y la implementación de un sistema electromecánico que integre sensores inerciales y procesamiento de datos puede automatizar la evaluación de posturas mediante el método RULA, superando las limitaciones de subjetividad y la ineficiencia temporal de los métodos de observación manual para la detección temprana de riesgos ergonómicos y la prevención de trastornos músculo-esqueléticos en trabajadores de la industria?

2.3 OBJETO Y CAMPO DE ACCIÓN

2.3.1 Objeto de investigación

Sistema automatizado para la evaluación de condiciones ergonómicas en trabajadores, basado en el método RULA.

2.3.2 Campo de acción

0714 Electrónica y automatización.

2.4 BENEFICIARIOS

A continuación, en la Tabla 2.1, se muestra los beneficiarios de este proyecto, tanto de forma directa como indirecta.

Tabla 2.1 Beneficiarios directos e indirectos.

Beneficiarios	Detalle
Directos	Los Trabajadores Personal de Seguridad y Salud Ocupacional (SSO)
Indirectos	Gerencia de la Empresa Familiares de los Trabajadores Sistema de Salud Pública - IESS

2.5 JUSTIFICACIÓN

El desarrollo de un sistema automatizado para la evaluación de condiciones ergonómicas en trabajadores responde a una necesidad crítica en los entornos laborales actuales. Personalmente, este proyecto representa la oportunidad de aplicar conocimientos adquiridos en ingeniería electromecánica para resolver problemas reales, integrando tecnología avanzada en beneficio de la salud ocupacional.

Académicamente, el proyecto tiene un enfoque teórico, metodológico y práctico. A nivel teórico, se fundamenta en el método RULA, ampliamente reconocido en la evaluación ergonómica, adaptándolo a una solución tecnológica innovadora. En el ámbito metodológico, busca optimizar procesos mediante el uso de sensores, dispositivos IoT y software, mientras que, en lo práctico, ofrece una herramienta útil y replicable para las empresas.

En resumen, esta iniciativa no solo atiende un problema prioritario de salud ocupacional, sino que también promueve la integración de tecnologías modernas para transformar y mejorar los procesos en el ámbito laboral, generando un impacto positivo en los trabajadores, las empresas y la sociedad en general.

2.6 OBJETIVOS

2.6.1 Objetivo General.

Desarrollar un sistema automático para la evaluación de condiciones ergonómicas de un trabajador usando el método RULA mediante sensores inerciales IMU y software para entornos 3D.

2.6.2 Objetivos Específicos

- Analizar información técnica sobre metodologías de evaluación ergonómica, sensores inerciales, protocolos de orientación corporal y herramientas de visualización en tiempo real con el fin de establecer una base conceptual y tecnológica para el diseño del sistema.
- Diseñar el modelo conceptual y funcional del sistema automatizado de evaluación de condiciones ergonómicas basado en el método RULA, estableciendo los requerimientos técnicos, tecnológicos y software necesario para su implementación.
- Evaluar el funcionamiento del sistema automatizado implementado aplicando el método RULA mediante pruebas en condiciones de uso reales.

2.7 SISTEMA DE TAREAS

A continuación, se observa la planificación de las actividades en la Tabla 2.2. Basándose en los objetivos planteados y que además se cuenta con los resultados deseados mediante el uso de técnicas, instrumentos y medios de investigación.

Tabla 2.2 Lista de tareas a realizar por objetivo.

Objetivos específicos	Actividades (tareas)	Resultados esperados	Técnicas, Medios e Instrumentos
Analizar información técnica sobre metodologías de evaluación ergonómica, sensores inerciales, protocolos de orientación corporal y herramientas de visualización en tiempo real con el fin de establecer una base conceptual y tecnológica para el diseño del sistema.	Investigar y analizar los fundamentos del método RULA.	Documento que detalla el funcionamiento del método RULA.	Revisión bibliográfica mediante libros, artículos científicos y sitios web.
	Analizar estudios de casos sobre automatización de evaluación ergonómica.	Reporte comparativo y crítico de soluciones similares.	Revisión de literatura y estudios técnicos.
	Estudiar sensores de movimiento, microcontroladores y motores gráficos.	Cuadro comparativo técnico con ventajas y desventajas.	Análisis técnico con hojas de datos y manuales.
	Identificar requerimientos funcionales y no funcionales del sistema.	Lista de requerimientos del sistema.	Técnicas de especificación de requerimientos, entrevistas, brainstorming.
Diseñar el modelo conceptual y funcional del sistema automatizado de evaluación de condiciones ergonómicas basado en el método RULA, estableciendo los requerimientos técnicos, tecnológicos y software necesario para su implementación.	Diseñar la arquitectura general del sistema (ESP32 + Unity).	Diagrama de bloques del sistema.	Diagramación con Lucidchart, Draw.io o similar.
	Diseñar la lógica embebida: lectura de sensores, uso de DMP, multitarea.	Diagrama de flujo de procesos embebidos y código documentado..	Arduino IDE, análisis estructurado, diagramas de flujo.
	Diseñar la lógica en Unity: lectura de WebSocket, modelo 3D, cálculo de ángulos.	Diagrama de flujo y scripts funcionales en Unity.	Unity3D, C#, WebSocketSharp, prototipo gráfico.
	Implementar el prototipo físico con sensores MPU9250 y ESP32.	Prototipo funcional que adquiere datos en tiempo real.	Ensamble electrónico, soldadura, multímetro, prueba de hardware.
	Implementar visualización 3D del cuerpo humano con rotaciones relativas.	Representación en Unity con modelo humanoide.	Modelado 3D, animación por cuaterniones, conversión a Euler.
	Implementar cálculo automático de puntaje RULA modular en Unity.	Sistema de evaluación automática integrada.	Programación estructurada, lógica condicional, interfaz gráfica.
Evaluar el funcionamiento del prototipo implementado mediante pruebas en condiciones de uso reales.	Realizar pruebas del sistema en condiciones de uso reales.	Datos recopilados del prototipo en condiciones reales.	Pruebas de campo y observación directa usando el prototipo.
	Comparar resultados del sistema con evaluaciones manuales.	Informe comparativo entre sistema automatizado y manual.	Comparación y análisis estadístico con tablas comparativas.
	Analizar los datos de las pruebas y generar conclusiones.	Informe de análisis con conclusiones sobre efectividad.	Análisis de datos e interpretación con herramientas de software y reportes de resultados.

3 FUNDAMENTACIÓN TEÓRICA

3.1 AUTOMATIZACIÓN DE LA EVALUACIÓN ERGONÓMICA

La automatización de la evaluación ergonómica mediante el método RULA (Evaluación Rápida de las Extremidades Superiores) ha avanzado significativamente gracias a tecnologías emergentes como la inteligencia artificial (IA) y la visión por computadora. Estas innovaciones permiten realizar evaluaciones más rápidas, precisas y eficientes, reduciendo la dependencia de observaciones manuales y subjetivas

3.1.1 Introducción a sistemas de monitoreo en tiempo real

Los sistemas de monitoreo son dispositivos tecnológicos que recopilan, procesan y analizan datos de un entorno físico a Tiempo real. el sistema debe procesar la información y dar un resultado dentro de un plazo de tiempo específico y predecible. Esta información que viaja debe tener el tiempo más bajo, si los datos llegan tarde, la información deja de ser útil.

La capacidad de obtener datos en tiempo real representa una ventaja puesto que estos datos son instantáneos. Una vez los datos se los tenga en cuestión de milisegundos se puede tomar decisiones de manera más efectiva, si lo ponemos en un contexto de industria se puede reaccionar rápidamente a situaciones inseguras, optimizando así la prevención de accidentes y mejorando la seguridad general en el ambiente laboral [1].

Si desglosamos punto a punto, un sistema de monitoreo en tiempo real da una foto del momento que está sucediendo, también recopila grandes volúmenes de datos a lo largo del tiempo. Esto permite a los expertos analizar tendencias, situaciones, identificar patrones de riesgo en tareas específicas o a lo largo, esto es ventajoso en un turno de trabajo, logrando tomar decisiones estratégicas para mejorar el diseño del puesto de trabajo o los procesos laborales.

3.1.2 Relevancia de la automatización en ergonomía.

La automatización ha transformado la ergonomía, pasando de ser un proceso predominantemente manual y subjetivo a uno cuantitativo. Al integrar tecnología como sensores, inteligencia artificial y sistemas de monitoreo, mejora la eficiencia de la automatización en la ergonomía.

En la industria es parte significativa la automatización, al mejorar que los procesos secuenciales o repetitivos sean automatizados representan tiempo, costos y efectividad. En todo el mundo siempre buscando la mejora continua de los procesos ya sea por medio de complejos y costosos robots hasta herramientas semiautomáticas que agilicen las tareas a los operadores y las empresas [2].

Los sistemas que ya son automatizados pueden recopilar un gran volumen de datos a lo largo del tiempo y en parte no es necesario de mucha supervisión humana. Estos big data posturales pueden ser analizados para identificar tendencias, patrones de riesgo en tareas específicas o la efectividad de una intervención ergonómica. Este tipo de análisis a gran escala permite a las empresas tomar decisiones estratégicas más informadas para el diseño de estaciones de trabajo, la rotación de tareas y la capacitación del personal.

Por lo tanto, no solo previene lesiones a nivel individual, sino que también mejora la gestión de la seguridad a nivel organizacional esto representa una ventaja a la industria que lo implementa, los cambios en organizaciones son constantes y si se logra automatizar la mayoría de los procesos representaría excelencia y calidad.

3.1.3 Antecedentes o trabajos similares a nivel internacional

En base a la automatización en la ergonomía se encontró dos trabajos los cuales se detallan en la Tabla 3.1, en estos logran tener dos puntos clave que buscan resolver:

- **Analysis of motion modelling approaches for industrial robot applications:** Los autores desarrollaron un sistema con sensores IMU para analizar los movimientos de un trabajador en tareas industriales. Su objetivo era evaluar los riesgos de lesiones musculoesqueléticas de forma automática y en tiempo real[3].
- **Using Accelerometer Data to Tune the Parameters of an Extended Kalman Filter for Optical Motion Capture: Preliminary Application to Gait Analysis:** El proyecto utilizaba los datos del acelerómetro para ajustar los parámetros de un filtro de Kalman Extendido (EKF). Este filtro es una herramienta matemática que se usa comúnmente para estimar el estado de un sistema (en este caso, la posición y orientación del cuerpo) a partir de mediciones ruidosas o imprecisas [4].

Tabla 3.1 Estudios relacionados.

Temas	Analysis of motion modelling approaches for industrial robot applications	Using Accelerometer Data to Tune the Parameters of an Extended Kalman Filter for Optical Motion Capture: Preliminary Application to Gait Analysis
Lugar de la investigación	Publicado en IEEE Xplore. Trabajo de investigación original.	Publicado en MDPI. Es un artículo de revisión sistemática, que analiza y sintetiza múltiples estudios internacionales.
Objetivo	Diseñar y validar un sistema vestible con sensores IMU para monitorear en tiempo real los riesgos musculoesqueléticos en tareas industriales repetitivas.	Analizar y resumir el estado del arte de los sistemas basados en sensores para la evaluación ergonómica, identificando las tecnologías, metodologías y desafíos clave en el campo.
Resultados o Hallazgos	El sistema propuesto es viable y preciso para la detección automática de posturas de riesgo, lo que demuestra que esta tecnología puede reemplazar a los métodos observacionales tradicionales. Se validó su capacidad para evaluar riesgos posturales de manera efectiva.	La revisión destaca que los sistemas con sensores mejoran la objetividad y precisión de la evaluación ergonómica. Sin embargo, también identifica desafíos como la validación de los datos, la usabilidad de los dispositivos y la necesidad de una estandarización en las metodologías de análisis.

El primer trabajo es un estudio práctico que creó un sistema para monitorear el riesgo de lesiones musculoesqueléticas en tiempo real usando sensores IMU. El objetivo principal fue diseñar y validar que esta tecnología es una alternativa viable y precisa a los métodos manuales de evaluación ergonómica. Por otro lado, el segundo trabajo es un estudio teórico que revisa y analiza múltiples investigaciones ya existentes sobre la misma área. Su objetivo principal fue resumir el estado actual del campo, destacando las ventajas como la objetividad y los desafíos como la falta de estandarización de usar sensores vestibles en ergonomía.

En esencia, el primer trabajo se centra en demostrar que un sistema funciona, mientras que el segundo se enfoca en entender qué se ha hecho y cuáles son las tendencias y problemas del campo.

3.1.4 Sistemas de referencia biomecánica.

En biomecánica, una cadena cinemática es un modelo teórico que representa el cuerpo humano como una serie de segmentos rígidos, huesos que están unidos por articulaciones.

De esta forma, el cuerpo es analizado como un sistema de eslabones interconectados que forman una estructura

El sistema de referencia biomecánico Es un sistema de coordenadas estandarizado permite describir de forma precisa la posición y orientación de cada segmento corporal. La correcta definición y calibración de los sistemas de referencia son esenciales para interpretar los datos de sensores y calcular los ángulos articulares, los cuales son la base para la evaluación ergonómica [5].

3.1.5 Tipos de representación de orientación.

Antes de hablar de los tipos de representación de orientación es necesario hacer una comparativa entre los diferentes modelos para conocer sus características para esto la Tabla 3.2 detalla cada modelo de representación y las ventajas y desventajas.

Tabla 3.2 Formas de representar orientación.

Tipo de Representación	Descripción	Ventajas	Desventajas/Uso
Ángulos de Euler	Tres rotaciones consecutivas alrededor de los ejes X, Y y Z.	Fácil de entender e interpretar por los humanos.	Sufren de Gimbal Lock, un bloqueo que causa movimientos erráticos. No son recomendables para animaciones en tiempo real.
Cuaterniones	Una representación de rotación en 4 dimensiones.	Evitan por completo el Gimbal Lock. Son muy eficientes para la interpolación (rotación suave). Estándar en motores como Unity.	Son difíciles de visualizar y no son intuitivos de entender.
Matrices de Rotación	Una matriz de 3x3 que representa una orientación.	Ideales para transformar puntos y vectores. No tienen Gimbal Lock.	Son computacionalmente más costosas que los cuaterniones para la interpolación y pueden acumular errores.

Para representar la orientación y la rotación de un objeto en tu entorno 3D haciendo referencia al avatar en Unity, existen diferentes métodos matemáticos. Los más comunes son los Ángulos de Euler, los Cuaterniones y las Matrices de Rotación. Mientras que los ángulos de Euler son intuitivos para nosotros, sufren de un problema grave llamado Gimbal Lock, que los hace poco confiables para movimientos complejos y en tiempo real. Los cuaterniones, aunque menos intuitivos, resuelven este problema y son el estándar en aplicaciones como la tuya, ofreciendo una rotación suave y eficiente. Por su parte, las matrices de rotación son más robustas para transformaciones, pero son menos eficientes para la animación continua[6].

3.2 IMU (UNIDAD DE MEDICIÓN INERCIAL).

La IMU es la Unidad de Medición Inercial, es un dispositivo electrónico que integra varios sensores para medir y reportar el movimiento de un objeto. Funciona combinando datos de un acelerómetro que mide la aceleración y la gravedad, un giroscopio que mide la velocidad de rotación, y a menudo un magnetómetro que mide el campo magnético terrestre como brújula. La calibración de sensores es un método para mejorar el rendimiento de un sensor eliminando errores estructurales de sus mediciones. Los errores estructurales son diferencias entre la salida esperada y la salida medida de un sensor, que se muestran consistentemente cada vez que se realiza una nueva medición. Al eliminar los errores estructurales, la calibración proporciona un medio para mejorar la precisión general de los sensores subyacentes. [7]

3.2.1 Principio de funcionamiento:

- Acelerómetro: Mide la aceleración y la gravedad.
- Giroscopio: Mide la velocidad de rotación.
- Magnetómetro: Mide el campo magnético de la Tierra, funcionando como una brújula.

3.2.2 MPU9250.

El MPU-9250 es un módulo de 9 ejes que combina en un solo chip los tres sensores esenciales para el seguimiento de movimiento, lo que lo hace compacto y eficiente. Sus principales características son:

- **Sensores integrados:** Contiene un acelerómetro de 3 ejes, un giroscopio de 3 ejes y un magnetómetro de 3 ejes. Esta combinación permite medir la aceleración, la velocidad angular y la dirección del campo magnético.
- **Alta precisión:** Utiliza un convertidor analógico-digital (ADC) de 16 bits, lo que le da una alta resolución para detectar incluso los cambios de movimiento más sutiles.
- **Bajo consumo de energía:** Su diseño de bajo consumo lo hace ideal para dispositivos portátiles y proyectos que funcionan con baterías.
- **Interfaz de comunicación:** Se comunica con el microcontrolador (como tu ESP32) a través de los protocolos I2C o SPI, lo que facilita su conexión e integración.

Incorpora el Digital Motion Processor (DMP) es un procesador de hardware integrado dentro del propio chip del MPU-9250. Su función principal es descargar de trabajo al microcontrolador principal y encargarse de tareas complejas directamente en el sensor[8].

3.3 ALGORITMOS DE ORIENTACIÓN Y FILTRADO

3.3.1 Algoritmo Madgwick

Es un algoritmo de fusión de sensores y computacionalmente eficiente, diseñado para obtener una orientación estable. Se basa en un filtro de gradiente descendente que estima la orientación del sensor generalmente en cuaterniones. Su principal característica es que corrige activamente los errores del giroscopio, la deriva usando los datos del acelerómetro y el magnetómetro.

3.3.2 Algoritmo Mahony

Similar al Madgwick, es otro algoritmo de fusión de sensores que utiliza un filtro de retroalimentación feedback para estimar la orientación. Corrige la orientación del giroscopio usando los datos del acelerómetro y el magnetómetro para compensar la deriva. A diferencia del Madgwick, su método de corrección es más directo y puede ser más sencillo de implementar en algunos casos.

3.3.3 Filtro mediante DMP (hardware)

No es un algoritmo de software, sino un procesador dedicado y programado que viene integrado en el chip de la IMU, como en el MPU-9250 de InvenSense. Ejecuta un algoritmo de fusión de sensores directamente en el hardware. Realiza las correcciones y combinaciones de los datos de los sensores antes de que la información llegue al microcontrolador.

3.3.4 Filtro complementario.

Es un filtro más simple que los anteriores, basado en la combinación de las fortalezas de dos sensores para compensar sus debilidades. Asume que la orientación del acelerómetro es más confiable a baja frecuencia (cuando el sensor se mueve despacio) y que la del giroscopio es más confiable a alta frecuencia (cuando el sensor gira rápidamente). El filtro combina ambos flujos de datos, usando una ponderación para cada uno. Los datos del acelerómetro corrigen la deriva a largo plazo, mientras que los datos del giroscopio dan una respuesta rápida y sin latencia [9].

3.4 ESP32

El ESP32 es un microcontrolador que se ha estado ocupando en proyectos IoT debido a su gran capacidad de procesamiento y sus múltiples opciones de conectividad.

3.4.1 Arquitectura del ESP32

- **CPU Dual-Core:** El ESP32 cuenta con un procesador de doble núcleo Xtensa LX6. Esto es fundamental para tu proyecto. Un núcleo puede dedicarse a tareas de alta prioridad, como procesar los datos de los sensores y ejecutar el filtro, mientras que el otro se encarga de la comunicación inalámbrica (Wi-Fi), garantizando que no haya interrupciones en la transmisión de datos[10].
- **Conectividad inalámbrica:** Integra un módulo Wi-Fi (802.11 b/g/n) y Bluetooth (clásico y de baja energía, BLE). Esto te permite transmitir los datos de los sensores a tu computadora de forma inalámbrica, sin la necesidad de cables.
- **Memoria:** Dispone de una memoria interna que incluye SRAM (memoria de acceso aleatorio estática) para el procesamiento rápido de datos y Flash para almacenar el firmware y el código de tu programa.
- **Periféricos:** Cuenta con una gran variedad de periféricos, incluyendo:

- a. GPIOs (Pines de entrada/salida de propósito general) para conectar los sensores.
- b. ADC (Convertidor analógico-digital) y DAC (Convertidor digital-analógico).
- c. Múltiples interfaces de comunicación como SPI, I2C y UART, que son los protocolos que usarás para conectar el sensor MPU-9250[10].

3.5 INTERNET DE LAS COSAS (IoT)

El Internet de las Cosas (IoT) es una red de dispositivos físicos, vehículos y otros objetos equipados con sensores, software y conectividad de red que les permite recopilar y transferir datos. En el contexto de este proyecto es un sistema en el que los sensores IMU y el microcontrolador ESP32 actúan como los objetos que capturan datos del entorno físico para luego enviarlos a un sistema de software para su análisis y visualización [11].

Para crear un sistema IoT se requiere de un método de comunicación, modelo cliente-servidor y un paquete que contenga la información con la que se trabaja, estos componentes están detallados en la Tabla 3.3.

Tabla 3.3 Elementos para el sistema IoT.

Concepto	Función	Características Clave
Modelo Cliente-Servidor	Define la relación entre los dispositivos. El cliente (ESP32) solicita la conexión y envía los datos de los sensores, mientras que el servidor (la computadora) recibe, procesa y visualiza esa información en el avatar 3D.	Establece roles claros y una arquitectura de comunicación bien definida, fundamental para la estabilidad del sistema.
Protocolo WebSocket	Es el canal de comunicación que permite el flujo continuo y en tiempo real de datos de los sensores del ESP32 a la computadora.	A diferencia de protocolos tradicionales, mantiene una conexión persistente, garantizando una baja latencia y un movimiento fluido del avatar sin interrupciones.
JSON	El ESP32 empaqueta los datos de los sensores (valores de los cuaterniones) en un objeto JSON, y el software en la computadora lo lee y lo interpreta.	Es un formato de texto ligero y fácil de leer tanto para máquinas como para humanos. Asegura que los datos se envíen de manera estructurada y que ambos dispositivos los entiendan sin problemas.

3.6 REPRESENTACIÓN 3D Y UNITY

3.6.1 Unity.

Unity es un potente motor de videojuegos y un entorno de desarrollo multiplataforma. Ofrece herramientas integradas para la creación de contenido 3D y 2D, así como un robusto sistema de scripting en C# [12].

Unity es el software que actúa como el servidor y la interfaz de usuario. Recibe los datos de los sensores a través de la comunicación inalámbrica, los procesa y los utiliza para manipular el modelo 3D del trabajador en tiempo real. Su capacidad para manejar animaciones y scripts lo hace ideal para recrear el movimiento corporal y para aplicar las reglas de evaluación ergonómica.

3.6.2 Modelos humanoides.

Un modelo humanoide en Unity es un tipo de modelo 3D que tiene una estructura esquelética que se asemeja a la de un cuerpo humano real cuya estructura de huesos y articulaciones permite que el modelo se anime y se mueva de manera realista. La representación de estos modelos humanoides se los conoce como avatar que representa al trabajador. La principal ventaja de usar un modelo de este tipo es que Unity tiene un sistema de "rigging" que facilita la asignación de las rotaciones de los sensores IMU a cada articulación correspondiente (codo, hombro, rodilla, etc.). Esto simplifica enormemente el proceso de animación en tu proyecto.

3.6.3 Aplicación de rotaciones en modelos 3D.

Esta es la etapa final del procesamiento de datos. Consiste en tomar la información de orientación que se ha calculado con los sensores y aplicarla directamente a las articulaciones del modelo humanoide. A través de scripts en C#, Unity recibirá los valores de cuaterniones de cada sensor (uno para el antebrazo, otro para el torso, etc.) [13].

