



Tecnológico de Monterrey

Definición de Proyecto Reto

Sebastián Hernández Peredo Vieyra	A00838407
Nicholas Losen Gereda	A00837190
Grecia Estefania Archuleta Rivera	A00840745
Andres Patricio Sanchez Chavez	A00841874

11 feb 2025

Materia: Implementación de internet de las cosas

Elsa Torres

Nombre de Proyecto: MediSense

Introducción.....	1
Definición de problema.....	1
Objetivo.....	1
ODS impactados.....	2
Definición de requerimientos.....	2
Descripción del proyecto.....	3
Diseño inicial del proyecto.....	4
Diseño inicial del dashboard.....	5
Modelo Entidad Relación.....	5
Mapeo ERD (Modelo Relacional).....	7
Maqueta.....	7
Software.....	9
Implementación de IA.....	11
Conclusión.....	12
Código general.....	12

Introducción

MediSense es un sistema hospitalario inteligente que integra sensores y tecnologías de inteligencia artificial para automatizar la gestión de inventarios, el monitoreo ambiental y el seguimiento de pacientes en tiempo real. A través de sensores que controlan temperatura, humedad, movimiento y signos vitales, así como mediante cámaras y un chatbot con IA para actualizar inventarios de manera automática, MediSense optimiza el uso de recursos, reduce errores humanos y asegura el almacenamiento adecuado de suministros médicos. Además, proporciona alertas para el reabastecimiento y la detección de riesgos ambientales, mejorando la eficiencia hospitalaria y la calidad de la atención médica.

Definición de problema

Los hospitales enfrentan importantes desafíos para gestionar adecuadamente los suministros médicos, realizar un seguimiento eficiente de los pacientes y mantener condiciones ambientales óptimas en áreas de almacenamiento. Las ineficiencias en estos procesos, como actualizaciones manuales de inventarios o la supervisión continua requerida por el personal, pueden provocar escasez de

medicamentos, deterioro de insumos sensibles y retrasos en la atención al