Luego, se aplican estos cuaterniones a las articulaciones correspondientes del avatar. Esta aplicación de rotaciones en tiempo real es lo que hará que el avatar 3D se mueva en sincronía con el trabajador, permitiendo una visualización instantánea y precisa de la postura.

3.7 ERGONOMÍA Y RIESGOS MÚSCULO-ESQUELÉTICOS.

3.7.1 Definición de ergonomía.

Según la ISO 6385:2016: La norma Principios ergonómicos en el diseño de sistemas de trabajo define la ergonomía como el conjunto de principios básicos para el diseño de sistemas de trabajo que buscan optimizar el bienestar, la seguridad y la salud humana, a la vez que se tienen en cuenta los aspectos tecnológicos, económicos y organizativos[14].

Según la Asociación Internacional de Ergonomía (IEA): La IEA la define como la disciplina científica relacionada con la comprensión de las interacciones entre los seres humanos y otros elementos de un sistema, y la profesión que aplica principios, datos y métodos de diseño para optimizar el bienestar humano y el rendimiento general del sistema [15].

En palabras simples se puede decir que la ergonomía busca adaptar el trabajo a la persona, no la persona al trabajo.

3.7.2 Importancia de la ergonomía en la prevención de TME (Trastornos Músculo-Esqueléticos).

La ergonomía es crucial para prevenir los Trastornos Músculo-Esqueléticos (TME). Los TME son lesiones o afecciones que afectan a los músculos, tendones, articulaciones y nervios, y son la principal causa de ausentismo laboral y discapacidad en muchos países.

La importancia de la ergonomía radica en que:

- **Identifica factores de riesgo:** Permite detectar posturas forzadas, movimientos repetitivos, la aplicación de fuerza excesiva y la falta de periodos de descanso, todos ellos factores que contribuyen al desarrollo de los TME.
- **Diseña soluciones preventivas:** A través del rediseño del puesto de trabajo, las herramientas o propone soluciones que reducen la carga física sobre el cuerpo del trabajador.
- **Mejora la calidad de vida:** Al prevenir estas lesiones, no solo se reduce el impacto económico para las empresas, sino que se mejora significativamente la salud y el bienestar del trabajador a largo plazo.

Según este artículo que habla sobre la revisión sistemática, concluye que las intervenciones ergonómicas son efectivas para reducir los TME. Al ser una intervención ergonómica de monitoreo en tiempo real, está basado en evidencia científica y tiene un impacto real en la prevención de riesgos [16].

3.7.3 Método RULA.

El método RULA es una forma de evaluación de posturas individuales o conjuntos de posturas, también se consideran ciertas condiciones de riesgo que originan una elevada carga postural [17].

El método consiste en evaluar ciertos ángulos del cuerpo y aplicar tablas, mediante este proceso se obtiene una puntuación con la cual se establece un nivel de actuación lo que determinara si la postura es aceptable o si requiere cambios o rediseños en el puesto de trabajo.

La Tabla 3.4 indica la puntuación que debe aplicarse en la evaluación del brazo en función de los ángulos presentados por esta extremidad. Para las demás evaluaciones se aplica un criterio similar.

Una vez obtenido la puntuación de cada sección del cuerpo se aplica la Tabla 3.5 en la que se combina la puntuación de todas las secciones de las extremidades superiores que se utiliza para la puntuación final del método RULA.

Tabla 3.4 Puntuación del brazo.

Posición	Puntuación
Desde 20° de extensión a 20° de flexión	1
Extensión >20° o flexión >20° y <45°	2
Flexión >45° y 90°	3
Flexión >90°	4

Tabla 3.5 Combinación de puntuación de extremidades superiores.

		Muñeca							
		1		2		3		4	
		Giro de Muñeca		Giro de Muñeca		Giro de Muñeca		Giro de Muñeca	
Brazo	Antebrazo	1	2	1	2	1	2	1	2
1	1	1	2	2	2	2	3	3	3
	2	2	2	2	2	3	3	3	3
	3	2	3	3	3	3	3	4	4
2	1	2	3	3	3	3	4	4	4
	2	3	3	3	3	3	4	4	4
	3	3	4	4	4	4	4	5	5
3	1	3	3	4	4	4	4	5	5
	2	3	4	4	4	4	4	5	5
	3	4	4	4	4	4	5	5	5
4	1	4	4	4	4	4	5	5	5
	2	4	4	4	4	4	5	5	5
	3	4	4	4	5	5	5	6	6
5	1	5	5	5	5	5	6	6	7
	2	5	6	6	6	6	7	7	7
	3	6	6	6	7	7	7	7	8
6	1	7	7	7	7	7	8	8	9
	2	8	8	8	8	8	9	9	9
	3	9	9	9	9	9	9	9	9

3.7.4 Legislación aplicable en Ecuador

La normativa ecuatoriana obliga a los empleadores a garantizar la seguridad y salud de sus trabajadores, incluyendo la prevención de riesgos ergonómicos.

El Reglamento de Seguridad y Salud en el Trabajo (Decreto Ejecutivo 255), emitido el 2 de mayo de 2024, actualiza el marco normativo anterior [18]. Este decreto fortalece la cultura de prevención y busca desarrollar políticas públicas que impulsen la seguridad y salud laboral. Este reglamento busca:

- Impulsar la seguridad y salud laboral: El decreto establece el marco para el desarrollo de políticas que promueven un entorno de trabajo seguro.
- Registro de Enfermedades Profesionales: Se crea un sistema para registrar enfermedades profesionales y accidentes. Los Trastornos Músculo-Esqueléticos (TME) son una de las principales causas de ausentismo laboral por lo que se debe tener un registro de las condiciones de riesgo que podrían provocar estas afecciones.

Código del Trabajo: En su Título IV, establece las obligaciones del empleador en materia de seguridad y salud, incluyendo la adopción de medidas adecuadas para prevenir los accidentes y enfermedades laborales [19].

La legislación ecuatoriana, principalmente a través del Reglamento de Seguridad y Salud en el Trabajo (Decreto Ejecutivo 255) y el Código del Trabajo, exige a los empleadores proteger a sus trabajadores de riesgos laborales, incluyendo los ergonómicos. Estos documentos establecen la obligatoriedad de la prevención y la creación de políticas de seguridad para mitigar los Trastornos Músculo-Esqueléticos (TME).

4 MÉTODOS Y PROCEDIMIENTOS

Este trabajo se enfoca en el desarrollo de un sistema automatizado para evaluación de condiciones ergonómicas, para esto se aplicará el uso de sensores inerciales, microcontroladores para procesamiento de datos y softwares con soporte para visualización de modelos 3D en tiempo real. Se empleará un enfoque cuantitativo-experimental con el que se busca diseñar, implementar, y probar un prototipo funcional que permita realizar la evaluación de diferentes posturas de un trabajador en base al método RULA.

4.1 METODOLOGÍA DE INVESTIGACIÓN

4.1.1 Metodología Waterfall (Cascada)

La metodología seleccionada para el desarrollo del proyecto es la llamada “Metodología Waterfall” la cual consiste en una metodología para gestión de proyectos que se divide en distintas fases. Cada fase comienza una vez ha concluido la anterior como se puede visualizar en la Figura 4. 1.

Esta metodología fue seleccionada ya que al tener una estructura secuencial permite mantener un orden claro y control sobre el cumplimiento de los requerimientos, ya que para esta metodología los requerimientos deben estar bien definidos desde el inicio del proyecto, permitiendo así definir etapas claras que guían el desarrollo del proyecto.



Figura 4. 1 Los 6 pasos del modelo Waterfall.

La metodología Waterfall, por lo general se visualiza en forma de diagrama de flujo o de Gantt. Se la llama Waterfall o cascada porque cada tarea cae en cascada sobre el paso siguiente. En un diagrama de Gantt se puede ver cómo la fase anterior cae en cascada sobre la siguiente.

Para aplicar esta metodología se definieron las siguientes etapas:

- **Análisis:** Identificación de necesidades que debe cumplir el sistema automatizado.
- **Diseño:** Propuesta de arquitectura de hardware y software, selección de sensores y plataforma de visualización.
- **Implementación:** Desarrollo del sistema de adquisición de datos, transmisión por WebSocket e interpretación en Unity.
- **Pruebas:** Validación del sistema con posturas simuladas, análisis de confiabilidad, latencia y respuesta.
- **Ajustes:** Corrección de errores, optimización de la comunicación y afinación de parámetros para mejorar la estabilidad del sistema.

4.2 ETAPA DE ANÁLISIS

En la etapa de análisis se realizó una evaluación de la situación laboral en relación a riesgos ergonómicos, considerando tanto tareas repetitivas como posturas forzadas y manipulación de cargas, que son factores que afectan a la ergonomía y presentan un riesgo a la salud y son factores que permiten el desarrollo de trastornos musculoesqueléticos (TME). Según datos de la OMS, los TME representan una de las principales causas de ausentismo laboral a nivel mundial, además se estima que más del 60% de los trabajadores en sectores industriales experimentan algún tipo de molestia musculoesquelética a lo largo de su vida laboral.

A nivel local, informes del IESS reportan que una proporción significativa de las incapacidades temporales registradas estaban relacionadas con dolencias físicas atribuibles a malas posturas mantenidas en el tiempo. Estos problemas no solo afectan la salud del trabajador, sino que también repercuten en la productividad y aumentan los costos operativos por compensaciones médicas y reemplazos temporales.

En este contexto, se identificó que, si bien existen metodologías de evaluación como el método RULA, su aplicación tradicional es manual y requiere de observación directa por

parte de un experto o un equipo completo por lo que resulta poco practica para un uso continuo o en tiempo real en entornos de trabajo dinámicos. Este método demanda que el evaluador realice una observación estática del trabajador durante una tarea, tomando nota de los ángulos de articulaciones como hombros, codos, muñecas, tronco y cuello y luego procese los datos mediante tablas para obtener un puntaje. Esto implica varias limitaciones concretas como:

- **Tiempo de aplicación prolongado:** una sola evaluación puede demorar entre 10 a 20 minutos dependiendo de la complejidad de la postura y la experiencia del evaluador.
- **Dependencia de un evaluador capacitado:** la subjetividad puede influir en la precisión, especialmente si no se cuenta con herramientas de medición exactas.
- **Imposibilidad de evaluar en tiempo real:** la aplicación del método es posterior a la observación o revisión de imágenes y videos, lo que impide la toma de decisiones inmediatas.
- **Costos asociados:** se requiere personal capacitado, interrupción de la jornada laboral y herramientas de observación. En un estudio de 2022 publicado en el *Journal of Occupational Safety* reporta que el costo promedio por implementación de evaluaciones ergonómicas tradicionales en pymes industriales supera los 150 USD por trabajador en procesos de auditoría puntual.

Con todas estas consideraciones se logró identificar una oportunidad para mejorar este proceso automatizándolo mediante sensores inerciales que capturen en tiempo real los movimientos del trabajador y permitan realizar la evaluación de forma automática.

Para lograr esto se definieron los siguientes requerimientos funcionales del sistema:

- Capturar datos de orientación del cuerpo humano a través de sensores inerciales.
- Transmitir los datos de forma inalámbrica en tiempo real.
- Representar en un modelo 3D las posturas del trabajador.
- Evaluar automáticamente las posturas con base en el método RULA.

Adicionalmente se establecieron los requerimientos no funcionales:

- El sistema debe ser modular y funcionar aunque no estén conectados todos los sensores.

- Debe tener bajo costo, y facilidad de uso en campo.
- Debe operar con baja latencia.
- La visualización y evaluación deben ser comprensibles por personal no técnico.

4.3 ETAPA DE DISEÑO

La etapa de diseño se centró en la definición de la arquitectura física y lógica del sistema de evaluación automática. Se realizó la selección de los componentes electrónicos y plataformas de software más adecuadas para garantizar la precisión, estabilidad y modularidad del sistema. A continuación, se detallan los elementos considerados en esta fase.

4.3.1 Arquitectura del sistema

El sistema fue concebido bajo una arquitectura modular distribuida, compuesta por sensores inerciales MPU9250 conectados a un microcontrolador ESP32 a través de dos multiplexores TCA9548A, encargados de gestionar múltiples canales I2C. La ESP32 actúa como unidad central de adquisición y envío de datos a través de una conexión WiFi mediante el protocolo WebSocket. Los datos son recibidos en un computador donde un entorno desarrollado en Unity interpreta y visualiza en tiempo real las posturas corporales del trabajador en un modelo humanoide 3D. Este modelo aplica la evaluación RULA automáticamente utilizando la orientación relativa entre sensores.

El flujo de datos puede resumirse como se ve en la Figura 4.1, en ella se detalla el flujo seguido para obtener la posición del cuerpo en tiempo real y finalmente obtener la calificación dada por el método RULA, para ello el primer punto corresponde a los sensores, distribuidos en todo el cuerpo según los requerimientos del método de evaluación, estos datos obtenidos deben llegar a la unidad de procesamiento mediante comunicación I2C, dado las limitaciones de los canales disponibles en los sensores se aplica el uso multiplexores para leer los datos desde diferentes canales a pesar de que los sensores tengan la misma dirección I2C, dentro de la unidad de procesamiento se empaqueta la información y se envía por Wifi vía WebSocket y es recibida en el cliente en Unity para ser mostrada en un modelo 3D y aplicar la evaluación RULA.

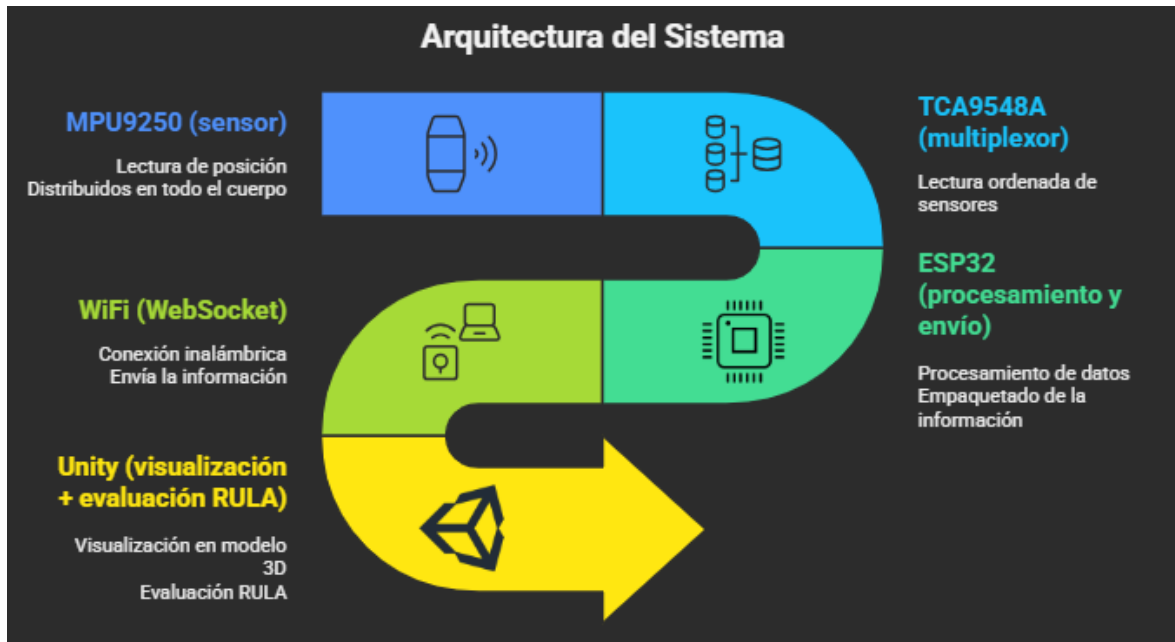


Figura 4.1 Componentes principales del sistema

4.3.2 Fundamentos matemáticos y representación de orientación

Para la lectura del movimiento entregado por los sensores se aplica el uso del cuaternión como representación de la orientación tridimensional. Para la interpretación de la posición se proponen principalmente 2 soluciones que son el uso de cuaterniones y ángulos de Euler, sin embargo, para el desarrollo de este proyecto se optó por el cuaternión para el cálculo interno mientras que los ángulos de Euler se utilizan para que el usuario pueda entender fácilmente la información dada por los sensores, esto debido principalmente a limitaciones presentes en los ángulos de Euler.

Representación con cuaterniones.

Un cuaternión corresponde a una extensión de los números complejos, utilizando 3 ejes imaginarios y un componente escalar se puede representar rotaciones tridimensionales de forma compacta y evitando problemas como el gimbal lock.

Un cuaternión unitario que representa una rotación puede definirse de la forma descrita en la ecuación (4.1).

$$q = w + xi + yj + zk \quad (4.1)$$

Donde w es el componente escalar y xi, yj, zk son los componentes vectoriales.

Otra forma de representar un cuaternión es la presentada en la ecuación (4.2).

$$q = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)(xi + yj + zk) \quad (4.2)$$

Donde θ es el ángulo de rotación y xi, yj, zk representa el eje de rotación normalizado.

Para aplicar una rotación q a un vector v , se utiliza la operación definida por la ecuación (4.3).

$$v_{rotado} = q * v * q^{-1} \quad (4.3)$$

Donde v corresponde al vector de rotación y q es el cuaternión.

Cuaterniones sobre ángulos de Euler.

Los ángulos de Euler son rotaciones sucesivas aplicadas sobre los ejes x, y, z y si bien son adecuados para el entendimiento de la rotación de un objeto en un plano tridimensional para un observador, al aplicarlos en un entorno de programación se presenta un problema crítico conocido como gimbal lock. El gimbal lock ocurre al aplicar rotaciones de forma continua lo que provoca que eventualmente 2 ejes se alineen perdiendo así un grado de libertad y afectando a la precisión del sistema.

Si en un movimiento del sensor se aplica rotación únicamente en el eje y, los ejes x y z van a rotar y terminar alineándose provocando que una siguiente rotación aplicada en alguno de estos 2 ejes afecte a ambos generando un movimiento errático y aplicado únicamente sobre 2 ejes perdiendo así el movimiento tridimensional.

Debido a esta limitación considerable los ángulos de Euler quedan descartados para procesos de cálculo, envío de datos o movimiento del modelo y quedan asignados únicamente a presentar la información al usuario.

Un cuaternión aplica el cálculo multiplicando los ejes imaginarios como se muestra en la ecuación (4.3), al trabajar de esta forma se evita problemas como el gimbal lock además ya que se multiplican los ejes de la forma vista en la Figura 4.2 para obtener un vector normalizado, el orden de las multiplicaciones y la cantidad de rotaciones aplicadas no provoca variación o inconsistencia en el cálculo de la posición final y los ejes de referencia mantienen su posición, siendo el vector resultante el que determina la rotación del sensor.

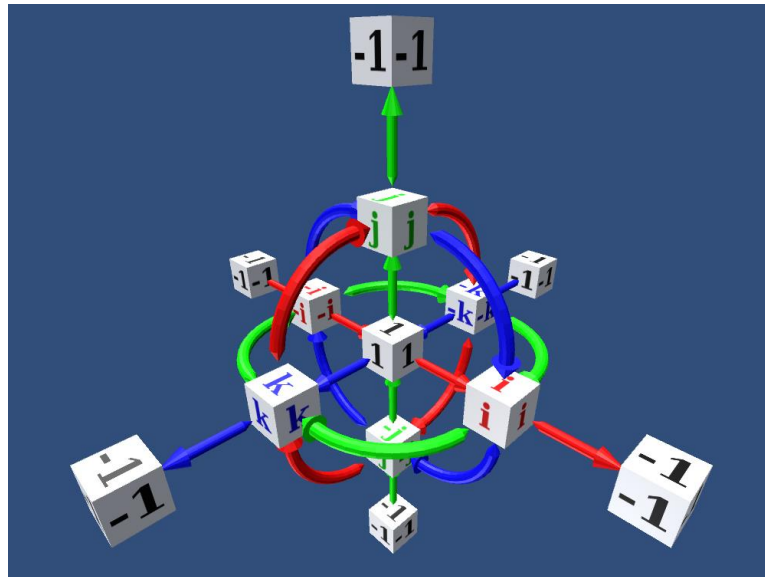


Figura 4.2 Modelo de cuaternión

Flujo matemático aplicado al sistema

Recordando la arquitectura presentada en la Figura 4.1 la ESP32 lee datos en forma de cuaternión desde el DMP presente en cada MPU9250, una vez obtenidas las lecturas esta información debe ser empaquetada para enviarse mediante WebSocket, para lograr esto se utiliza un JSON que contiene la id del sensor que envía datos y una estructura de cuaternión basada en la presentada en la ecuación (4.1), un ejemplo del paquete enviado para una rotación de 90° alrededor del eje Y sería:

```
{
  "id": 0,
  "q0": 0.7071,
  "q1": 0.0,
  "q2": 0.7071,
  "q3": 0.0
}
```

Para esta representación q_0 corresponde al componente escalar w y los componentes q_1 , q_2 , q_3 corresponden a xi , yj , zk respectivamente.

Dentro de Unity se aplican funciones que permiten modificar el comportamiento de la rotación para que el movimiento presentado en Unity coincida con el movimiento real de

los sensores MPU9250, esto es necesario ya que en la posición predeterminada los sensores tienen diferentes posiciones por lo que los ejes deben ser configurados.

1. **Remapeo de ejes:** se implementa un remapeo de ejes que permite cambiar los ejes leídos, es decir, aplicar los datos recibidos para el eje x en el eje z, por ejemplo, esto debido a la posición física del sensor.
2. **Inversión de ejes:** se aplica un signo negativo al componente seleccionado, esto es útil por ejemplo al trabajar con simetrías como puede ser el sensor ubicado en el brazo izquierdo y derecho ya que en ese caso un mismo movimiento del sensor puede expresar un movimiento diferente para el modelo 3D
3. **Calibración (offset):** se obtiene la rotación actual de cada sensor y se la aplica como rotación base. Luego, cada rotación será ajustada de la forma descrita en la ecuación (4.4).

$$q_{final} = q_{offset} * q_{ajustado} \quad (4.4)$$

Donde q_{final} es la rotación aplicada al modelo, q_{offset} es la rotación obtenida en la calibración y $q_{ajustado}$ es la rotación leída en tiempo real.

4. **Cálculo de rotación relativa:** para representar el movimiento no se puede aplicar una rotación directa, es necesario aplicar una rotación relativa entre 2 partes del cuerpo, para esto se designa un sensor padre y un sensor hijo, con esta estructura se aplica la ecuación (4.5) y se obtiene este ángulo relativo, por ejemplo, antebrazo respecto al brazo.

$$q_{relativa} = q_{base}^{-1} * q_{hijo} \quad (4.5)$$

Donde q corresponde a un cuaternión.

5. **Conversión a ángulos de Euler:** se aplica únicamente para visualización en pantalla y aplicación de tablas de evaluación RULA.

Ejemplo completo de cálculo de rotación aplicada:

Se calculará la rotación aplicada al brazo respecto al torso, los cuaterniones recibidos son:

Cuaternión que expresa una rotación de 45° sobre el eje z en el torso:

$$q_{torso} = (0.9239, 0, 0.3826, 0)$$

Cuaternión que expresa una rotación de 90° sobre el eje x en el brazo:

$$q_{brazo} = (0.7071, 0.7071, 0, 0)$$

Para el cálculo se aplicará la ecuación (4.5) por lo que primero se calcula la inversa de q_{torso} :

$$q^{-1} = (w, -x, -y, -z)$$

$$q_{torso}^{-1} = (0.9239, 0, -0.3826, 0)$$

Luego, se multiplica por q_{brazo} :

$$q_{relativo} = (0.9239, 0, -0.3826, 0) * (0.7071, 0.7071, 0, 0)$$

$$q_{relativo} = (0.6533, 0.6533, -0.2706, 0)$$

Convertido en ángulos de Euler para mayor comprensión el resultado es:

$$Euler = (70^\circ, -30^\circ, 0^\circ)$$

4.3.3 Selección de algoritmos de orientación

Para la toma de datos correspondientes a la rotación de un objeto, en este caso diferentes partes del cuerpo humano, se está utilizando un sensor MPU9250 el cual integra acelerómetro, giroscopio y magnetómetro, sin embargo, para poder interpretar los lados leídos es necesario aplicar un algoritmo de fusión de sensores, del cual depende la precisión entregada. Para el sistema propuesto se ha tomado en cuenta 3 posibles enfoques: DMP, Madgwick y Mahony.

Digital Motion Processor (DMP).

El sensor MPU9250 integra dentro de su estructura un coprocesador que opera de forma autónoma y se encarga de ejecutar un algoritmo de fusión de sensores además cuenta con una memoria propia llamada FIFO (First In First Out) en la que se almacenan los datos procesados y listos para ser leídos por el microcontrolador. Este coprocesador es capaz de entregar como salida cuaterniones ya procesados lo que permite reducir la carga computacional sobre la ESP32.

Ventajas:

- El sensor es el encargado de procesar los datos crudos.
- Alta estabilidad en movimientos complejos.
- Salida directamente de cuaterniones desde la FIFO.

Desventajas

- Poca flexibilidad o acceso a la lógica interna.
- Configuración limitada.
- Dependencia del entorno debido al magnetómetro.

Filtro Madgwick.

Este es un filtro desarrollado por Sebastian Madgwick, el filtro tiene la capacidad de fusionar datos de giroscopios, acelerómetros y magnetómetros para determinar la rotación de un objeto, además tiene como base el uso de cuaterniones para evitar los problemas generados con ángulos de Euler. Su funcionamiento consiste una estimación comparando el vector de aceleración medido y el vector esperado agregando un término de corrección proporcional.

Ventajas:

- Alta precisión si la aceleración es moderada.
- Bajo consumo de recursos computacionales.
- Corrección ante interferencias en el magnetómetro.

Desventajas:

- Requiere ajustarse detallado del parámetro β .
- Mayor consumo de recursos frente al uso del DMP.

Filtro Mahony.

Este es un filtro complementario, es decir, se utiliza en conjunto con otro filtro para obtener un resultado final, en este caso se combina con el filtro Madgwick. El filtro Mahony se centra en un observador no lineal con un control PI (proporcional-integral).

Ventajas:

- Respuesta más suave.
- Mayor compatibilidad con señales con presencia de ruido.

Desventajas:

- Retraso ante movimientos bruscos
- Requiere ajuste de constantes PI.
- Se requiere un filtro adicional

La Tabla 4.1 resume las principales características de estos 3 algoritmos de fusión de sensores, una vez comparados y probados cada uno de ellos, se decantó por usar el DMP interno del sensor MPU9250 ya que al manejar tantos sensores la velocidad de lectura y cálculo requerido por los otros filtros puede llegar a saturar la ESP32 además al estar todos los componentes ubicados físicamente en diferentes posiciones se requiere de un ajuste fino para cada uno lo que puede conllevar errores en la precisión de la lectura y una carga aún mayor en el microcontrolador.

Tabla 4.1 Comparación de características entre filtros.

Algoritmo	Tipo	Magnetómetro	Precisión	Uso de CPU	Calibración	Flexibilidad
DMP	Hardware	Opcional	Media/Alta	Muy baja	Sí (interna)	Baja
Madgwick	Software	Opcional	Alta	Media	Sí	Alta
Mahony	Software	Opcional	Media	Media	Sí	Alta

4.3.4 Justificación de selección de componentes.

Dados los requerimientos del sistema, el filtro seleccionado entre otros factores se escogieron los componentes que más se adaptan al trabajo a realizar.

MPU9250: Sensor inercial de 9 grados de libertad que incluye acelerómetro, giroscopio y magnetómetro. Se eligió por su bajo costo, tamaño compacto y soporte para el procesamiento DMP interno, utilizado a través de la librería SparkFunMPU9250-DMP para obtener cuaterniones estables.

Multiplexores TCA9548A: Debido a la limitación del bus I2C (dirección fija de los sensores), fue necesario usar dos multiplexores, lo cual permite gestionar hasta 8 sensores por dispositivo sin conflicto de direcciones.

ESP32: Microcontrolador con capacidad WiFi integrada, alta velocidad de procesamiento, múltiples interfaces I2C y buena compatibilidad con Arduino. Permite mantener una conexión en tiempo real con la PC mediante WebSocket.

Unity: Plataforma de desarrollo 3D utilizada para visualizar al trabajador en un entorno virtual. Se utilizó un avatar humanoide compatible con rotaciones por cuaterniones. La lógica de evaluación RULA fue programada internamente en Unity de forma modular, dividiendo el cálculo en grupos A y B y un evaluador final.

4.3.5 Distribución y lógica de sensores.

Para el desarrollo del prototipo se implementó un sistema compuesto de 13 sensores MPU9250 distribuidos en todo el cuerpo, la Tabla 4.2 expresa la distribución de los sensores, cada uno tiene su propia ID que se utiliza como identificador del sensor específico y para cálculos posteriores se utiliza un sensor padre con el que se obtiene el ángulo relativo aplicando la ecuación (4.5).

Justificación ergonómica de la ubicación.

Para la evaluación RULA se divide en 2 grupos principales:

- **Grupo A:** incluye brazos superiores, antebrazos y muñecas.
- **Grupo B:** incluye cuello, tronco y piernas

Con esta división se consideran los movimientos requeridos en la evaluación para cada parte por lo que los sensores se ubican en función de la sección y el movimiento requerido:

- **Brazo y antebrazo:** utilizados para medir la flexión y elevación lateral.
- **Mano:** usados para inferir el ángulo de la muñeca.
- **Espalda baja:** se utiliza como referencia de orientación global. Sin este sensor todo el movimiento que no sea mirando hacia la dirección en la que se realizó la calibración será incorrecta.
- **Espalda y cabeza:** implementados para evaluar inclinaciones laterales y posturas del cuello, además de servir como referencia para movimiento de brazos
- **Piernas y pantorrillas:** se incluyen para tener un movimiento más realista del modelo 3D, sin embargo, para RULA pueden ser omitidos.

Esta distribución permite una lectura amplia del movimiento además, debido a la forma de evaluación del método RULA se aplica una estructura de padre-hijo mediante la cual se calcula el ángulo relativo entre extremidades, por ejemplo el sensor 1 correspondiente al antebrazo tiene como padre al sensor 0 que se ubica en el brazo, por lo que el ángulo medido sirve para aplicar el criterio de la Tabla 3.4

Tabla 4.2 Distribución de sensores en el cuerpo con ID

Distribución de sensores		
Extremidad	ID	ID PADRE
Extremidades superiores		
Brazo izquierdo	0	6
Antebrazo izquierdo	1	0
Mano izquierda	2	1
Brazo derecho	8	6
Antebrazo derecho	9	8
Mano derecha	10	9
Tronco y eje central		
Espalda baja (referencia de rotación)	5	6
Espalda	6	-1
Cabeza	7	6
Extremidades inferiores		
Pierna izquierda	3	5
Pantorrilla izquierda	4	3
Pierna derecha	11	5
Pantorrilla derecha	12	11

4.3.6 Estructura trabajo con microcontrolador.

El sistema está diseñado para funcionar utilizando una ESP32 como microcontrolador encargado de la lectura de datos, empaquetado de información y envío de datos en tiempo real. Este microcontrolador cuenta con varias características importantes para la ejecución de este proyecto, estas características son:

- Procesador dual-core 240MHz: característica usada para realizar tareas en paralelo y reducir la latencia entre el movimiento real y el mostrado en el modelo 3D.

- RAM de 512kB: capacidad suficiente para almacenar las lecturas de múltiples sensores y para manipulación de estructuras JSON.
- Wifi integrado: característica necesaria para la comunicación inalámbrica entre el prototipo y el cliente al que llega la información de los sensores.
- Interfaces I2C: la interfaz I2C del ESP32 posee 2 modos de trabajo que cambian la frecuencia a la que opera, modo estándar (100kHz) y modo rápido (400kHz) lo que ayuda a leer a mayor velocidad la información de cada uno de los sensores.

Uso de doble núcleo y multitarea

Dado que la ESP32 cuenta con 2 núcleos se puede asignar tareas específicas a cada núcleo con lo que se dividió de la siguiente forma:

Núcleo 0: asignado a la lectura constante de los sensores.

Núcleo 1: gestiona el servidor WebSocket y la comunicación inalámbrica.

Con esta división se logra mantener estable el servidor WebSocket con una comunicación fluida de los datos de los sensores y leer constantemente la memoria de todos los sensores evitando así que la memoria se sature.

Optimización del bus I2C

Para este sistema está definido el uso de varios sensores MPU9250, aunque este sensor tiene la limitación de poseer únicamente 2 direcciones para comunicación I2C (0x68 y 0x69) para solucionar este problema se aplica el uso de multiplexores TCA9548A que permiten usar hasta 8 canales independientes por cada uno.

Al aplicar el uso de multiplexores se soluciona el problema de las direcciones iguales en los sensores, pero incrementa el tiempo que hay entre la lectura de sensores ya que se requiere tiempo para cumplir con el flujo descrito en la Figura 4.3:



Figura 4.3 Proceso de lectura de los sensores

Como se mencionó anteriormente los sensores MPU9250 tienen una memoria interna, la cual almacena los datos obtenidos por los sensores, si esta memoria llega a saturarse es necesario vaciarla caso contrario presentará errores o puede provocar incluso que el sistema llegue a detenerse al esperar un dato que nunca va a llegar, con este comportamiento de la memoria se vio la necesidad de optimizar el flujo de lectura del bus I2C ya que al aplicar la solución de los multiplexores el tiempo de lectura de sensores incrementa y la memoria se satura, por lo que para evitar este problema se cambió la velocidad de lectura del bus I2C al modo rápido (400kHz) lo que permite aplicar un retardo utilizando el método `delayMicroseconds(3000)` al final de la lectura, reduciendo con esto el tiempo de estabilización y al tener una mayor frecuencia en el bus I2C el tiempo de lectura de cada sensor se reduce, por último se designa la lectura de sensores a un núcleo específico con lo que ya no se presentan retrasos al manejar el servidor WebSocket reduciendo el tiempo total para el proceso de lectura de los sensores.

4.3.7 Diseño del sistema en Unity

Una vez diseñado el prototipo con sensores para la obtención de rotación del cuerpo, se requiere visualizar, analizar e interactuar con estos datos en tiempo real, por lo que un motor 3D como Unity es adecuado para este fin.

Motor 3D y modelo humanoide

Para la representación del movimiento en tiempo real se utiliza un modelo 3D humanoide basado en la estructura Mecanism de Unity mediante la cual se puede aplicar un cuaternión de rotación directamente a los huesos del modelo mediante el uso de objetos tipo Transform.

Conversión de cuaterniones a rotación local

La ESP32 envía los datos mediante un JSON en la forma descrita previamente, el cuaternión recibido representa la orientación absoluta de cada sensor respecto a un sistema de referencia inercial por lo que para ser utilizados en el modelo 3D y en los cálculos primero deben ser procesados de la siguiente manera:

- Se ajusta el orden y signo de los componentes según la configuración específica de cada sensor considerando que Unity usa x, y, z, w como orden de los ejes del cuaternión.
- Se aplica una rotación de calibración como se describe en la ecuación (4.4) para ajustar la posición predeterminada del modelo 3D
- Si el sensor tiene un padre jerárquico (por ejemplo, el antebrazo depende del brazo) se calcula la rotación relativa con respecto a dicho padre aplicando la ecuación (4.5). En Unity se vería de la forma:
$$\text{Quaternion relativeQ} = \text{Quaternion.Inverse}(\text{parentQ}) * \text{childQ};$$
- Se aplica la rotación relativa como localRotation del objeto correspondiente manteniendo un movimiento coherente y ajustado al movimiento real, aplicando en Unity el método:
$$\text{segment.targetTransform.localRotation} = \text{relativeQ};$$

Sistema de offset y remapeo de ejes.

En el montaje físico de los sensores sobre el cuerpo los sensores tienen diferentes posiciones al encontrarse en la postura predeterminada por lo que cada uno debe tener configuraciones de ejes específicas según la orientación en el montaje. Esto se implementa mediante funciones creadas en unity:

- **Invertir componentes:** invierte los ejes del cuaternión de forma independiente cada uno según el montaje del sensor. Para esta función se aplica axisInversion: vector (1, -1, 1) que indica qué ejes invertir.

- **Remapear ejes:** permite el intercambio de ejes del cuaternión, por ejemplo, intercambiar los datos del eje x y z. Para esto se utiliza el método axisMapping que define qué eje del sensor tomar para cada componente (xFrom, yFrom, zFrom).

Diseño del sistema de evaluación RULA

La lógica del método RULA fue adaptada de su versión manual a un formato computacional en tiempo real. Para ello, se crearon tres módulos en Unity:

Evaluador Grupo A: Evalúa brazo, antebrazo, muñeca y giro de muñeca.

Evaluador Grupo B: Evalúa tronco, cuello y piernas.

Evaluador Final: Determina el puntaje total usando las tablas RULA y considera factores adicionales como fuerza o carga si son ingresados manualmente.

Con la rotación relativa calculada en cada articulación se puede transformar a ángulos de Euler en cada eje para su interpretación:

```
Vector3 euler = relativeQuaternion.eulerAngles;
```

```
euler.x = NormalizeAngle(euler.x);
```

```
euler.y = NormalizeAngle(euler.y);
```

```
euler.z = NormalizeAngle(euler.z);
```

De forma matemática el ángulo de rotación puede calcularse como se muestra en la ecuación (4.6).

$$\theta = 2 * \cos^{-1} w \quad (4.6)$$

Donde w es el componente escalar del cuaternión.

Criterios del método RULA implementados.

Las tablas de calificación del método RULA se implementan en scripts en Unity como reglas de decisión para cada grupo. Por ejemplo:

- Si el ángulo de flexión del codo esta entre 60° y 100° se asigna un puntaje de 1
- Si la muñeca presenta desviación radial mayor a 15° se asigna un puntaje de 2

Además, se considera también si el segmento esta rotado o flexionado respecto a su posición base.

- **Tronco:**

Flexión entre 0° - 20° : puntuación = 2

Flexión $>20^{\circ}$ y $\leq 60^{\circ}$: puntuación = 3

Flexión $>60^{\circ}$: puntuación = 4

Para realizar un mejor control sobre la evaluación se da la posibilidad de congelar el modelo en una pose, manteniendo los ángulos dados en ese momento y permitiendo el ingreso de datos correspondientes a factores externos como peso, repetitividad, etc.

4.4 ETAPA DE IMPLEMENTACIÓN

La etapa de implementación se enfocó en el desarrollo físico y lógico del sistema, partiendo de los diseños previamente establecidos. Durante esta fase, se integraron los sensores, multiplexores y microcontrolador, se configuró el sistema de adquisición y transmisión de datos, y se desarrolló la lógica en Unity para representar las posturas del trabajador y calcular automáticamente el puntaje RULA. A continuación, se presentan los principales procedimientos implementados, ilustrados mediante diagramas de flujo que facilitan la comprensión del funcionamiento general del sistema.

Todos los parámetros de diseño establecidos en la etapa de diseño sirvieron para desarrollar un esquema de conexión que incluye todos los componentes electrónicos del sistema, la Figura 4.4 presenta el esquema de conexiones aplicado a un solo sensor MPU9250 partiendo desde el sistema de alimentación, la conexión del multiplexor y la llegada del sensor al canal del multiplexor para el esquema completo se puede revisar el ANEXO A donde se detallan las conexiones entre todos los componentes del sistema.

Basado en el plano eléctrico del sistema se diseñó un esquema para PCB que facilite el montaje de los componentes en una sola placa, en la Figura 4.5 se presenta el esquema PCB en la que están montados los 2 multiplexores, el ESP32, un sistema de alimentación y carga y un regulador de voltaje para la alimentación de los sensores.

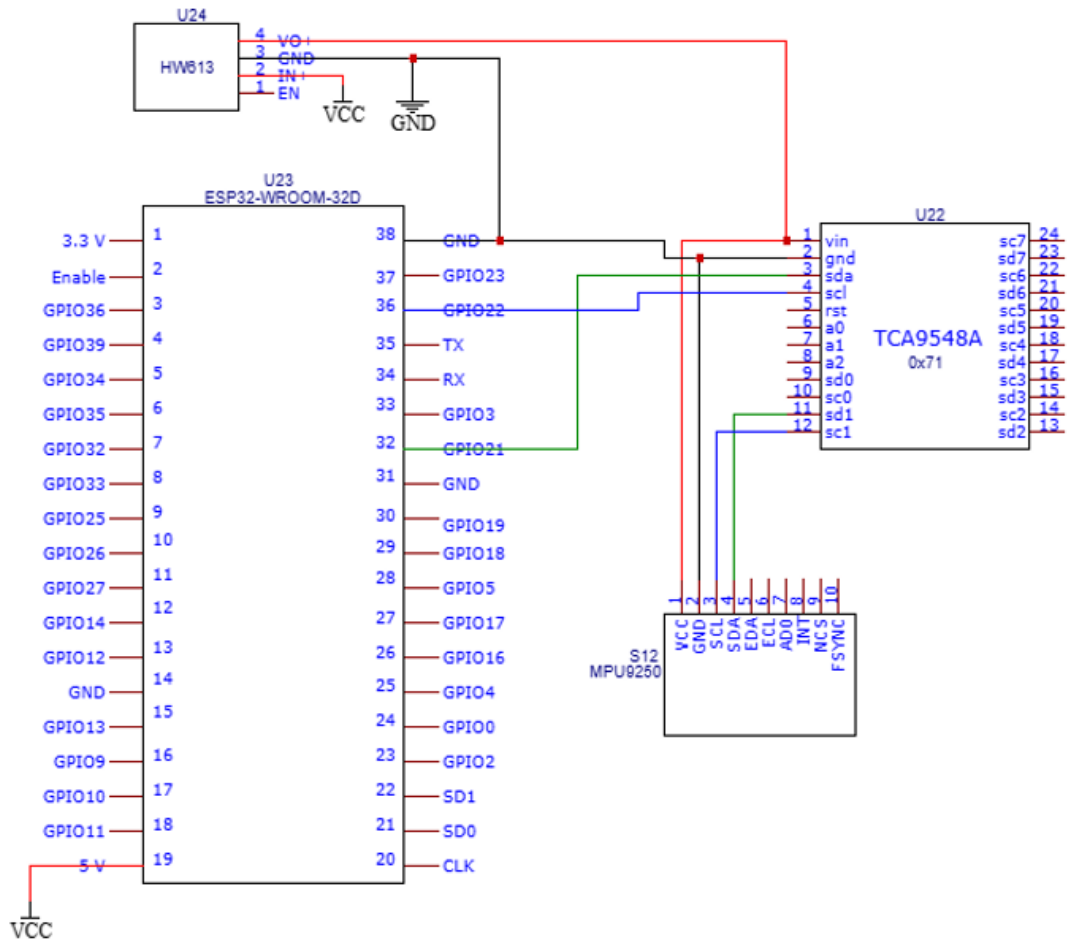


Figura 4.4 Esquema básico de conexión.

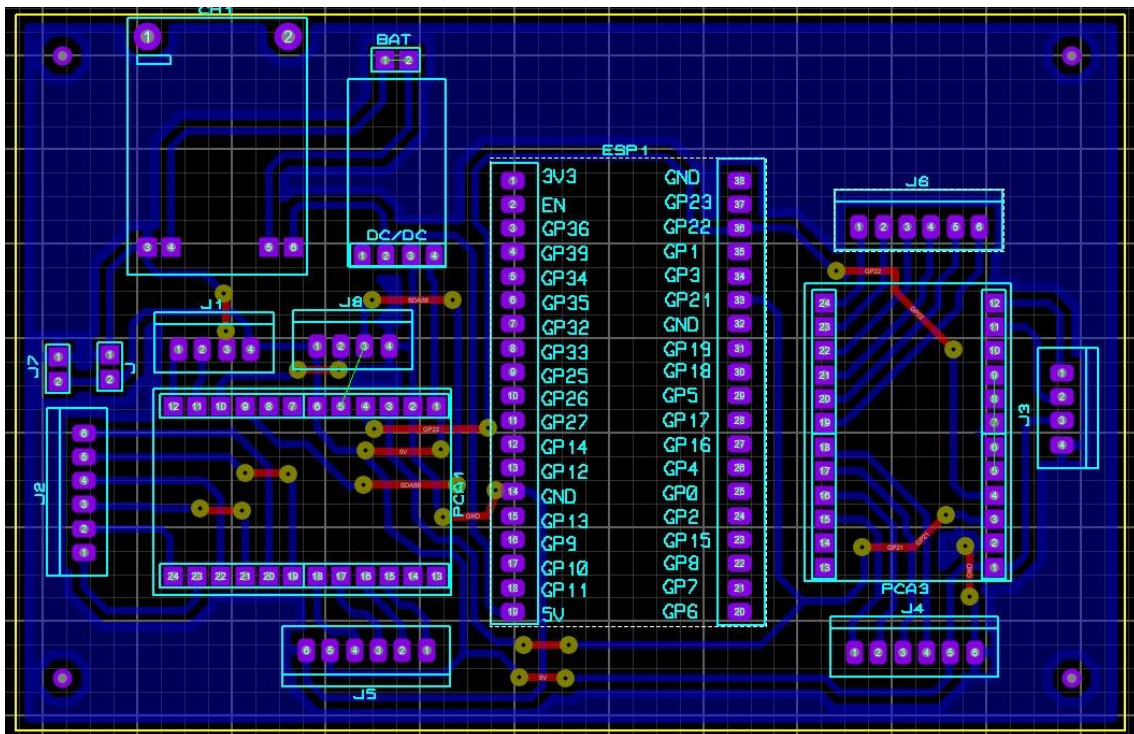


Figura 4.5 Esquema PCB

La implementación comenzó con el ensamblaje físico del sistema, realizando las conexiones entre los sensores, el multiplexor y la ESP32, en la Figura 4.6 se muestra el montaje físico de uno de los sensores.



Figura 4.6 Montaje de sensor

Una vez finalizado el montaje de todos los componentes se programó el microcontrolador para leer los datos de los sensores y enviarlos vía WebSocket hacia el cliente en Unity, para esto se utilizó el entorno de Arduino IDE, como se puede ver en la Figura 4.7 al iniciar el programa se realizan 3 funciones principales:

- Verificación de sensores inactivos al arranque.
- Detección y reinicio automático de FIFO saturada.
- Multiplexación por grupos (0, 1, 2) para minimizar la latencia en la lectura.

```
Output Serial Monitor x
Not connected. Select a board and a port to connect automatically.
...
WiFi conectado!
IP del ESP32: 172.16.114.181
Continuando en 5 segundos...
WebSocket iniciado en puerto 81
Sensor 0 inicializado
Sensor 1 inicializado
E (8321) i2c.master: I2C hardware NACK detected
E (8321) i2c.master: I2C transaction unexpected nack detected
E (8321) i2c.master: s_i2c_synchronous_transaction(924): I2C transaction failed
E (8327) i2c.master: i2c_master_multi_buffer_transmit(1186): I2C transaction failed
Sensor 2 inicializado
Sensor 5 inicializado
Sensor 6 inicializado
Sensor 7 inicializado
Sensor 8 inicializado
Sensor 9 inicializado
Sensor 10 inicializado
Grupo 2 | Sensor 5 | Q = [1.000, -0.017, 0.002, -0.010]
⚠ FIFO LLENA sensor 0 - reiniciando FIFO
⚠ FIFO LLENA sensor 1 - reiniciando FIFO
Grupo 0 | Sensor 2 | Q = [1.000, 0.009, -0.000, 0.004]
Grupo 0 | Sensor 6 | Q = [1.000, -0.010, 0.002, 0.026]
Grupo 1 | Sensor 7 | Q = [1.000, -0.017, -0.004, 0.008]
Grupo 1 | Sensor 8 | Q = [1.000, -0.020, 0.002, -0.018]
Grupo 1 | Sensor 9 | Q = [1.000, 0.012, -0.001, -0.001]
```

Figura 4.7 Inicio de sensores, detección de errores y lectura de datos.

4.4.1 Diagramas de flujo para la ESP32

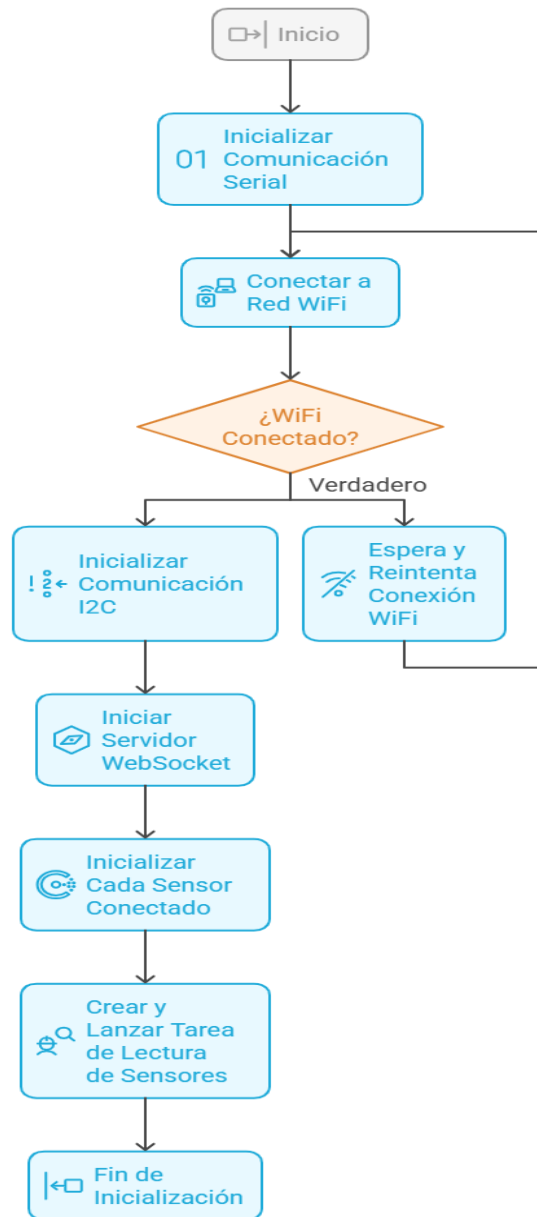


Figura 4.8 Diagrama de flujo de macroprocesos de la ESP32

La Figura 4.8 describe la secuencia principal que incluye la inicialización del sistema, el manejo de sensores y la transmisión de datos por WebSocket. Este diagrama es la base del funcionamiento del sistema ya que describe el proceso a seguir para tomar lectura de los sensores y enviarlos de forma inalámbrica.

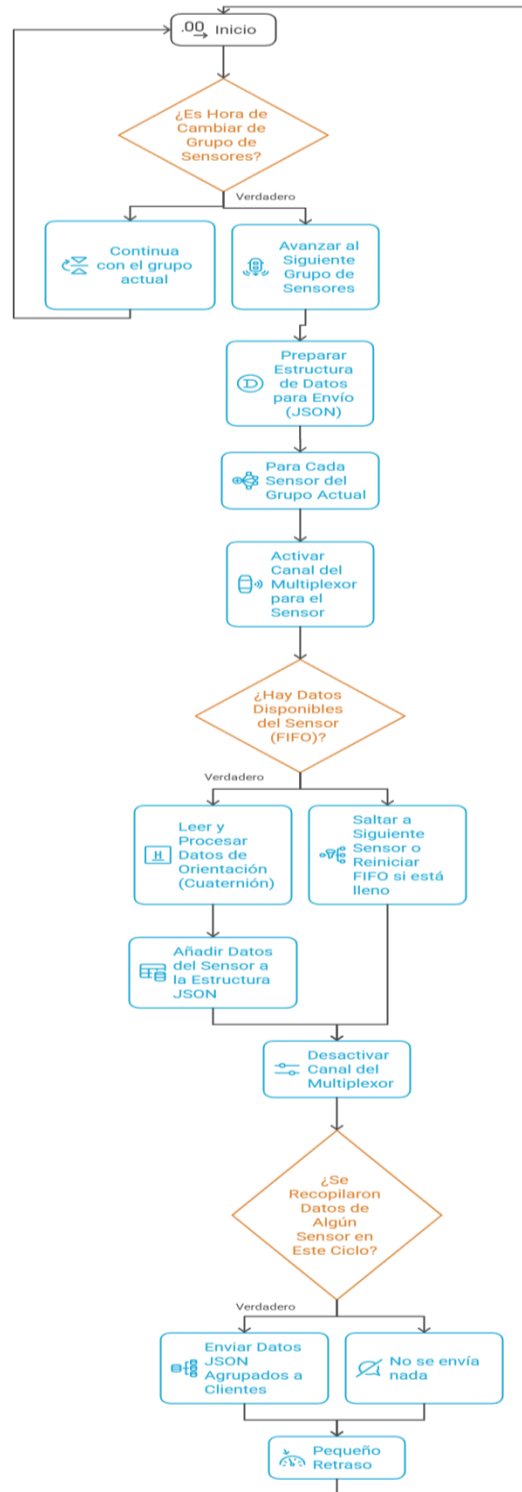


Figura 4.9 Diagrama de flujo del proceso de lectura y envío de datos de sensores.

En la Figura 4.9 se detalla cómo la ESP32 realiza la selección de canal I2C, lectura del sensor MPU9250 y construcción del mensaje JSON para envío



Figura 4.10 Diagrama de flujo del funcionamiento del servidor WebSocket.

La Figura 4.10 representa el ciclo de vida del servidor WebSocket desde su inicialización hasta la gestión de conexiones y envío de datos periódicos. Es importante saber que la secuencia descrita en este diagrama se ejecuta en un núcleo de la ESP32 independiente y en paralelo al código encargado de la lectura de sensores, esto para evitar retrasos en la lectura de sensores provocados por el envío de datos mediante WebSocket con el fin de evitar el llenado de la FIFO (First-In First-Out) y mantener una lectura fluida.

El resultado del uso de estos diagramas de flujo puede verse en el código presentado en el ANEXO B.

4.4.2 Diagramas de flujo para Unity



Figura 4.11 Diagrama de flujo de conexión y recepción de datos por WebSocket.

En la Figura 4.11 se muestra cómo se establece la conexión con la ESP32, recepción de mensajes JSON y su deserialización, Estos datos crudos son los que se utilizan posteriormente calcular ángulos relativos y usarlos para el movimiento del modelo 3D y el cálculo de puntuación RULA.

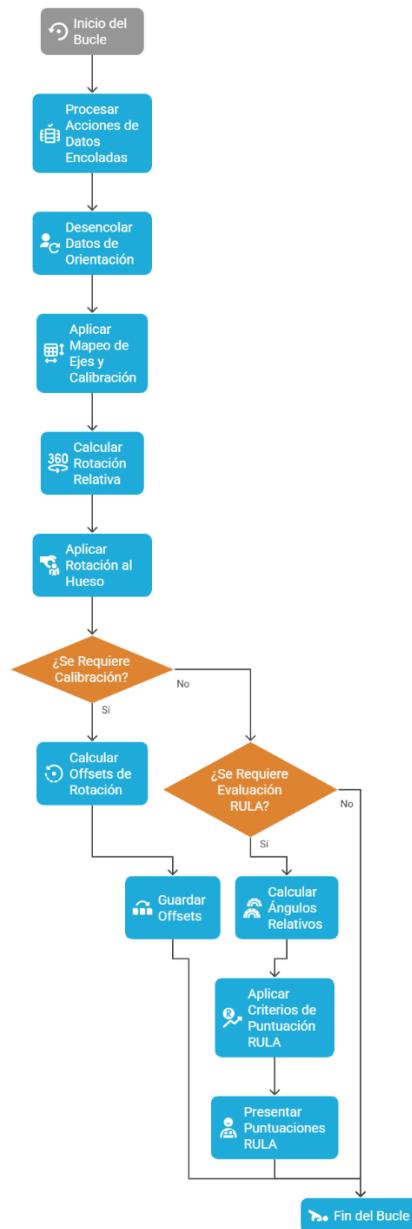


Figura 4.12 Diagrama de flujo de actualización del modelo 3D y cálculo de puntaje RULA.

Explica el flujo interno de Unity para aplicar rotaciones a cada parte del cuerpo del modelo 3D, seguido del proceso de evaluación automática según los módulos A, B y final del método RULA. Este diagrama es el corazón del sistema ya que trata los datos crudos y los transforma a ángulos usables para aplicar las tablas del método RULA.

4.5 ETAPA DE PRUEBAS

Con el sistema implementado, se llevó a cabo una etapa de pruebas enfocada en verificar el correcto funcionamiento de cada componente y evaluar el desempeño general del prototipo. Las pruebas se dividieron en cuatro áreas principales: estabilidad de sensores, comunicación, visualización 3D y precisión en la evaluación RULA.

- **Pruebas de funcionamiento de sensores**

Se conectaron los 13 sensores MPU9250 distribuidos en el cuerpo, utilizando dos multiplexores I2C TCA9548A controlados por la ESP32. Durante las pruebas iniciales se verificó:

La detección automática de sensores conectados.

El correcto funcionamiento del sistema cuando hay sensores deshabilitados o ausentes.

La calidad de los datos de orientación obtenidos desde la librería FastIMU.

Se observó que los sensores con magnetómetro integrado ofrecían buena estabilidad cuando se utilizaba la librería SparkFunMPU9250-DMP, sin embargo, en la migración hacia librerías más flexibles como FastIMU o MPU9250_asukiaaa, se detectó la presencia de ruido en los cuaterniones cuando el sensor estaba en reposo. Esto obligó a realizar pruebas comparativas para identificar el algoritmo de filtro más estable (Madgwick vs Mahony), concluyendo que FastIMU con configuración Mahony ofrecía un compromiso aceptable entre estabilidad y tiempo de respuesta, sin embargo el no utilizar el DMP integrado en el sensor MPU9250 agregaba una alta carga para la ESP32 lo que provocó lecturas erróneas al utilizar varios sensores, estos errores incluían DRIFT, ruido, posición de referencia incorrecta, por lo que finalmente se trabajó con la librería de SparkFun, implementando el uso de DMP y con un flujo de trabajo dividido en varios grupos y usando ambos núcleos de la ESP32 para un trabajo en paralelo en lectura y envío de datos.

- **Pruebas de comunicación y latencia**

Se realizaron pruebas en la ESP32 para evaluar la eficiencia del sistema de lectura y envío de datos. Se optimizó la función `selectMuxChannel()` para reducir su latencia, logrando un comportamiento fluido incluso al trabajar con más de 10 sensores activos.

Durante las pruebas se monitoreó la latencia desde la captura de datos en la ESP32 hasta la visualización en Unity, determinando un retardo promedio menor a 100 ms, suficiente para representar movimientos en tiempo casi real. El sistema WebSocket demostró ser estable y soportar la carga de transmisión continua de datos JSON sin pérdida significativa.

- **Pruebas en Unity: movimiento y evaluación**

En la interfaz de Unity se verificó:

- La correcta asignación de sensores a segmentos del cuerpo humanoide.
- La aplicación de cuaterniones con offset inicial calibrado.
- El remapeo e inversión de ejes de acuerdo con la posición física del sensor.

Las pruebas demostraron que el modelo 3D respondía de manera precisa a los movimientos realizados por el usuario. Se simularon varias posturas relevantes para la evaluación RULA, y se observó una actualización automática y coherente del puntaje en cada grupo (A y B), así como en el resultado final.

- **Robustez del sistema ante fallos**

Se comprobó el comportamiento del sistema ante:

- Desconexión de uno o más sensores.
- Datos incompletos o fuera de rango.
- Reinicios parciales del microcontrolador.

Gracias al diseño modular, el sistema siguió funcionando y evaluando con los sensores disponibles, asignando valores por defecto a los segmentos sin datos. Esto valida la escalabilidad y tolerancia a fallos del diseño propuesto.

Observaciones generales

- El sistema de calibración por botón en Unity resultó clave para la alineación inicial del modelo.

- Se identificaron pequeñas diferencias entre la evaluación automática y una evaluación manual de referencia, atribuibles a menor ruido en los datos de ángulos y simplificaciones del método.

En conjunto, los resultados de la etapa de pruebas indican que el sistema cumple con los requerimientos funcionales y no funcionales definidos en la etapa de análisis, y es capaz de evaluar posturas laborales en tiempo real de forma no intrusiva, confiable y modular. y clara cómo interactúan los diferentes módulos que componen el sistema, y sirven como referencia para su mantenimiento, mejora o ampliación futura.

4.6 ETAPA DE AJUSTES

Una vez finalizada la etapa de pruebas, se procedió con los ajustes necesarios para mejorar la estabilidad, precisión y flexibilidad del sistema. Esta etapa fue clave para convertir el prototipo funcional en una herramienta robusta y adaptable, lista para ser utilizada en distintos contextos laborales y educativos. A continuación, se describen los ajustes realizados, categorizados por componente y funcionalidad.

- **Ajustes en firmware y adquisición de datos**

Durante las pruebas se detectaron inconsistencias en la lectura de datos cuando múltiples sensores estaban activos simultáneamente. Para solucionarlo:

- Se optimizó la función de selección de canales `selectMuxChannel()` para reducir su tiempo de ejecución.
- Se implementó una estructura de control para que los sensores deshabilitados sean automáticamente omitidos, evitando intentos de lectura innecesarios.
- Se ajustó el tiempo de espera entre lecturas y el manejo del bus I2C para evitar colisiones o bloqueos.

Además, se añadió un sistema de identificación dinámica de sensores al inicio del programa, lo que permite que la ESP32 continúe operando incluso si uno o varios sensores no están conectados.

- **Ajustes en la estructura de datos**

Se estandarizó el formato JSON para que siempre incluya únicamente los sensores activos, con claves identificadas como "q0", "q1", etc. Esto permitió reducir el tamaño del paquete transmitido, mejorando la eficiencia del sistema.

También se incluyó una validación de datos antes del envío para garantizar que los cuaterniones estén correctamente formateados y dentro de rangos esperados.

- **Ajustes en Unity**

En la interfaz Unity se realizaron múltiples ajustes basados en pruebas de usabilidad y consistencia:

- Se refinó el sistema de calibración, permitiendo guardar el offset inicial como referencia para futuras sesiones.
- Se amplió el sistema de configuración en el Inspector para permitir una reasignación rápida de sensores a partes del cuerpo.
- Se implementó un sistema de remapeo de ejes configurable, adaptándose a diferentes orientaciones físicas de montaje.

En cuanto a la evaluación RULA, se modularizó el código de cada evaluador (Grupo A, Grupo B y Final), permitiendo que el sistema siga funcionando incluso si alguno de los módulos no puede calcular un valor (por falta de sensores o error de datos).

Ajustes en la interfaz y usabilidad

- Se agregó un botón de calibración accesible desde la interfaz gráfica.
- Se eliminaron funciones de edición manual de datos que estaban presentes en versiones anteriores utilizadas para pruebas.
- Se ajustaron los mensajes de error y alertas para facilitar el diagnóstico durante el uso del sistema.

Ajustes en estabilidad y tolerancia a fallos

El sistema fue sometido a situaciones intencionales de error (desconexión de sensores, reinicio del microcontrolador, pérdida de red). Se comprobó que:

- La ESP32 reconecta automáticamente con la red y con Unity tras un reinicio.
- El servidor WebSocket retoma la conexión sin necesidad de reiniciar manualmente.

- Unity continúa evaluando con los sensores disponibles y reemplaza con valores neutros los datos ausentes.

Resultados de los ajustes

Con los ajustes aplicados, se logró:

- Una reducción significativa en los errores de lectura y transmisión.
- Un modelo 3D más preciso, fluido y coherente con las posturas reales.
- Mayor flexibilidad del sistema para adaptarse a configuraciones variables.

Esta etapa de ajustes consolidó la propuesta, transformándola en un sistema de evaluación ergonómica automatizado, eficiente, tolerante a fallos y aplicable en diversos entornos reales o de investigación. y clara cómo interactúan los diferentes módulos que componen el sistema, y sirven como referencia para su mantenimiento, mejora o ampliación futura.

La Figura 4.13 corresponde al prototipo desarrollado montado sobre un usuario en condiciones normales de uso, esta es la propuesta final en la que se aplicaron todas las correcciones y ajustes descritos en esta etapa y es el resultado de las pruebas hechas.



Figura 4.13 Dispositivo montado.

5 ANALISIS Y DISCUSIÓN DE LOS RESULTADOS

5.1 EVALUACIÓN DEL FUNCIONAMIENTO DEL PROTOTIPO.

Para validar el funcionamiento del prototipo se lo sometió a pruebas en condiciones controladas utilizando posturas específicas para evaluar el comportamiento de todo el sistema. Estas pruebas se enfocaron en una articulación específica y los resultados obtenidos se compararon con valores de referencia conocidos. La visualización de los ángulos se realizó mediante la interfaz gráfica desarrollada en Unity, que presenta los ángulos relativos entre sensores en formato de ángulos de Euler para una interpretación más intuitiva.

5.1.1 Lectura de sensores y estabilidad

El primer criterio a evaluar fueron los filtros o algoritmos de orientación propuestos en la etapa de diseño para lo cual se realizaron múltiples pruebas utilizando un único sensor y se midieron los resultados, para estas pruebas se estableció que se realizarían 15 lecturas en las que se buscaba medir el error de orientación, la fluctuación presente durante el reposo y el retardo que presentaba.

Los filtros que se evaluaron fueron Madgwick, Mahony y el DMP del sensor, para la prueba de estabilidad se mantuvo al sensor en reposo durante un periodo de 30 segundos, para el error promedio en orientación se aplicaron los mismos movimientos en el sensor para cada uno de los filtros y para el retardo se escribía en puerto serie el intervalo entre lecturas.

Tabla 5.1 Comparativa de filtros.

Filtro	Error promedio en orientación (°)	Fluctuación en reposo (°)	Retardo estimado (ms)	Requiere calibración previa	Comentarios finales
DMP	6.45	±0.5	5 – 10	No	Alta estabilidad, uso eficiente del hardware.
Madgwick	8.90	±1.2	10 – 15	Sí (offsets)	Preciso, pero sensible a ruido e interferencia.
Mahony	10.75	±1.8	12 – 18	Sí	Inestable sin buen ajuste de parámetros.

Como se aprecia en la Tabla 5.1, el DMP obtuvo el mejor desempeño general, destacando por su baja fluctuación en reposo, respuesta inmediata y precisión aceptable frente a movimientos controlados. Esto se debe a que el DMP es un coprocesador dedicado incluido en el propio chip MPU9250, lo que permite liberar a la ESP32 del cálculo de filtros complejos y asegura una lectura optimizada directamente desde la FIFO.

En contraste, tanto Madgwick como Mahony requieren procesamiento en la ESP32, lo que incrementa el uso de CPU, agrega retardo, y dificulta la implementación de múltiples sensores simultáneamente, especialmente bajo un esquema multitarea.

Además, el DMP no necesita calibraciones previas para ángulo de giro ni ajustes de offset, lo que simplifica su integración en un sistema modular como el diseñado para esta propuesta.

Por estos motivos, el DMP fue seleccionado como núcleo del sistema de orientación, y se implementó en el prototipo final de forma estable con hasta 13 sensores conectados simultáneamente mediante multiplexores I2C.

Pruebas de lectura de postura del prototipo.

Para la prueba de lectura de sensores se aplicó una postura estática y diferentes movimientos en diferentes partes del cuerpo, con eso se logró comprobar la respuesta de los sensores y el tiempo que tarda en aplicarse el movimiento en el modelo 3D.

- **Prueba 1:** Postura firme (Calibración)

Esta postura se utilizó como referencia para establecer la posición neutra del cuerpo. El modelo estaba en posición recta, con la cabeza mirando al frente y sin movimientos; la Figura 5.2 muestra la postura real de prueba mientras que la Figura 5.3 muestra la postura en Unity y los ángulos leídos por los sensores; para la visualización se tienen datos de rotación de cada eje, estos valores representan el ángulo respecto a cada eje y están acorde a la Figura 5.1 siendo así que el primer dato corresponde a la rotación en el eje X o α , el segundo dato corresponde a la rotación en el eje Y o β y por último el tercer dato corresponde a la rotación presente en el eje Z o θ :

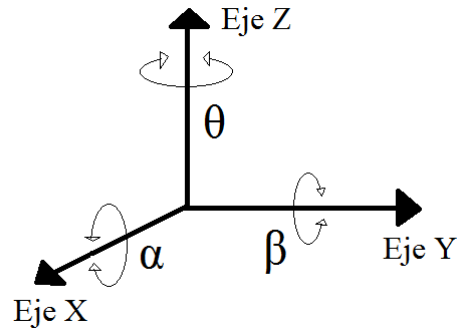


Figura 5.1 Ejes de rotación aplicados en el modelo.

Articulación evaluada: Cuello.

Ángulos mostrados por Unity: (0.47°, 0.35°, -0.35°)

Ángulos esperados: (0°, 0°, 0°)



Figura 5.2 Postura firme



Figura 5.3 Postura firme en Unity.

El sistema mostró una alta estabilidad en posición de reposo, la Tabla 5.2 muestra la comparación de los resultados, con variaciones menores a $\pm 0.5^\circ$. Este nivel de precisión es suficiente para establecer una referencia base en la evaluación automática.

Tabla 5.2 Comparación de resultados en postura firme.

Eje	Ángulo esperado (°)	Ángulo Unity (°)	Error Absoluto (°)	Error porcentual (%)
X	0.00	0.47	0.47	-
Y	0.00	0.35	0.35	-
Z	0.00	-0.35	-0.35	-

- **Prueba 2:** Posición en T.

El sujeto adoptó la postura en T (brazos extendidos horizontalmente a los lados), para las comparaciones se muestra la posición real en la Figura 5.4 y la Figura 5.5 presenta las lecturas y movimiento en Unity.

Articulación evaluada: Brazo derecho.

Ángulos mostrados por Unity: (77.62°, -16.93°, -4.84°)

Ángulos esperados: (90°, 0°, 0°)



Figura 5.4 Posición en T

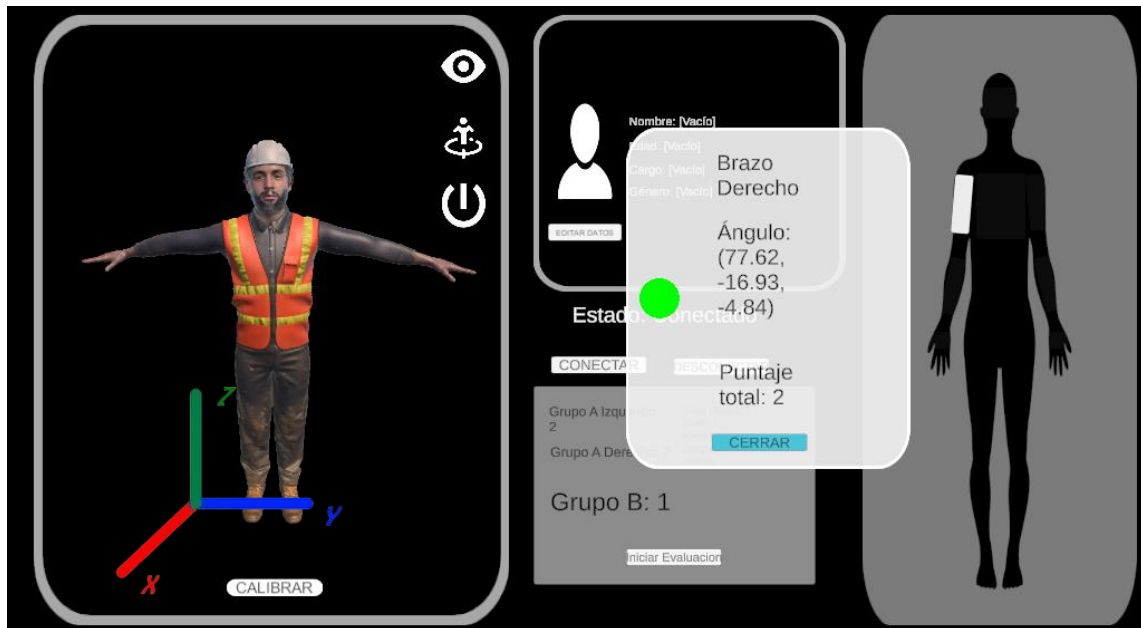


Figura 5.5 Posición en T en Unity.

En la Tabla 5.3 se muestra la comparación de los resultados leídos y los esperados, al analizarlos se logra deducir que, aunque el eje principal (X) se aproxima con un error de 13.75 %, se observa interferencia de rotaciones en los ejes Y y Z. Esto puede deberse a

pequeñas inclinaciones en la colocación del sensor o una calibración incompleta del modelo. Sin embargo, el ángulo predominante mantiene coherencia con la postura.

Tabla 5.3 Comparación de resultados en posición en T.

Eje	Ángulo esperado (°)	Ángulo Unity (°)	Error Absoluto (°)	Error porcentual (%)
X	90.00	77.62	12.38	13.75
Y	0.00	-16.93	-16.93	-
Z	0.00	-4.84	-4.84	-

- **Prueba 3: Brazo levantado.**

El participante levantó el antebrazo izquierdo con un ángulo definido en el plano sagital, la Figura 5.6 muestra el movimiento del brazo en el entorno real y la Figura 5.7 presenta su movimiento en Unity.

Articulación evaluada: Antebrazo izquierdo.

Ángulos mostrados por Unity: (3.49 °, 18.70°, -43.31°)

Ángulos esperados: (0°, 10°, -45°)



Figura 5.6 Brazo levantado.

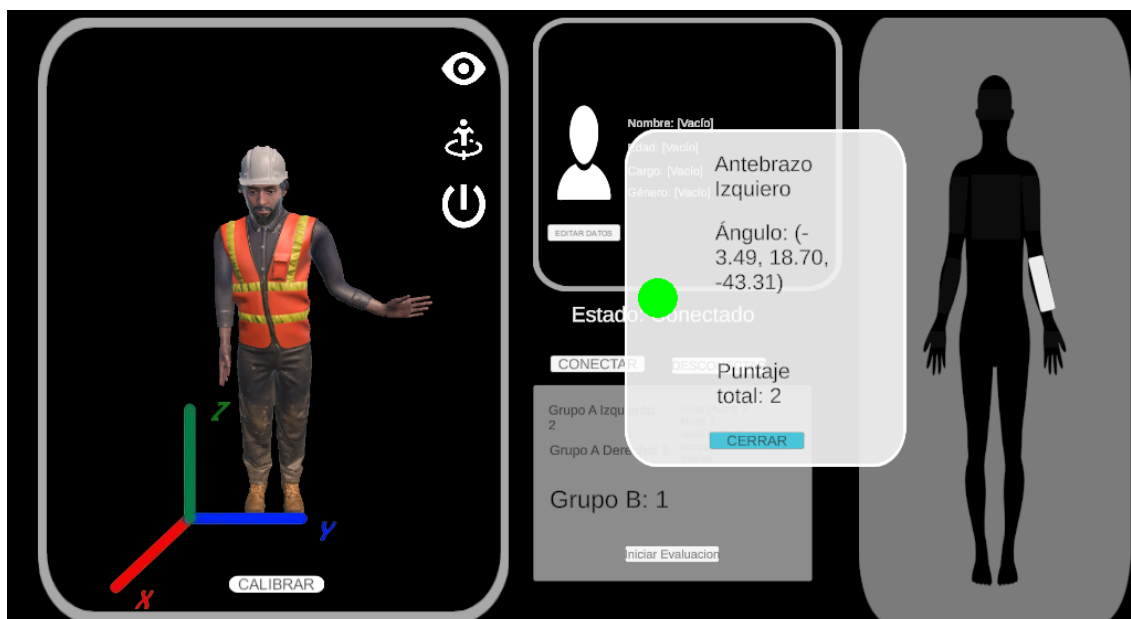


Figura 5.7 Brazo levantado en Unity.

Para el análisis de los resultados dados en esta prueba se utilizaron los datos mostrados en la Tabla 5.4 con estos datos se mostró que el sistema identificó correctamente la flexión principal en el eje Z. El error en el eje Y es alto debido a una pequeña torsión del brazo, no prevista en el ángulo real. Esta prueba demuestra que el sistema tiene buena sensibilidad, aunque la interpretación de movimientos compuestos requiere refinamiento.

Tabla 5.4 Comparación de resultados con brazo levantado.

Eje	Ángulo esperado (°)	Ángulo Unity (°)	Error Absoluto (°)	Error porcentual (%)
X	0.00	3.49	3.49	-
Y	10.00	18.70	8.70	87
Z	-45.00	-43.31	1.69	3.76

- **Prueba 4:** Movimiento del cuello.

El usuario giró la cabeza hacia la izquierda y giro la barbilla, la Figura 5.8 describe el movimiento realizado y la Figura 5.9 muestra el movimiento correspondiente a Unity.

Articulación evaluada: Cuello.

Ángulos mostrados por Unity: (-33.51 °, -17.47°, -30.97°)

Ángulos esperados: (-35°, -20°, -25°)



Figura 5.8 Movimiento de cuello.

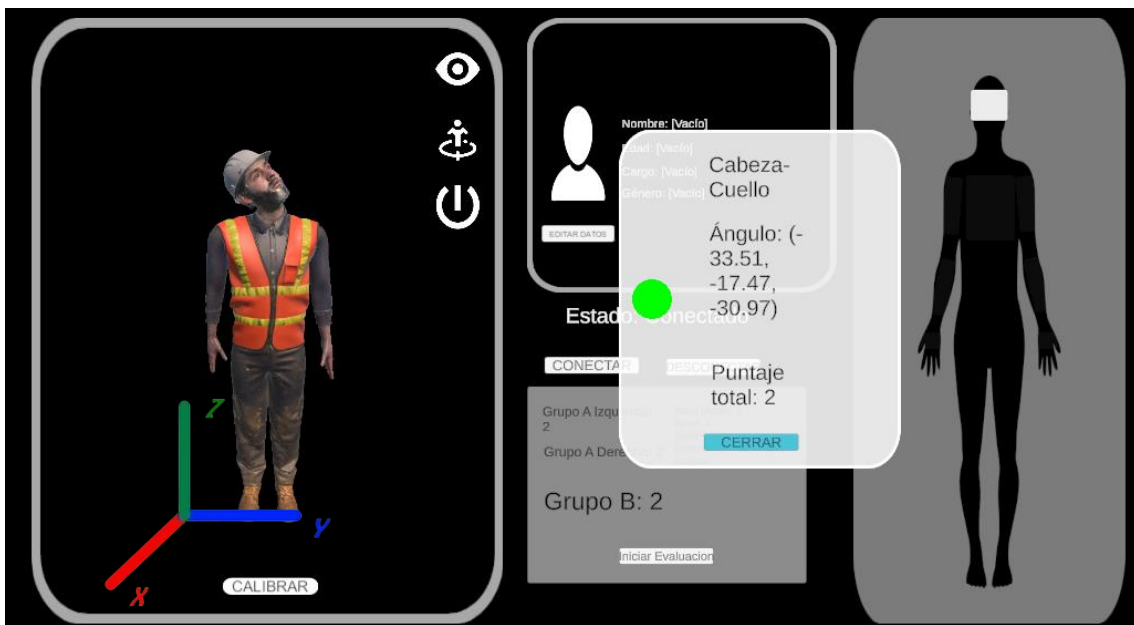


Figura 5.9 Movimiento de cuello en Unity.

Al comparar los resultados mostrados en la Tabla 5.5 se encontró que el sistema muestra buena concordancia en los tres ejes, con errores menores al 25%. La mayor discrepancia aparece en el eje Z, posiblemente debido a interpolaciones del modelo de Unity. Aun así, la orientación relativa es representativa del movimiento realizado.

Tabla 5.5 Comparación de resultados para el movimiento del cuello.

Eje	Ángulo esperado (°)	Ángulo Unity (°)	Error Absoluto (°)	Error porcentual (%)
X	-35.00	-33.51	1.49	4.26
Y	-20.00	-17.47	2.53	12.65
Z	-25.00	-30.97	5.87	23.88

5.2 DISCUSIÓN GENERAL DE RESULTADOS.

Los resultados obtenidos muestran que el sistema logra capturar y representar las posturas del cuerpo humano en tiempo real. El uso de cuaterniones elimina los errores de gimbal lock, y la interpretación de ángulos relativos permite una evaluación eficaz con el método RULA. Con lo descrito en la Tabla 5.6 Si bien existen errores en componentes menores de movimiento, el eje dominante de cada postura fue correctamente identificado lo que representa una menor precisión, pero permite realizar una evaluación automatizada aplicando el método RULA.

Tabla 5.6 Error en las pruebas.

Nº	Prueba	Parte del cuerpo	Ángulo real (°)	Ángulo en Unity (°)	Error absoluto (°)	Error porcentual (%)
1	Postura firme	Cuello	(0, 0, 0) (T-pose de referencia)	(0.47, 0.35, -0.35)	(0.47, 0.35, 0.35)	(N/A: base de calibración)
2	Posición T	Brazo derecho	(90, 0, 0) (ideal en eje X)	(77.62, -16.93, -4.84)	(12.38, 16.93, 4.84)	(13.8%, 18.8%, 5.4%)
3	Brazo levantado	Antebrazo izquierdo	(0, 10, -45)	(3.49, 18.70, -43.31)	(3.49, 8.70, 1.69)	(0%, 87.0%, 3.8%)
4	Movimiento de cuello	Cuello	(-35, -20, -25)	(-33.51, -17.47, -30.97)	(1.49, 2.53, 5.97)	(4.3%, 12.7%, 23.9%)

Cálculo de error general.

Utilizando los datos de la Tabla 5.6 se puede calcular un error promedio debido a que los datos presentados representan el porcentaje de error medido en los resultados entregados por el prototipo frente a los datos reales conocidos, al analizar 3 posturas principales y realizar el estudio en el sensor de mayor impacto en cada postura se obtuvo 9 datos que pueden representar el comportamiento del prototipo y el rango de error en el que opera :

Suma de errores absolutos (°):

$$\text{Brazo derecho: } 12.38 + 16.96 + 4.84 = 34.15^\circ$$

$$\text{Antebrazo izquierdo: } 3.49 + 8.70 + 1.69 = 13.88^\circ$$

$$\text{Cuello (movimiento): } 1.49 + 2.53 + 5.97 = 10.00^\circ$$

$$\text{Total, general de error en grados: } 34.15 + 13.88 + 10.00 = 58.03^\circ$$

$$\text{Error absoluto promedio (°): } \frac{58.03}{9} = 6.45^\circ$$

Suma de errores porcentuales (solo donde hay datos reales):

$$\text{Brazo derecho: } 13.8 + 18.8 + 5.4 = 38.0\%$$

$$\text{Antebrazo izquierdo: } 0 + 87.0 + 3.8 = 90.8\%$$

$$\text{Cuello (movimiento): } 4.3 + 12.7 + 23.9 = 40.9\%$$

$$\text{Total: } 38.0 + 90.8 + 40.9 = 169.7\%$$

$$\text{Error porcentual promedio general: } \frac{169.7}{9} = 18.85\%$$

En conjunto, las pruebas demuestran que el sistema es capaz de capturar correctamente la orientación de las extremidades humanas y transformarla en una visualización coherente dentro de Unity, permitiendo así la aplicación automatizada del método RULA.

No obstante, se identifica como principal fuente de error la posición relativa de los sensores al momento de la calibración inicial. Un mal alineamiento genera ángulos relativos incorrectos, incluso si el sensor en sí funciona correctamente. Por tanto, se concluye que la precisión del sistema depende más de la calidad del proceso de montaje y calibración que de los componentes utilizados.

5.3 ANÁLISIS ECONÓMICO

El sistema propuesto se diseñó con componentes disponibles comercialmente, buscando mantener un balance entre costo, rendimiento y disponibilidad. La Tabla 5.7 presenta un resumen de los costos asociados al prototipo.

Tabla 5.7 Costo componentes para fabricación del prototipo.

Componente	Cantidad	Costo unitario (USD)	Costo total (USD)
ESP32 DevKit	1	13.00	13.00
Sensor MPU9250	13	12.00	156.00
Multiplexor TCA9548A	2	3.00	6.00
Batería Li-Po	1	15.00	15.00
Regulador de voltaje	1	4.00	4.00
Modulo de carga	1	1.00	1.00
Cables, conectores, PCB	—	—	15.00
Arnés o estructura física	1	20.00	20.00
Costo desarrollo	80 horas	4.50/hora	360.00
TOTAL	—		590.00

Con un presupuesto total estimado en 590 USD, el sistema demuestra ser una alternativa de bajo costo frente a soluciones comerciales de captura de movimiento, cuyo valor puede exceder fácilmente el costo directo del sistema propuesto. Esta diferencia económica sustancial facilita la replicación del proyecto en empresas o instituciones educativas con recursos limitados ya que una vez construido el prototipo el costo de desarrollo se reduce, siendo únicamente necesario el tiempo que se utilice para la evaluación.

Una ventaja clave del sistema propuesto es la reducción significativa en el tiempo necesario para realizar evaluaciones. La Tabla 5.8 presenta una estimación del tiempo requerido para ambos métodos, el tiempo estimado para el método manual está tomando como referencia experiencias prácticas documentadas por organismos de salud ocupacional.

Tabla 5.8 Tiempo necesario para aplicar la evaluación manual y automática.

Proceso	Método Manual (min)	Sistema Automatizado (min)
Observación de postura	5 – 10	—
Identificación de segmentos y ángulos	10 – 20	—
Aplicación del método RULA	10 – 15	—
Colocación del prototipo	—	3 – 5
Calibración del sistema	—	15-20
Conexión al sistema de visualización	—	2 – 3
Toma de datos	—	5 – 6
Total estimado	25 – 45 min	25 – 34 min

Al comparar directamente los valores se percibe un tiempo mínimo similar tanto en la evaluación manual como en la evaluación automática, sin embargo, en la evaluación manual el proceso completo debe repetirse por cada evaluación mientras que en la evaluación automática una vez colocados los sensores y calibrados el tiempo de evaluación se reduce notablemente ya que se pueden tomar datos en cualquier momento.

Para obtener una mejor referencia en la optimización presentada por el sistema propuesto se realizó un cálculo de la reducción de tiempo promedio:

Tiempo promedio:

$$\text{Manual: } \frac{25+45}{2} = 35 \text{ minutos}$$

$$\text{Automática: } \frac{25+34}{2} = 29.5 \text{ minutos}$$

Reducción promedio:

$$\text{Reducción (\%)} = \frac{35-29.5}{35} * 100 = 15.71\%$$

Con esto se logra determinar que el sistema ofrece una reducción promedio del tiempo utilizado para una evaluación del 15.71%, este resultado puede variar dependiendo de factores externos como la experiencia del evaluador, la cantidad de personal disponible para realizar la evaluación o la complejidad de las evaluaciones, sin embargo, ofrece una referencia clara de la mejora del sistema.

6 CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- A partir de la revisión bibliográfica y el análisis técnico, se establecieron los criterios para la selección de los componentes que conforman el sistema automatizado integrando 13 sensores inerciales MPU9250 conectados a un microcontrolador ESP32 mediante multiplexores TCA9548A para la gestión de la comunicación. Se determinaron las zonas específicas donde se ubican los sensores según los lineamientos del método RULA incluyendo extremidades superiores, inferiores, tronco y cabeza.
- Asimismo, se definió un flujo de trabajo estructurado que inicia con la calibración del sistema, luego con la captura simultánea de datos de los sensores, la transmisión a través de WebSocket, la conversión y remapeo de los datos en Unity y finalmente la interpretación de los ángulos críticos para obtención del puntaje RULA. Esta implementación permite que el evaluador observe las posturas desde una ubicación distinta a la del trabajador, reduciendo el sesgo de comportamiento que puede producirse por observación directa y facilitando la evaluación sin interferencias.
- Las pruebas realizadas en entornos controlados demostraron que el sistema puede reconocer posturas y calcular ángulos relativos con un margen de error promedio del 18.85% y una desviación angular media de 6,45°, siendo la principal causa de error la posición inicial de los sensores durante la calibración.
- Aunque el sistema no sustituye completamente al método manual por sus limitaciones de precisión, constituye una alternativa viable para evaluaciones rápidas, periódicas y con menor intervención humana. Además, presenta un beneficio de optimización de recursos al reducir en un 15,71 % el tiempo de evaluación respecto al método manual, lo que representa un ahorro significativo en contextos donde se requieran mediciones frecuentes.

6.2 RECOMENDACIONES

- Para futuros trabajos se debería incluir información sobre las condiciones laborales del campo específico donde se aplicará el estudio, esto permitirá integrar el sistema automatizado al campo y mejorar su aplicabilidad en el ámbito local e industrial.
- Para reducir notablemente el error presente en el sistema automatizado se recomienda desarrollar una estructura anatómicamente guiada que permita posicionar de manera precisa los sensores en todo el cuerpo, con esta solución las lecturas serán más cercanas a la posición real y mejorará los resultados generales del sistema.
- Con el objetivo de asegurar la validez del sistema como herramienta ergonómica confiable, se recomienda realizar un estudio comparativo controlado a cargo de expertos en ergonomía, este estudio debe contrastar los resultados generados por expertos mediante observación directa y los obtenidos mediante el sistema automatizado, con esto se logra cuantificar el nivel de confiabilidad del sistema y tener un criterio de mejora en los algoritmos de evaluación.

7 REFERENCIAS

- [1] J. González, E. Quintero, F. Garci, A. García, Y. Sáez, y E. Collado, «Data science-based tool to reduce measurement errors in atmospheric monitoring systems; [Herramienta basada en ciencia de datos para reducir errores en las mediciones en sistemas de monitoreo atmosférico]», en *Proceedings of the LACCEI international Multi-conference for Engineering, Education and Technology*, Latin American and Caribbean Consortium of Engineering Institutions, 2024. doi: 10.18687/LACCEI2024.1.1.1185.
- [2] F. Contreras Argumedo, A. Villanueva Montellano, S. Tilvaldyev, y D. Cornejo Monroy, «Mejora de la productividad, calidad y ergonomía de una operación de ensamble por medio de la automatización», *Cultura Científica y Tecnológica*, vol. 0, n.º 58, dic. 2016, [En línea]. Disponible en: <https://erevistas.uacj.mx/ojs/index.php/culcyt/article/view/1429>
- [3] A. Senfelds, «Analysis of motion modelling approaches for industrial robot applications», en *2019 IEEE 7th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, IEEE, nov. 2019, pp. 1-4. doi: 10.1109/AIEEE48629.2019.8977112.
- [4] J. Cuadrado, F. Michaud, U. Lugrís, y M. Pérez Soto, «Using Accelerometer Data to Tune the Parameters of an Extended Kalman Filter for Optical Motion Capture: Preliminary Application to Gait Analysis», *Sensors 2021, Vol. 21, Page 427*, vol. 21, n.º 2, p. 427, ene. 2021, doi: 10.3390/S21020427.
- [5] S. Y. Chuang, N. Sahoo, H. W. Lin, y Y. H. Chang, «Predictive Maintenance with Sensor Data Analytics on a Raspberry Pi-Based Experimental Platform», *Sensors 2019, Vol. 19, Page 3884*, vol. 19, n.º 18, p. 3884, sep. 2019, doi: 10.3390/S19183884.
- [6] T. Sadi, I. Radevici, P. Kivisaari, y J. Oksanen, «Electroluminescent Cooling in III–V Intracavity Diodes: Efficiency Bottlenecks», *IEEE Trans Electron Devices*, vol. 66, n.º 6, pp. 2651-2656, 2019, doi: 10.1109/TED.2019.2910219.
- [7] «Calibración y caracterización de IMU · VectorNav». Accedido: 30 de julio de 2025. [En línea]. Disponible en: <https://www-vectornav-com.translate.goog/resources/inertial-navigation-primer/specifications--and-->

error-budgets/specs-
imucal?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc

- [8] «MPU-9250 | TDK InvenSense». Accedido: 31 de julio de 2025. [En línea]. Disponible en: <https://invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250/>
- [9] «9DoF Razor IMU M0 Hookup Guide - SparkFun Learn». Accedido: 31 de julio de 2025. [En línea]. Disponible en: <https://learn.sparkfun.com/tutorials/9dof-razor-imu-m0-hookup-guide>
- [10] M. Babiuch, P. Foltýnek, y P. Smutný, «Using the ESP32 Microcontroller for Data Processing», en *2019 20th International Carpathian Control Conference (ICCC)*, 2019, pp. 1-6. doi: 10.1109/CarpathianCC.2019.8765944.
- [11] M. Jamuna y A. M. Prakash, *A Study of Communication Protocols for Internet of Things (IoT) Devices: Review*. 2021. doi: 10.2991/ahis.k.210913.033.
- [12] «Unity - Manual: Mecanim Animation System». Accedido: 31 de julio de 2025. [En línea]. Disponible en: <https://docs.unity3d.com/462/Documentation/Manual/MecanimAnimationSystem.html>
- [13] G. V Eleftheriades, «Remembering Keith G. Balmain [In Memoriam]», *IEEE Antennas Propag Mag*, vol. 61, n.º 2, p. 138, 2019, doi: 10.1109/MAP.2019.2895596.
- [14] «UNE-EN ISO 6385:2016 (Ratificada) Principios ergonómicos para ...» Accedido: 31 de julio de 2025. [En línea]. Disponible en: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0057285>
- [15] «¿Qué es la ergonomía (HFE)? | Asociación Internacional de Ergonomía». Accedido: 31 de julio de 2025. [En línea]. Disponible en: <https://iea.cc/about/what-is-ergonomics/>
- [16] S. A. Tomaz *et al.*, «Self-Reported Physical Activity in Middle-Aged and Older Adults in Rural South Africa: Levels and Correlates», *International Journal of Environmental Research and Public Health* 2020, Vol. 17, Page 6325, vol. 17, n.º 17, p. 6325, ago. 2020, doi: 10.3390/IJERPH17176325.

- [17] «Método RULA - Rapid Upper Limb Assessment». Accedido: 6 de agosto de 2025. [En línea]. Disponible en: <https://www.ergonautas.upv.es/metodos/rula/rula-ayuda.php>
- [18] «Reglamento de Seguridad y Salud en el Trabajo - Ministerio del Trabajo». Accedido: 31 de julio de 2025. [En línea]. Disponible en: <https://www.trabajo.gob.ec/reglamento-de-seguridad-y-salud-en-el-trabajo/>
- [19] Ministerio del Trabajo de Ecuador, *Reglamento de Seguridad y Salud de los Trabajadores*. 2024.

ANEXOS

ANEXO A

DIAGRAMA ELECTRICO DEL SISTEMA AUTOMATIZADO.

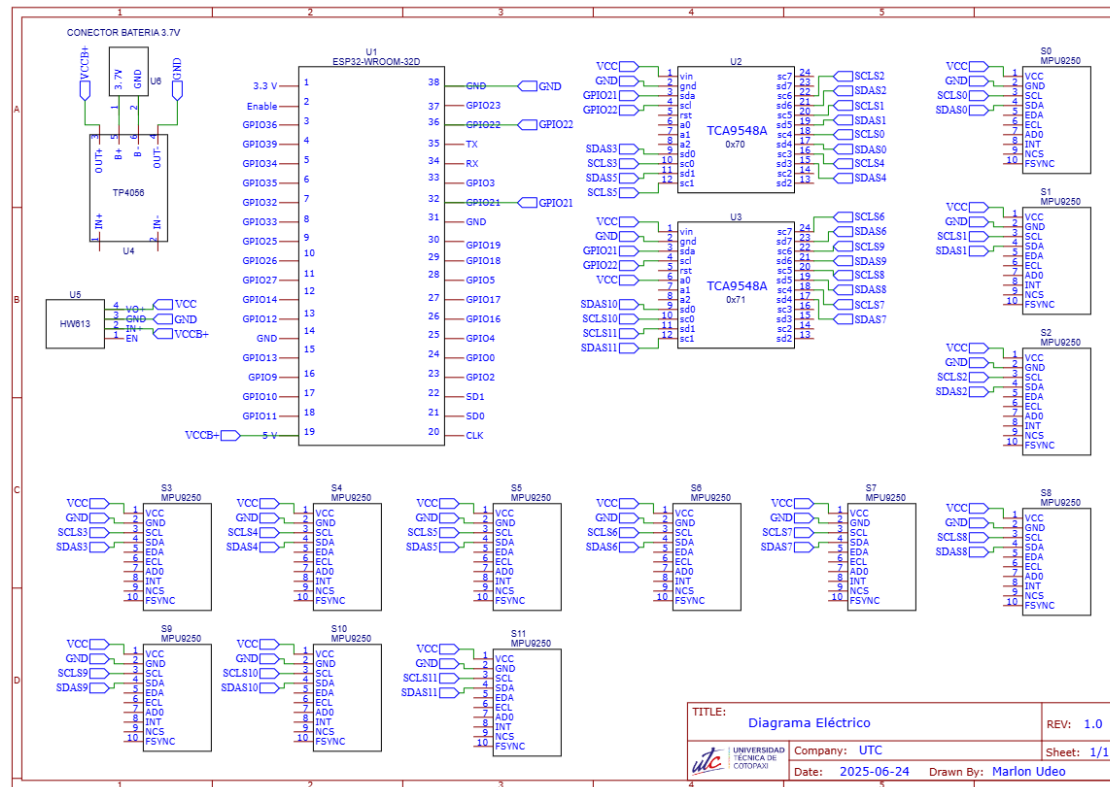


Figura A.1 Diagrama de conexiones

ANEXO B

CÓDIGO PARA LECTURA DE SENSORES Y ENVÍO DE DATOS MEDIANTE WEBSOCKET.

```
#include <Wire.h>
#include <WiFi.h>
#include <WebSocketsServer.h>
#include <ArduinoJson.h>
#include <SparkFunMPU9250-DMP.h>

const char* ssid = "WIFI_UTC";
const char* password = "M4tr1Z#%$";

// WebSocket server
WebSocketsServer webSocket(81);

// Estructura de sensor con grupo
struct SensorConfig {
    uint8_t mpxAddress;
    uint8_t mpxChannel;
    uint8_t sensorID;
    MPU9250_DMP imu;
    uint8_t grupo;
};

// Sensores conectados a multiplexores
SensorConfig sensores[] = {
    {0x71, 2, 0, {}, true, 0}, // Brazo izquierdo
    {0x71, 3, 1, {}, true, 0}, // Antebrazo izquierdo
    {0x71, 4, 2, {}, true, 0}, // Mano izquierda
    {0x71, 5, 3, {}, false, 2}, // Pierna izquierda
    {0x71, 6, 4, {}, false, 2}, // Pantorrilla izquierda
    {0x71, 1, 5, {}, true, 2}, // Espalda baja
    {0x71, 0, 6, {}, true, 0}, // Espalda
    {0x70, 1, 7, {}, true, 1}, // Cabeza
    {0x70, 2, 8, {}, true, 1}, // Brazo derecho
    {0x70, 3, 9, {}, true, 1}, // Antebrazo derecho
    {0x70, 4, 10, {}, true, 1}, // Mano derecha
    {0x70, 5, 11, {}, false, 2}, // Pierna derecha
    {0x70, 6, 12, {}, false, 2} // Pantorrilla derecha
};

const signed char orientationDefault[9] = { 0, 1, 0, 0, 0, 1, 1, 0, 0 };

uint8_t grupoActual = 0;
const uint8_t TOTAL_GRUPOS = 3;
```

```

const uint16_t tiempoGrupoMs = 6;

void selectMuxChannel(uint8_t muxAddr, uint8_t channel) {
    Wire.beginTransmission(muxAddr);
    Wire.write(1 << channel);
    Wire.endTransmission();
    delayMicroseconds(3000);
}

void deselectMuxChannel(uint8_t muxAddr) {
    Wire.beginTransmission(muxAddr);
    Wire.write(0x00);
    Wire.endTransmission();
    delay(1);
}

bool iniciarSensor(SensorConfig &sensor) {
    if (!sensor.activo) return false;

    selectMuxChannel(sensor.mpxAddress, sensor.mpxChannel);

    // Verificar presencia de MPU9250 (dirección 0x68)
    Wire.beginTransmission(0x68);
    bool conectado = (Wire.endTransmission() == 0);

    if (!conectado) {
        Serial.printf("Sensor %d no responde (MPX 0x%02X canal %d)\n",
            sensor.sensorID, sensor.mpxAddress, sensor.mpxChannel);
        sensor.activo = false;
        deselectMuxChannel(sensor.mpxAddress);
        return false;
    }

    // Intentar inicialización
    if (sensor.imu.begin() != INV_SUCCESS) {
        Serial.printf("Sensor %d falló al inicializar IMU\n",
            sensor.sensorID);
        sensor.activo = false;
        deselectMuxChannel(sensor.mpxAddress);
        return false;
    }

    sensor.imu.enableInterrupt();
    sensor.imu.setIntLevel(INT_ACTIVE_LOW);
    sensor.imu.setIntLatched(INT_LATCHED);
    sensor.imu.dmpBegin(DMP_FEATURE_6X_LP_QUAT | DMP_FEATURE_GYRO_CAL, 10);
    sensor.imu.dmpSetOrientation(orientationDefault);
}

```

```

    Serial.printf("✓ Sensor %d inicializado correctamente\n",
sensor.sensorID);
    deselectMuxChannel(sensor.mpxAddress);
    return true;
}

void setup() {
    Serial.begin(115200);
    Serial.printf("Conectando a WiFi: %s\n", ssid);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("\nWiFi conectado!");
    Serial.print("IP del ESP32: ");
    Serial.println(WiFi.localIP());

    Serial.println("Continuando en 5 segundos...");
    delay(5000);

    Wire.begin();
    Wire.setClock(400000); // I2C a 400 kHz

    websocket.begin();
    Serial.println("WebSocket iniciado en puerto 81");

    for (SensorConfig &sensor : sensores) {
        iniciarSensor(sensor);
    }

    xTaskCreatePinnedToCore(
        sensorTask,
        "SensorTask",
        NULL,
        1,
        NULL,
        0
    );
}

void sensorTask(void *param) {
    for (;;) {
        if (millis() - ultimoCambioGrupo >= tiempoGrupoMs) {
            ultimoCambioGrupo = millis();
        }
    }
}

```

```

StaticJsonDocument<3072> doc;
JSONArray sensoresArray = doc.createNestedArray("sensores");

for (SensorConfig &sensor : sensores) {
    if (!sensor.activo || sensor.grupo != grupoActual) continue;

    selectMuxChannel(sensor.mpxAddress, sensor.mpxChannel);

    if (sensor.imu.fifoAvailable() >= 1024) {
        Serial.printf("\u260a FIFO LLENA sensor %d - reiniciando FIFO\n",
sensor.sensorID);
        sensor.imu.resetFifo();
        deselectMuxChannel(sensor.mpxAddress);
        continue;
    }

    if (!sensor.imu.fifoAvailable()) {
        deselectMuxChannel(sensor.mpxAddress);
        continue;
    }

    if (sensor.imu.dmpUpdateFifo() == INV_SUCCESS) {
        float q0 = sensor.imu.calcQuat(sensor.imu.qw);
        float q1 = sensor.imu.calcQuat(sensor.imu.qx);
        float q2 = sensor.imu.calcQuat(sensor.imu.qy);
        float q3 = sensor.imu.calcQuat(sensor.imu.qz);

        JsonObject obj = sensoresArray.createNestedObject();
        obj["id"] = sensor.sensorID;
        obj["q0"] = q0;
        obj["q1"] = q1;
        obj["q2"] = q3;
        obj["q3"] = q2;

        Serial.printf("Grupo %d | Sensor %2d | Q = [%.3f, %.3f, %.3f,
%.3f]\n",
                    grupoActual, sensor.sensorID, q0, q1, q3, q2);
    }

    deselectMuxChannel(sensor.mpxAddress);
}

if (sensoresArray.size() > 0) {
    String json;
    serializeJson(doc, json);
    websocket.broadcastTXT(json);
}

```

```
        delay(1);
    }
}

void loop() {
    websocket.loop();
    delay(1);
}
```

ANEXO C

Introducción al Método RULA

El método RULA evalúa posturas individuales y no conjuntos o secuencias de posturas, por ello, es necesario seleccionar aquellas posturas que serán evaluadas de entre las que adopta el trabajador en el puesto. Se seleccionarán aquellas que, a priori, supongan una mayor carga postural bien por su duración, bien por su frecuencia o porque presentan mayor desviación respecto a la posición neutra.

Para ello, el primer paso consiste en la observación de las tareas que desempeña el trabajador. Se observarán varios ciclos de trabajo y se determinarán las posturas que se evaluarán. Si el ciclo es muy largo o no existen ciclos, se pueden realizar evaluaciones a intervalos regulares. En este caso se considerará, además, el tiempo que pasa el trabajador en cada postura.

Las mediciones a realizar sobre las posturas adoptadas por el trabajador son fundamentalmente angulares (los ángulos que forman los diferentes miembros del cuerpo respecto a determinadas referencias). Estas mediciones pueden realizarse directamente sobre el trabajador mediante transportadores de ángulos, electro goniómetros, o cualquier dispositivo que permita la toma de datos angulares. También es posible emplear fotografías del trabajador adoptando la postura estudiada y medir los ángulos sobre éstas. Si se utilizan fotografías es necesario realizar un número suficiente de tomas desde diferentes puntos de vista (alzado, perfil, vistas de detalle). Es muy importante en este caso asegurarse de que los ángulos a medir aparecen en verdadera magnitud en las imágenes, es decir, que el plano en el que se encuentra el ángulo a medir es paralelo al plano de la cámara como se ve en la Figura C.1. Para esta tarea puedes emplear RULER, la herramienta de Ergonautas para medir ángulos sobre fotografías.

El método debe ser aplicado al lado derecho y al lado izquierdo del cuerpo por separado. El evaluador experto puede elegir a priori el lado que aparentemente esté sometido a mayor carga postural, pero en caso de duda es preferible analizar los dos lados.

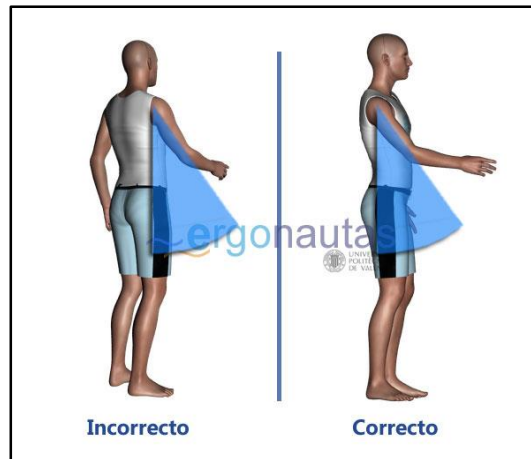


Figura C. 1 Medición de ángulos en RULA



Figura C. 2 Grupos de miembros en RULA.

RULA divide el cuerpo en dos grupos como se ve en la Figura C.2, el Grupo A que incluye los miembros superiores (brazos, antebrazos y muñecas) y el Grupo B, que comprende las piernas, el tronco y el cuello. Mediante las tablas asociadas al método, se asigna una puntuación a cada zona corporal (piernas, muñecas, brazos, tronco...) para, en función de dichas puntuaciones, asignar valores globales a cada uno de los grupos A y B.

La clave para la asignación de puntuaciones a los miembros es la medición de los ángulos que forman las diferentes partes del cuerpo del operario. El método determina para cada miembro la forma de medición del ángulo. Posteriormente, las puntuaciones globales de

los grupos A y B son modificadas en función del tipo de actividad muscular desarrollada, así como de la fuerza aplicada durante la realización de la tarea. Por último, se obtiene la puntuación final a partir de dichos valores globales modificados.

El valor final proporcionado por el método RULA es proporcional al riesgo que conlleva la realización de la tarea, de forma que valores altos indican un mayor riesgo de aparición de lesiones músculo-esqueléticas. El método organiza las puntuaciones finales en niveles de actuación que orientan al evaluador sobre las decisiones a tomar tras el análisis. Los niveles de actuación propuestos van del nivel 1, que estima que la postura evaluada resulta aceptable, al nivel 4, que indica la necesidad urgente de cambios en la actividad.

Aplicación del método

Evaluación del Grupo A

La puntuación del Grupo A se obtiene a partir de las puntuaciones de cada uno de los miembros que lo componen (brazo, antebrazo y muñeca). Así pues, como paso previo a la obtención de la puntuación del grupo hay que obtener las puntuaciones de cada miembro.

Puntuación del brazo

La puntuación del brazo se obtiene a partir de su grado de flexión/extensión. Para ello se medirá el ángulo formado por el eje del brazo y el eje del tronco. La Figura C 3 muestra los diferentes grados de flexión/extensión considerados por el método. La puntuación del brazo se obtiene mediante la Tabla C. 1.

La puntuación obtenida de esta forma valora la flexión del brazo. Esta puntuación será aumentada en un punto si existe elevación del hombro, si el brazo está abducido (separado del tronco en el plano sagital) o si existe rotación del brazo. Si existe un punto de apoyo sobre el que descansa el brazo del trabajador mientras desarrolla la tarea la puntuación del brazo disminuye en un punto. Si no se da ninguna de estas circunstancias la puntuación del brazo no se modifica. Para obtener la puntuación definitiva del brazo puede consultarse la Tabla C. 2 y la Figura C. 4.

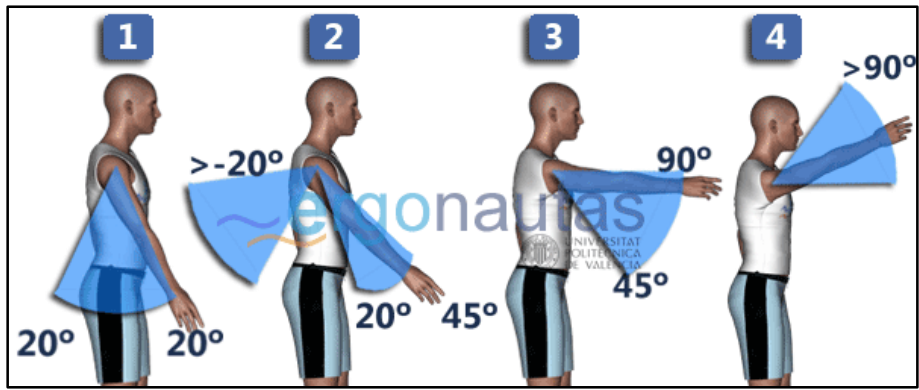


Figura C. 3 Medición del ángulo del brazo.

Tabla C. 1 Puntuación del brazo.

Posición	Puntuación
Desde 20° de extensión a 20° de flexión	1
Extensión >20° o flexión >20° y <45°	2
Flexión >45° y 90°	3
Flexión >90°	4

Tabla C. 2 Modificación de la puntuación del brazo.

Posición	Puntuación
Hombro elevado o brazo rotado	+1
Brazos abducidos	+1
Existe un punto de apoyo	-1

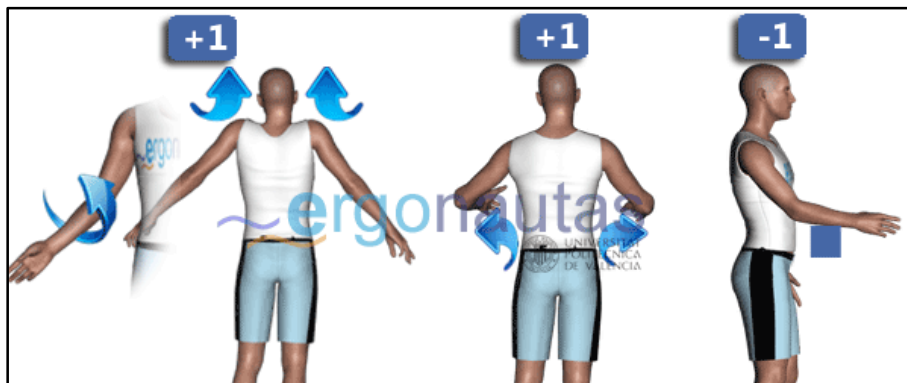


Figura C. 4 Modificación de la puntuación del brazo.

Puntuación del antebrazo

La puntuación del antebrazo se obtiene a partir de su ángulo de flexión, medido como el ángulo formado por el eje del antebrazo y el eje del brazo. La Figura C. 5 muestra los intervalos de flexión considerados por el método. La puntuación del antebrazo se obtiene mediante la Tabla C. 3.

La puntuación así obtenida valora la flexión del antebrazo. Esta puntuación se aumentará en un punto si el antebrazo cruza la línea media del cuerpo, o si se realiza una actividad a un lado del cuerpo como se ve en la Figura C. 6. Ambos casos son excluyentes, por lo que como máximo se aumentará un punto la puntuación inicial del antebrazo. La Tabla C. 4 muestra los incrementos a aplicar.

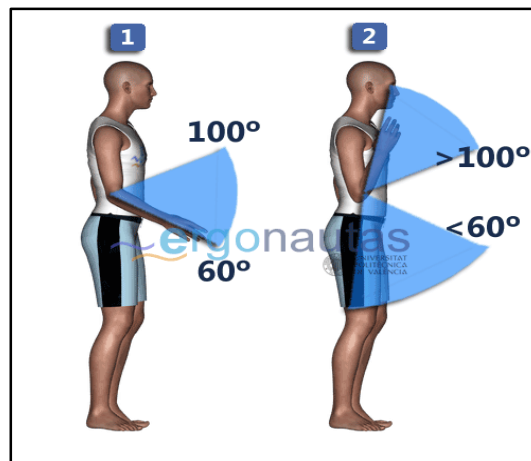


Figura C. 5 Medición del ángulo del antebrazo.

Tabla C. 3 Puntuación del antebrazo.

Posición	Puntuación
Flexión entre 60° y 100°	1
Flexión <60° o >100°	2



Figura C. 6 Modificación de la puntuación del antebrazo.

Tabla C. 4 Modificación de la puntuación del antebrazo.

Posición	Puntuación
A un lado del cuerpo	+1
Cruza la línea media	+1

Puntuación de la muñeca

La puntuación de la muñeca se obtiene a partir del ángulo de flexión/extensión medido desde la posición neutra. La Figura C. 7 muestra las referencias para realizar la medición.

La puntuación de la muñeca se obtiene mediante la Tabla C. 5.

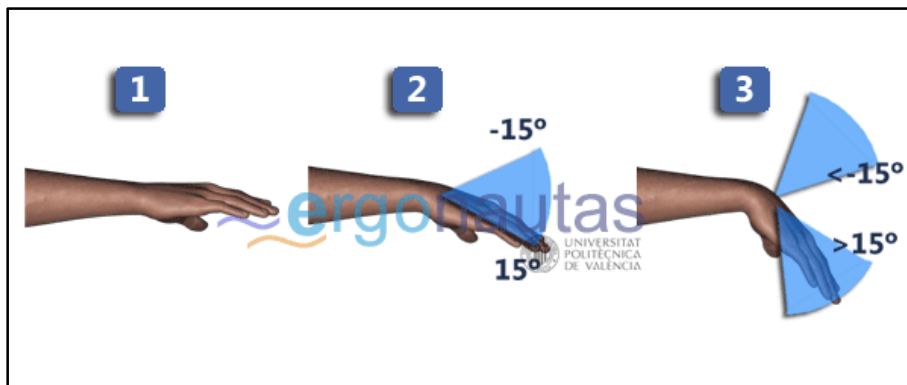


Figura C. 7 Medición del ángulo de la muñeca.

Tabla C. 5 Puntuación de la muñeca.

Posición	Puntuación
Posición neutra	1
Flexión o extensión $> 0^\circ$ y $< 15^\circ$	2
Flexión o extensión $> 15^\circ$	3

La puntuación obtenida de esta forma valora la flexión de la muñeca. Esta puntuación se aumentará en un punto si existe desviación radial o cubital. Ambos casos son excluyentes, por lo que como máximo se aumentará un punto la puntuación inicial de la muñeca.

Una vez obtenida la puntuación de la muñeca se valorará el giro de la misma. Este nuevo valor será independiente y no se añadirá a la puntuación anterior, si no que servirá posteriormente para obtener la valoración global del Grupo A. Se trata de valorar el grado de pronación o supinación de la mano (medio o extremo). Si no existe pronación/supinación o su grado es medio se asignará una puntuación de 1; si el grado es extremo la puntuación será 2 muestra el incremento a aplicar.



Figura C. 8 Modificación de la puntuación de la muñeca.

Tabla C. 6 Modificación de la puntuación de la muñeca.

Posición	Puntuación
Desviación radial	+1
Desviación cubital	+1

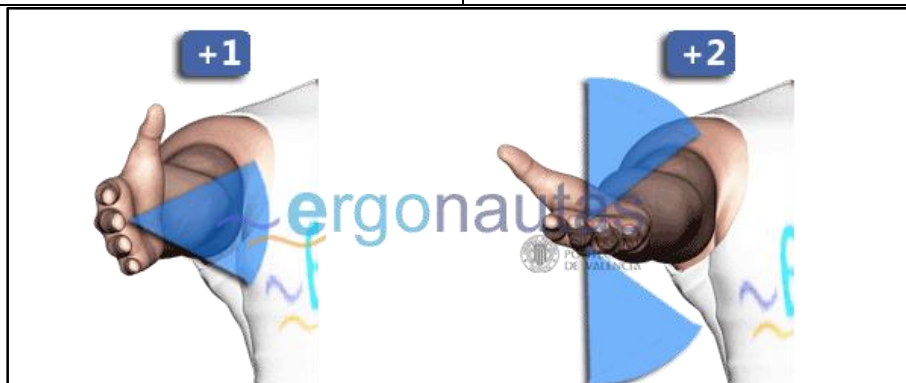


Figura C. 9 Puntuación del giro de muñeca.

Tabla C. 7 Puntuación del giro de la muñeca.

Posición	Puntuación
Pronación o supinación media	1
Pronación o supinación extrema	2

Evaluación del Grupo B

La puntuación del Grupo B se obtiene a partir de las puntuaciones de cada uno de los miembros que lo componen (cuello, tronco y piernas). Por ello, como paso previo a la obtención de la puntuación del grupo hay que obtener las puntuaciones de cada miembro.

Puntuación del cuello

La puntuación del cuello se obtiene a partir de la flexión/extensión medida por el ángulo formado por el eje de la cabeza y el eje del tronco. La Figura C. 10 muestra las referencias para realizar la medición. La puntuación del cuello se obtiene mediante la Tabla C. 8.

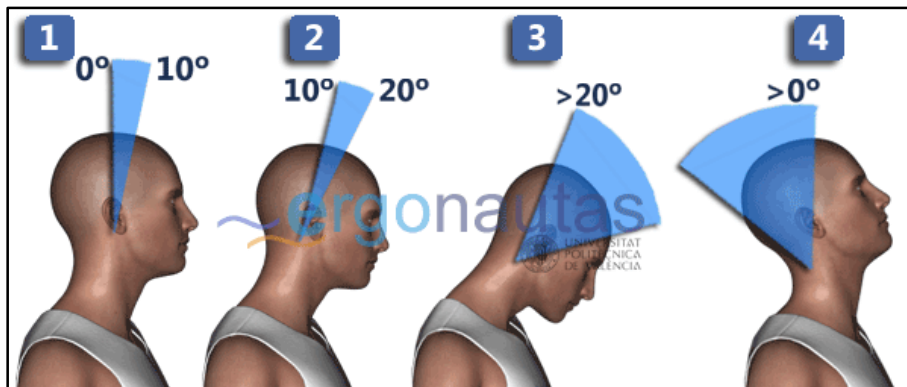


Figura C. 10 Medición del ángulo del cuello.

Tabla C. 8 Puntuación del cuello.

Posición	Puntuación
Flexión entre 0° y 10°	1
Flexión >10° y ≤20°	2
Flexión >20°	3
Extensión en cualquier grado	4

La puntuación obtenida de esta forma valora la flexión del cuello. Esta puntuación será aumentada en un punto si existe rotación o inclinación lateral de la cabeza. Ambas circunstancias pueden ocurrir simultáneamente, por lo que la puntuación del cuello puede aumentar hasta en dos puntos. Si no se da ninguna de estas circunstancias la puntuación

del cuello no se modifica. Para obtener la puntuación definitiva del cuello puede consultarse la Figura C. 11 y Tabla C. 9.

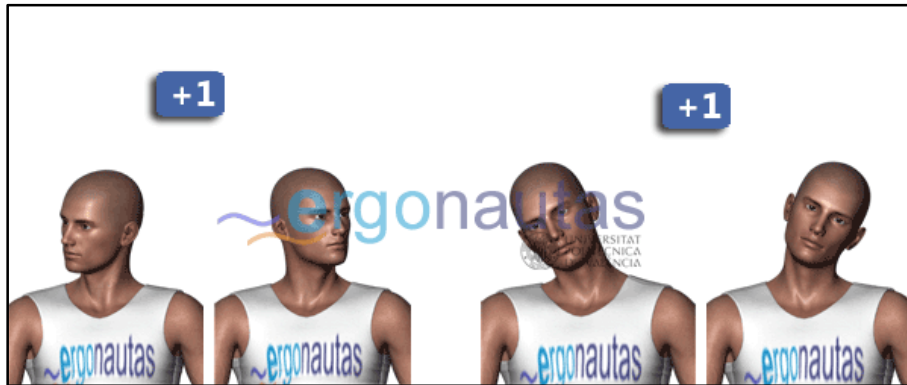


Figura C. 11 Modificación de la puntuación del cuello.

Tabla C. 9 Modificación de la puntuación del cuello.

Posición	Puntuación
Cabeza rotada	+1
Cabeza con inclinación lateral	+1

Puntuación del tronco

La puntuación del tronco dependerá de si el trabajador realiza la tarea sentado o de pie. En este último caso la puntuación dependerá del ángulo de flexión del tronco medido por el ángulo entre el eje del tronco y la vertical. La Figura C. 12 muestra las referencias para realizar la medición. La puntuación del tronco se obtiene mediante la

Tabla C. 10.

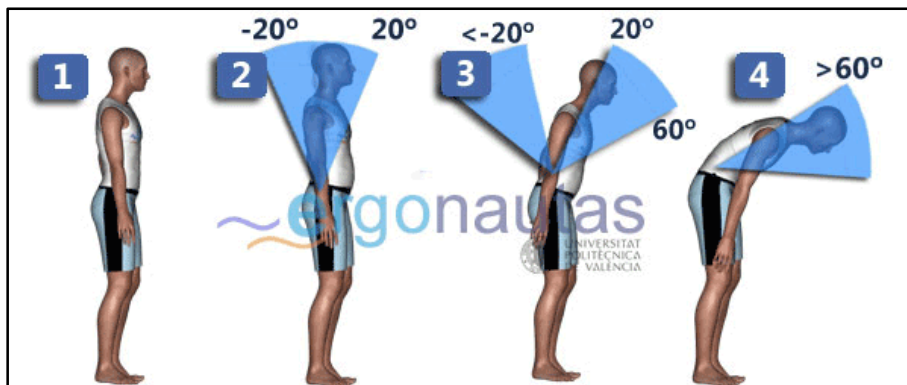


Figura C. 12 Medición del ángulo del tronco.

Tabla C. 10 Puntuación del tronco.

Posición	Puntuación
Sentado, bien apoyado y con un ángulo tronco-caderas $>90^\circ$	1
Flexión entre 0° y 20°	2
Flexión $>20^\circ$ y $\leq 60^\circ$	3
Flexión $>60^\circ$	4

La puntuación obtenida de esta forma valora la flexión del tronco. Esta puntuación será aumentada en un punto si existe rotación o inclinación lateral del tronco. Ambas circunstancias pueden ocurrir simultáneamente, por lo que la puntuación del tronco puede aumentar hasta en dos puntos Si no se da ninguna de estas circunstancias la puntuación del tronco no se modifica. Para obtener la puntuación definitiva del tronco puede consultarse la Figura C. 13 y Tabla C. 11.



Figura C. 13 Modificación de la puntuación del tronco.

Tabla C. 11 Modificación de la puntuación del tronco.

Posición	Puntuación
Tronco rotado	+1
Tronco con inclinación lateral	+1

Puntuación de las piernas

La puntuación de las piernas dependerá de la distribución del peso entre las ellas, los apoyos existentes y si la posición es sedente. La puntuación de las piernas se obtiene mediante la Figura C. 14 y Tabla C. 12.

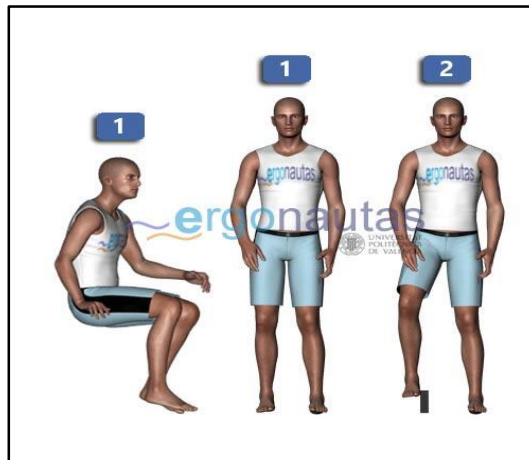


Figura C. 14 Puntuación de las piernas.

Tabla C. 12 Puntuación de las piernas.

Posición	Puntuación
Sentado, con piernas y pies bien apoyados	1
De pie con el peso simétricamente distribuido y espacio para cambiar de posición	1
Los pies no están apoyados o el peso no está simétricamente distribuido	2

Puntuación de los Grupos A y B

Obtenidas las puntuaciones de cada uno de los miembros que conforman los Grupos A y B se calculará las puntuaciones globales de cada Grupo. Para obtener la puntuación del Grupo A se empleará la

Tabla C. 13, mientras que para la del Grupo B se utilizará la Tabla C. 14.

Tabla C. 13 Puntuación del Grupo A.

		Muñeca							
		1		2		3		4	
		Giro de Muñeca		Giro de Muñeca		Giro de Muñeca		Giro de Muñeca	
Brazo	Antebrazo	1	2	1	2	1	2	1	2
1	1	1	2	2	2	2	3	3	3
	2	2	2	2	2	3	3	3	3
	3	2	3	3	3	3	3	4	4
2	1	2	3	3	3	3	4	4	4
	2	3	3	3	3	3	4	4	4
	3	3	4	4	4	4	4	5	5
3	1	3	3	4	4	4	4	5	5
	2	3	4	4	4	4	4	5	5
	3	4	4	4	4	4	5	5	5
4	1	4	4	4	4	4	5	5	5
	2	4	4	4	4	4	5	5	5
	3	4	4	4	5	5	5	6	6
5	1	5	5	5	5	5	6	6	7
	2	5	6	6	6	6	7	7	7
	3	6	6	6	7	7	7	7	8
6	1	7	7	7	7	7	8	8	9
	2	8	8	8	8	8	9	9	9
	3	9	9	9	9	9	9	9	9

Tabla C. 14 Puntuación del Grupo B.

		Tronco											
		1		2		3		4		5		6	
		Piernas		Piernas		Piernas		Piernas		Piernas		Piernas	
Cuello		1	2	1	2	1	2	1	2	1	2	1	2
1		1	3	2	3	3	4	5	5	6	6	7	7
2		2	3	2	3	4	5	5	5	6	7	7	7
3		3	3	3	4	4	5	5	6	6	7	7	7
4		5	5	5	6	6	7	7	7	7	7	8	8
5		7	7	7	7	7	8	8	8	8	8	8	8
6		8	8	8	8	8	8	8	9	9	9	9	9

Puntuación final

Las puntuaciones globales de los Grupos A y B consideran la postura del trabajador. A continuación, se valorará el carácter estático o dinámico de la misma y las fuerzas ejercidas durante su adopción.

La puntuación de los Grupos A y B se incrementarán en un punto si la actividad es básicamente estática (la postura se mantiene más de un minuto seguido) o bien si es repetitiva (se repite más de 4 veces cada minuto). Si la tarea es ocasional, poco frecuente y de corta duración, se considerará actividad dinámica y las puntuaciones no se modificarán Tabla C. 15.

Por otra parte, se incrementarán las puntuaciones anteriores en función de las fuerzas ejercidas. La Tabla C. 16 muestra el incremento en función de la carga soportada o fuerzas ejercidas.

Las puntuaciones de los Grupos A y B, incrementadas por las puntuaciones correspondientes al tipo de actividad y las cargas o fuerzas ejercidas pasarán a denominarse puntuaciones C y D respectivamente.

Las puntuaciones C y D permiten obtener la puntuación final del método empleando la Tabla C. 17. Ésta puntuación final global para la tarea oscilará entre 1 y 7, siendo mayor cuanto más elevado sea el riesgo.

Tabla C. 15 Puntuación por tipo de actividad.

Tipo de actividad	Puntuación
Estática (se mantiene más de un minuto seguido)	+1
Repetitiva (se repite más de 4 veces cada minuto)	+1
Ocasional, poco frecuente y de corta duración	0

Tabla C. 16 Puntuación por carga o fuerzas ejercidas.

Carga o fuerza	Puntuación
Carga menor de 2 Kg. mantenida intermitentemente	0
Carga entre 2 y 10 Kg. mantenida intermitentemente	+1
Carga entre 2 y 10 Kg. estática o repetitiva	+2
Carga superior a 10 Kg mantenida intermitentemente	+2
Carga superior a 10 Kg estática o repetitiva	+3
Se producen golpes o fuerzas bruscas o repentinas	+3

Tabla C. 17 Puntuación Final RULA.

Puntuación C	Puntuación D						
	1	2	3	4	5	6	7
1	1	2	3	3	4	5	5
2	2	2	3	4	4	5	5
3	3	3	3	4	4	5	6
4	3	3	3	4	5	6	6
5	4	4	4	5	6	7	7
6	4	4	5	6	6	7	7
7	5	5	6	6	7	7	7
8	5	5	6	7	7	7	7

Nivel de Actuación

Obtenida la puntuación final la Tabla C. 18 propone diferentes niveles de actuación sobre el puesto. Puntuaciones entre 1 y 2 indican que el riesgo de la tarea resulta aceptable y que no son precisos cambios. Puntuaciones entre 3 y 4 indican que es necesario un estudio en profundidad del puesto porque pueden requerirse cambios. Puntuaciones entre 5 y 6 indican que los cambios son necesarios y 7 indica que los cambios son urgentes. Las puntuaciones de cada miembro y grupo, así como las puntuaciones de fuerza y actividad muscular, indicarán al evaluador los aspectos en los que actuar para mejorar el puesto.

Finalmente, la Figura C. 15 resume el proceso de obtención del Nivel de Actuación en el método Rula.

Tabla C. 18 Niveles de actuación según la puntuación final obtenida.

Puntuación	Nivel	Actuación
1 o 2	1	Riesgo Aceptable
3 o 4	2	Pueden requerirse cambios en la tarea; es conveniente profundizar en el estudio
5 o 6	3	Se requiere el rediseño de la tarea
7	4	Se requieren cambios urgentes en la tarea



Figura C. 15 Esquema de puntuaciones.

ANEXO D

ESP32 Datasheet



Espressif Systems

October 8, 2016

1. Overview

ESP32 is a single chip 2.4 GHz Wi-Fi and Bluetooth combo chip designed with TSMC ultra low power 40 nm technology. It is designed and optimized for the best power performance, RF performance, robustness, versatility, features and reliability, for a wide variety of applications, and different power profiles.

1.1 Featured Solutions

1.1.1 Ultra Low Power Solution

ESP32 is designed for mobile, wearable electronics, and Internet of Things (IoT) applications. It has many features of the state-of-the-art low power chips, including fine resolution clock gating, power modes, and dynamic power scaling.

For instance, in a low-power IoT sensor hub application scenario, ESP32 is woken up periodically and only when a specified condition is detected; low duty cycle is used to minimize the amount of energy that the chip expends. The output power of the power amplifier is also adjustable to achieve an optimal trade off between communication range, data rate and power consumption.

Note:

For more information, refer to Section [3.7](#) RTC and Low-Power Management.

1.1.2 Complete Integration Solution

ESP32 is the most integrated solution for Wi-Fi + Bluetooth applications in the industry with less than 10 external components. ESP32 integrates the antenna switch, RF balun, power amplifier, low noise receive amplifier, filters, and power management modules. As such, the entire solution occupies minimal Printed Circuit Board (PCB) area.

ESP32 uses CMOS for single-chip fully-integrated radio and baseband, and also integrates advanced calibration circuitries that allow the solution to dynamically adjust itself to remove external circuit imperfections or adjust to changes in external conditions.

As such, the mass production of ESP32 solutions does not require expensive and specialized Wi-Fi test equipment.

1.2 Basic Protocols

1.2.1 Wi-Fi

- 802.11 b/g/n/e/i
- 802.11 n (2.4 GHz), up to 150 Mbps
- 802.11 e: QoS for wireless multimedia technology
- WMM-PS, UAPSD
- A-MPDU and A-MSDU aggregation
- Block ACK

- Fragmentation and defragmentation
- Automatic Beacon monitoring/scanning
- 802.11 i security features: pre-authentication and TSN
- Wi-Fi Protected Access (WPA)/WPA2/WPA2-Enterprise/Wi-Fi Protected Setup (WPS)
- Infrastructure BSS Station mode/SoftAP mode
- Wi-Fi Direct (P2P), P2P Discovery, P2P Group Owner mode and P2P Power Management
- UMA compliant and certified
- Antenna diversity and selection

Note:

For more information, refer to Section 3.5 Wi-Fi.

1.2.2 Bluetooth

- Compliant with Bluetooth v4.2 BR/EDR and BLE specification
- Class-1, class-2 and class-3 transmitter without external power amplifier
- Enhanced power control
- +10 dBm transmitting power
- NZIF receiver with -98 dBm sensitivity
- Adaptive Frequency Hopping (AFH)
- Standard HCI based on SDIO/SPI/UART
- High speed UART HCI, up to 4 Mbps
- BT 4.2 controller and host stack
- Service Discover Protocol (SDP)
- General Access Profile (GAP)
- Security Manage Protocol (SMP)
- Bluetooth Low Energy (BLE)
- ATT/GATT
- HID
- All GATT-based profile supported
- SPP-Like GATT-based profile
- BLE Beacon
- A2DP/AVRCP/SPP, HSP/HFP, RFCOMM
- CVSD and SBC for audio codec
- Bluetooth Piconet and Scatternet

1.3 MCU and Advanced Features

1.3.1 CPU and Memory

- Xtensa® Dual-Core 32-bit LX6 microprocessors, up to 600 DMIPS
- 448 KByte ROM
- 520 KByte SRAM
- 16 KByte SRAM in RTC
- QSPI Flash/SRAM, up to 4 x 16 MBytes
- Power supply: 2.2 V to 3.6 V

1.3.2 Clocks and Timers

- Internal 8 MHz oscillator with calibration
- Internal RC oscillator with calibration
- External 2 MHz to 40 MHz crystal oscillator
- External 32 kHz crystal oscillator for RTC with calibration
- Two timer groups, including 2 x 64-bit timers and 1 x main watchdog in each group
- RTC timer with sub-second accuracy
- RTC watchdog

1.3.3 Advanced Peripheral Interfaces

- 12-bit SAR ADC up to 18 channels
- 2 x 8-bit D/A converters
- 10 x touch sensors
- Temperature sensor
- 4 x SPI
- 2 x I2S
- 2 x I2C
- 3 x UART
- 1 host (SD/eMMC/SDIO)
- 1 slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- CAN 2.0
- IR (TX/RX)
- Motor PWM
- LED PWM up to 16 channels
- Hall sensor
- Ultra low power analog pre-amplifier

1.3.4 Security

- IEEE 802.11 standard security features all supported, including WPA, WPA/WPA2 and WAPI
- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration:
 - AES
 - HASH (SHA-2) library
 - RSA
 - ECC
 - Random Number Generator (RNG)

1.3.5 Development Support

- SDK Firmware for fast on-line programming
- Open source toolchains based on GCC

Note:

For more information, refer to Chapter 7 Supported Resources.

1.4 Application

- Generic low power IoT sensor hub
- Generic low power IoT loggers
- Video streaming from camera
- Over The Top (OTT) devices
- Music players
 - Internet music players
 - Audio streaming devices
- Wi-Fi enabled toys
 - Loggers
 - Proximity sensing toys
- Wi-Fi enabled speech recognition devices
- Audio headsets
- Smart power plugs
- Home automation
- Mesh network

- Industrial wireless control
- Baby monitors
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags
- Healthcare
 - Proximity and movement monitoring trigger devices
 - Temperature sensing loggers

1.5 Block Diagram

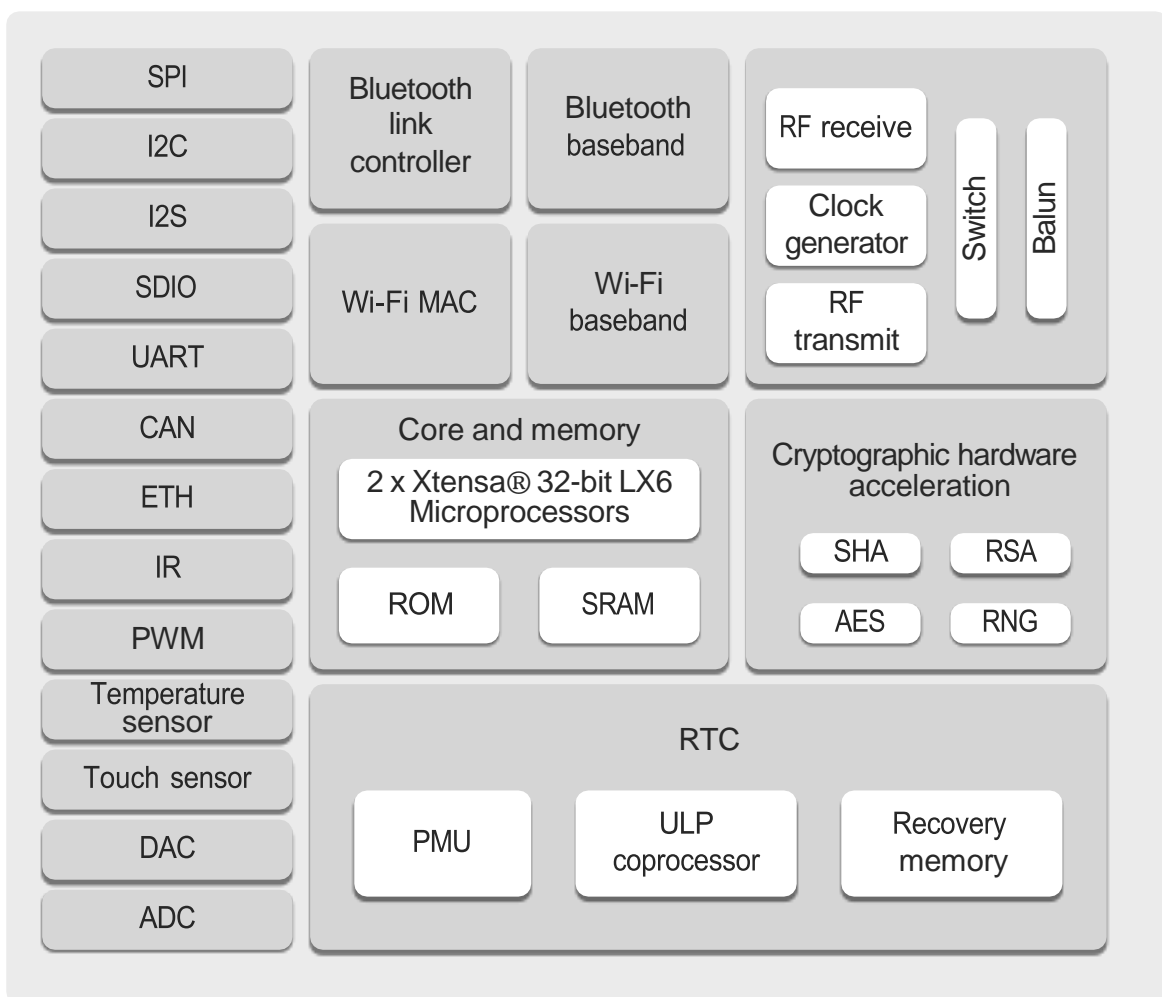


Figure 1: Function Block Diagram

2. Pin Definitions

2.1 Pin Layout

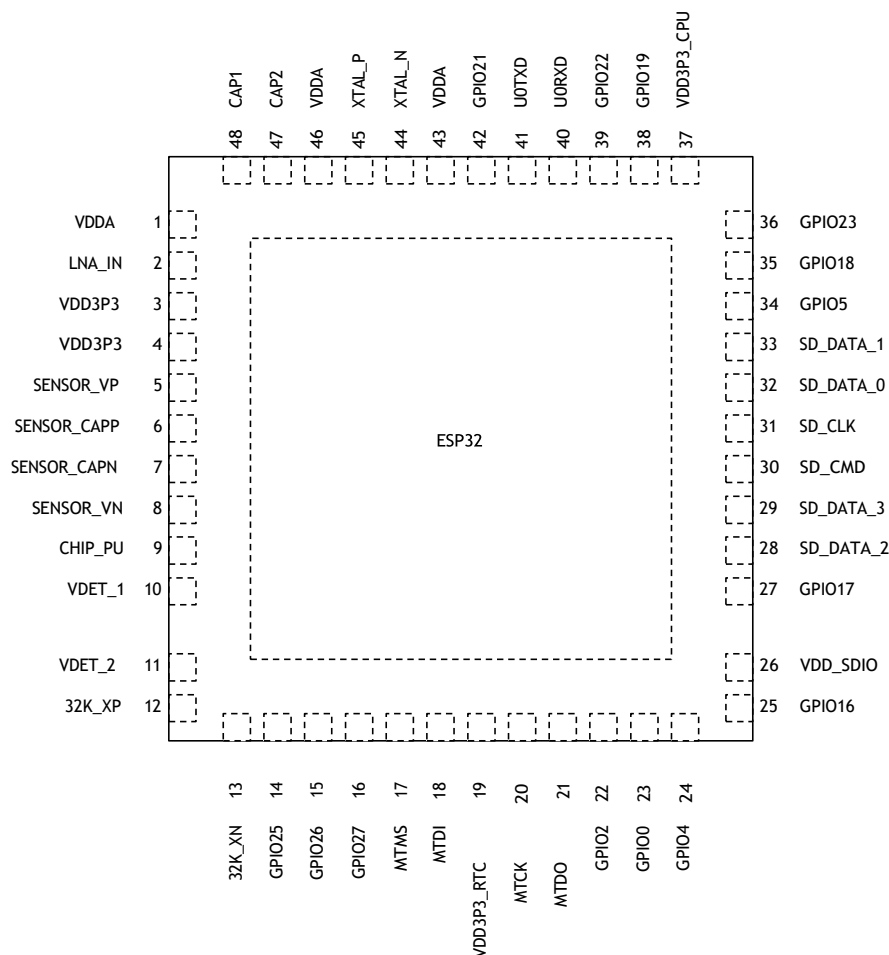


Figure 2: ESP32 Pin Layout

2.2 Pin Description

Table 1: Pin Description

Name	No.	Type	Function
Analog			
VDDA	1	P	Analog power supply (2.3V ~ 3.6V)
LNA_IN	2	I/O	RF input and output
VDD3P3	3	P	Amplifier power supply (2.3V ~ 3.6V)
VDD3P3	4	P	Amplifier power supply (2.3V ~ 3.6V)
VDD3P3_RTC			
SENSOR_VP	5	I	GPIO36, ADC_PRE_AMP, ADC1_CH0, RTC_GPIO0 Note: Connects 270 pF capacitor from SENSOR_VP to SENSOR_CAPP when used as ADC_PRE_AMP.

Name	No.	Type	Function
SENSOR_CAPP	6	I	GPIO37, ADC_PRE_AMP, ADC1_CH1, RTC_GPIO1 Note: Connects 270 pF capacitor from SENSOR_VP to SENSOR_CAPP when used as ADC_PRE_AMP.
SENSOR_CAPN	7	I	GPIO38, ADC1_CH2, ADC_PRE_AMP, RTC_GPIO2 Note: Connects 270 pF capacitor from SENSOR_VN to SENSOR_CAPN when used as ADC_PRE_AMP.
SENSOR_VN	8	I	GPIO39, ADC1_CH3, ADC_PRE_AMP, RTC_GPIO3 Note: Connects 270 pF capacitor from SENSOR_VN to SENSOR_CAPN when used as ADC_PRE_AMP.
CHIP_PU	9	I	Chip Enable (Active High) High: On, chip works properly Low: Off, chip works at the minimum power Note: Do not leave CHIP_PU pin floating
VDET_1	10	I	GPIO34, ADC1_CH6, RTC_GPIO4
VDET_2	11	I	GPIO35, ADC1_CH7, RTC_GPIO5
32K_XP	12	I/O	GPIO32, 32K_XP (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
32K_XN	13	I/O	GPIO33, 32K_XN (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
GPIO25	14	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
GPIO26	15	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
GPIO27	16	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
MTMS	17	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPI-CLK, HS2_CLK, SD_CLK, EMAC_TXD2
MTDI	18	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
VDD3P3_RTC	19	P	RTC IO power supply input (1.8V - 3.3V)
MTCK	20	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
MTDO	21	I/O	GPIO15, ADC2_CH3, TOUCH3, RTC_GPIO13, MTDO, HSPICS0, HS2_CMD, SD_CMD, EMAC_RXD3
GPIO2	22	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
GPIO0	23	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
GPIO4	24	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
VDD_SDIO			
GPIO16	25	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
VDD_SDIO	26	P	1.8V or 3.3V power supply output
GPIO17	27	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
SD_DATA_2	28	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SD_DATA_3	29	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SD_CMD	30	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SD_CLK	31	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS

Name	No.	Type	Function
SD_DATA_0	32	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SD_DATA_1	33	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
VDD3P3_CPU			
GPIO5	34	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
GPIO18	35	I/O	GPIO18, VSPICLK, HS1_DATA7
GPIO23	36	I/O	GPIO23, VSPID, HS1_STROBE
VDD3P3_CPU	37	P	CPU IO power supply input (1.8V - 3.3V)
GPIO19	38	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
GPIO22	39	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
U0RXD	40	I/O	GPIO3, U0RXD, CLK_OUT2
U0TXD	41	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
GPIO21	42	I/O	GPIO21, VSPIHD, EMAC_TX_EN
Analog			
VDDA	43	I/O	Analog power supply (2.3V - 3.6V)
XTAL_N	44	O	External crystal output
XTAL_P	45	I	External crystal input
VDDA	46	P	Digital power supply for PLL (2.3V - 3.6V)
CAP2	47	I	Connects with a 3 nF capacitor and 20 k Ω resistor in parallel to CAP1
CAP1	48	I	Connects with a 10 nF series capacitor to ground

2.3 Power Scheme

ESP32 digital pins are divided into three different power domains:

- VDD3P3_RTC
- VDD3P3_CPU
- VDD_SDIO

VDD3P3_RTC is also the input power supply for RTC and CPU. VDD3P3_CPU is also the input power supply for CPU.

VDD_SDIO connects to the output of an internal LDO, whose input is VDD3P3_RTC. When VDD_SDIO is connected to the same PCB net together with VDD3P3_RTC; the internal LDO is disabled automatically.

The internal LDO can be configured as 1.8V, or the same voltage as VDD3P3_RTC. It can be powered off via software to minimize the current of Flash/SRAM during the Deep-sleep mode.

Note:

It is required that the power supply of VDD3P3_RTC, VDD3P3_CPU and analog must be stable before the pin CHIP_PU is set at high level.

2.4 Strapping Pins

ESP32 has 6 strapping pins:

- MTDI/GPIO12: internal pull-down
- GPIO0: internal pull-up
- GPIO2: internal pull-down
- GPIO4: internal pull-down
- MTDO/GPIO15: internal pull-up
- GPIO5: internal pull-up

Software can read the value of these 6 bits from the register "GPIO_STRAPPING".

During the chip power-on reset, the latches of the strapping pins sample the voltage level as strapping bits of "0" or "1", and hold these bits until the chip is powered down or shut down. The strapping bits configure the device boot mode, the operating voltage of VDD_SDIO and other system initial settings.

Each strapping pin is connected with its internal pull-up/pull-down during the chip reset. Consequently, if a strapping pin is unconnected or the connected external circuit is high-impedance, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values, users can apply the external pull-down/pull-up resistances, or apply the host MCU's GPIOs to control the voltage level of these pins when powering on ESP32.

After reset, the strapping pins work as the normal functions pins.

Refer to Table 2 for detailed boot modes configuration by strapping pins.

Table 2: Strapping Pins

Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3V		1.8V	
MTDI	Pull-down	0		1	
Bootling Mode					
Pin	Default	SPI Boot		Download Boot	
GPIO0	Pull-up	1		0	
GPIO2	Pull-down	Don't-care		0	
Debugging Log on U0TXD During Bootling					
Pin	Default	U0TXD Toggling		U0TXD Silent	
MTDO	Pull-up	1		0	
Timing of SDIO Slave					
Pin	Default	Falling-edge Input Falling-edge Output	Falling-edge Input Rising-edge Output	Rising-edge Input Falling-edge Output	Rising-edge Input Rising-edge Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

Note:

Firmware can configure register bits to change the setting of "Voltage of Internal LDO (VDD_SDIO)" and "Timing of SDIO Slave" after bootling.

3. Functional Description

This chapter describes the functions implemented in ESP32.

3.1 CPU and Memory

3.1.1 CPU

ESP32 contains two low-power Xtensa® 32-bit LX6 microprocessors with the following features.

- 7-stage pipeline to support the clock frequency of up to 240 MHz
- 16/24-bit Instruction Set provides high code-density
- Support Floating Point Unit
- Support DSP instructions, such as 32-bit Multiplier, 32-bit Divider, and 40-bit MAC
- Support 32 interrupt vectors from about 70 interrupt sources

The dual CPUs interface through:

- Xtensa RAM/ROM Interface for instruction and data
- Xtensa Local Memory Interface for fast peripheral register access
- Interrupt with external and internal sources
- JTAG interface for debugging

3.1.2 Internal Memory

ESP32's internal memory includes:

- 448 KBytes ROM for booting and core functions
- 520 KBytes on-chip SRAM for data and instruction
- 8 KBytes SRAM in RTC, which is called RTC SLOW Memory and can be used for co-processor accessing during the Deep-sleep mode
- 8 KBytes SRAM in RTC, which is called RTC FAST Memory and can be used for data storage and main CPU during RTC Boot from the Deep-sleep mode
- 1 Kbit of EFUSE, of which 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including Flash-Encryption and Chip-ID

3.1.3 External Flash and SRAM

ESP32 supports 4 x 16 MBytes of external QSPI Flash and SRAM with hardware encryption based on AES to protect developer's programs and data.

ESP32 accesses external QSPI Flash and SRAM by the high-speed caches

- Up to 16 MBytes of external Flash are memory mapped into the CPU code space, supporting 8-bit, 16-bit and 32-bit access. Code execution is supported.

3.4.2 2.4 GHz Transmitter

The 2.4 GHz transmitter up-converts the quadrature baseband signals to the 2.4 GHz RF signal, and drives the antenna with a high powered Complementary Metal Oxide Semiconductor (CMOS) power amplifier. The use of digital calibration further improves the linearity of the power amplifier, enabling state-of-the-art performance of delivering +20.5 dBm of average power for 802.11b transmission and +17 dBm for 802.11n transmission. Additional calibrations are integrated to cancel any imperfections of the radio, such as:

- Carrier leakage
- I/Q phase matching
- Baseband nonlinearities
- RF nonlinearities
- Antenna matching

These built-in calibration routines reduce the amount of time and required for product test and make test equipment unnecessary.

3.4.3 Clock Generator

The clock generator generates quadrature 2.4 GHz clock signals for the receiver and transmitter. All components of the clock generator are integrated on the chip, including all inductors, varactors, filters, regulators and dividers. The clock generator has built-in calibration and self test circuits. Quadrature clock phases and phase noise are optimized on-chip with patented calibration algorithms to ensure the best performance of the receiver and transmitter.

3.5 Wi-Fi

ESP32 implements TCP/IP, full 802.11 b/g/n/e/i WLAN MAC protocol, and Wi-Fi Direct specification. It supports Basic Service Set (BSS) STA and SoftAP operations under the Distributed Control Function (DCF) and P2P group operation compliant with the latest Wi-Fi P2P protocol.

Passive or active scanning, as well as the P2P discovery procedure are performed autonomously when initiated by appropriate commands. Power management is handled with minimum host interaction to minimize active duty period.

3.5.1 Wi-Fi Radio and Baseband

The ESP32 Wi-Fi Radio and Baseband support the following features:

- 802.11b and 802.11g data-rates
- 802.11n MCS0-7 in both 20 MHz and 40 MHz bandwidth
- 802.11n MCS32
- 802.11n 0.4 μ S guard-interval
- Data-rate up to 150 Mbps
- Receiving STBC 2x1
- Up to 21 dBm transmitting power
- Adjustable transmitting power

- Antenna diversity and selection (software-managed hardware)

3.5.2 Wi-Fi MAC

The ESP32 Wi-Fi MAC applies low level protocol functions automatically as follows:

- Request To Send (RTS), Clear To Send (CTS) and Acknowledgement (ACK/BA)
- Fragmentation and defragmentation
- Aggregation AMPDU and AMSDU
- WMM, U-APSD
- 802.11 e: QoS for wireless multimedia technology
- CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4) and CRC
- Frame encapsulation (802.11h/RFC 1042)
- Automatic beacon monitoring/scanning

3.5.3 Wi-Fi Firmware

The ESP32 Wi-Fi Firmware provides the following functions:

- Infrastructure BSS Station mode / P2P mode / softAP mode support
- P2P Discovery, P2P Group Owner, P2P Group Client and P2P Power Management
- WPA/WPA2-Enterprise and WPS driver
- Additional 802.11i security features such as pre-authentication and TSN
- Open interface for various upper layer authentication schemes over EAP such as TLS, PEAP, LEAP, SIM, AKA or customer specific
- Clock/power gating combined with 802.11-compliant power management dynamically adapted to current connection condition providing minimal power consumption
- Adaptive rate fallback algorithm sets the optimal transmission rate and transmit power based on actual Signal Noise Ratio (SNR) and packet loss information
- Automatic retransmission and response on MAC to avoid packet discarding on slow host environment

3.5.4 Packet Traffic Arbitration (PTA)

ESP32 has a configurable Packet Traffic Arbitration (PTA) that provides flexible and exact timing Bluetooth co-existence support. It is a combination of both Frequency Division Multiplexing (FDM) and Time Division Multiplexing (TDM), and coordinates the protocol stacks.

- It is preferable that Wi-Fi works in the 20 MHz bandwidth mode to decrease its interference with BT.
- BT applies AFH (Adaptive Frequency Hopping) to avoid using the channels within Wi-Fi bandwidth.
- Wi-Fi MAC limits the time duration of Wi-Fi packets, and does not transmit the long Wi-Fi packets by the lowest data-rates.
- Normally BT packets are of higher priority than normal Wi-Fi packets.
- Protect the critical Wi-Fi packets, including beacon transmission and receiving, ACK/BA transmission and receiving.

3.7 RTC and Low-Power Management

With the advanced power management technologies, ESP32 can switch between different power modes (see Table 4).

- Power mode
 - Active mode: The chip radio is powered on. The chip can receive, transmit, or listen.
 - Modem-sleep mode: The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth base-band and radio are disabled.
 - Light-sleep mode: The CPU is paused. The RTC and ULP-coprocessor are running. Any wake-up events (MAC, host, RTC timer, or external interrupts) will wake up the chip.
 - Deep-sleep mode: Only RTC is powered on. Wi-Fi and Bluetooth connection data are stored in RTC memory. The ULP-coprocessor can work.
 - Hibernation mode: The internal 8MHz oscillator and ULP-coprocessor are disabled. The RTC recovery memory are power-down. Only one RTC timer on the slow clock and some RTC GPIOs are active. The RTC timer or the RTC GPIOs can wake up the chip from the Hibernation mode.
- Sleep Pattern
 - Association sleep pattern: The power mode switches between the active mode and Modem-sleep/Light-sleep mode during this sleep pattern. The CPU, Wi-Fi, Bluetooth, and radio are woken up at predetermined intervals to keep Wi-Fi/BT connections alive.
 - ULP sensor-monitored pattern: The main CPU is in the Deep-sleep mode. The ULP co-processor does sensor measurements and wakes up the main system, based on the measured data from sensors.

Table 4: Functionalities Depending on the Power Modes

Power mode	Active	Modem-sleep	Light-sleep	Deep-sleep	Hibernation
Sleep pattern	Association sleep pattern			ULP sensor-monitored pattern	-
CPU	ON	PAUSE	ON	OFF	OFF
Wi-Fi/BT base-band and radio	ON	OFF	OFF	OFF	OFF
RTC	ON	ON	ON	ON	OFF
ULP co-processor	ON	ON	ON	ON/OFF	OFF

The power consumption varies with different power modes/sleep patterns and work status of functional modules (see Table 5).

Table 5: Power Consumption by Power Modes

Power mode	Description	Power consumption
Active (RF working)	Wi-Fi Tx packet 13 dBm ~ 21 dBm	160 ~ 260 mA
	Wi-Fi / BT Tx packet 0 dBm	120 mA
	Wi-Fi / BT Rx and listening	80 ~ 90 mA
	Association sleep pattern (by Light-sleep)	0.9 mA@DTIM3, 1.2 mA@DTIM1
Modem-sleep	The CPU is powered on.	Max speed: 20 mA
		Normal speed: 5 ~ 10 mA
		Slow speed: 3 mA
Light-sleep	-	0.8 mA
Deep-sleep	The ULP co-processor is powered on.	0.15 mA
	ULP sensor-monitored pattern	25 μ A @1% duty
	RTC timer + RTC memory	10 μ A
Hibernation	RTC timer only	2.5 μ A

Note:

For more information about RF power consumption, refer to Section 5.3 RF Power Consumption Specifications.

4.12 I2C Interface

ESP32 has two I2C bus interfaces which can serve as I2C master or slave depending on the user's configuration. The I2C interfaces support:

- Standard mode (100 kbit/s)
- Fast mode (400 kbit/s)
- Up to 5 MHz, but constrained by SDA pull up strength
- 7-bit/10-bit addressing mode
- Dual addressing mode

Users can program command registers to control I2C interfaces to have more flexibility.

4.13 I2S Interface

Two standard I2S interfaces are available in ESP32. They can be operated in the master or slave mode, in full duplex and half-duplex communication modes, and can be configured to operate with an 8-/16-/32-/40-/48-bit resolution as input or output channels. BCK clock frequency from 10 kHz up to 40 MHz are supported. When one or both of the I2S interfaces are configured in the master mode, the master clock can be output to the external DAC/CODEC.

Both of the I2S interfaces have dedicated DMA controllers. PDM and BT PCM interfaces are supported.

4.14 Infrared Remote Controller

The infrared remote controller supports eight channels of infrared remote transmission and receiving. Through programming the pulse waveform, it supports various infrared protocols. Eight channels share a 512 x 32-bit block of memory to store the transmitting or receiving waveform.

4.15 Pulse Counter

The pulse counter captures pulse and counts pulse edges through seven modes. It has 8 channels; each channel captures four signals at a time. The four input signals include two pulse signals and two control signals. When the counter reaches a defined threshold, an interrupt is generated.

4.16 Pulse Width Modulation (PWM)

The Pulse Width Modulation (PWM) controller can be used for driving digital motors and smart lights. The controller consists of PWM timers, the PWM operator and a dedicated capture sub-module. Each timer provides timing in synchronus or independent form, and each PWM operator generates the waveform for one PWM channel. The dedicated capture sub-module can accurately capture external timing events.

4.17 LED PWM

The LED PWM controller can generate 16 independent channels of digital waveforms with the configurable periods and configurable duties.

The 16 channels of digital waveforms operate at 80 MHz APB clock, among which 8 channels have the option of using the 8 MHz oscillator clock. Each channel can select a 20-bit timer with configurable counting range and its accuracy of duty can be up to 16 bits with the 1 ms period.

The software can change the duty immediately. Moreover, each channel supports step-by-step duty increasing or decreasing automatically. It is useful for the LED RGB color gradient generator.

4.18 Serial Peripheral Interface (SPI)

ESP32 features three SPIs (SPI, HSPI and VSPI) in slave and master modes in 1-line full-duplex and 1/2/4-line half-duplex communication modes. These SPIs also support the following general-purpose SPI features:

- 4 timing modes of the SPI format transfer that depend on the polarity (POL) and the phase (PHA)
- up to 80 MHz and the divided clocks of 80 MHz
- up to 64-Byte FIFO

All SPIs can also be used to connect to the external Flash/SRAM and LCD. Each SPI can be served by DMA controllers.

4.19 Accelerator

ESP32 is equipped with hardware accelerators of general algorithms, such as AES (FIPS PUB 197), SHA (FIPS PUB 180-4), RSA, and ECC, which support independent arithmetic such as Big Integer Multiplication and Big Integer Modular Multiplication. The maximum operation length for RSA, ECC, Big Integer Multiply and Big Integer Modular Multiplication is 4096 bits.

The hardware accelerators greatly improve operation speed and reduce software complexity. They also support code encryption and dynamic decryption which ensures that codes in the Flash will not be stolen.

5. Electrical Characteristics

Note:

The specifications in this chapter are tested in general condition: $V_{BAT} = 3.3V$, $T_A = 27^\circ C$, unless otherwise specified.

5.1 Absolute Maximum Ratings

Table 7: Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Unit
Input low voltage	V_{IL}	-0.3	$0.25 \times V_{IO}$	V
Input high voltage	V_{IH}	$0.75 \times V_{IO}$	3.3	V
Input leakage current	I_{IL}	-	50	nA
Output low voltage	V_{OL}	-	$0.1 \times V_{IO}$	V
Output high voltage	V_{OH}	$0.8 \times V_{IO}$	-	V
Input pin capacitance	C_{pad}	-	2	pF
VDDIO	V_{IO}	1.8	3.3	V
Maximum drive capability	I_{MAX}	-	12	mA
Storage temperature range	T_{STR}	-40	150	$^\circ C$

5.2 Recommended Operating Conditions

Table 8: Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit
Battery regulator supply voltage	V_{BAT}	2.8	3.3	3.6	V
I/O supply voltage	V_{IO}	1.8	3.3	3.6	V
Operating temperature range	T_{OPR}	-40	-	125	$^\circ C$
CMOS low level input voltage	V_{IL}	0	-	$0.3 \times V_{IO}$	V
CMOS high level input voltage	V_{IH}	$0.7 \times V_{IO}$	-	V_{IO}	V
CMOS threshold voltage	V_{TH}	-	$0.5 \times V_{IO}$	-	V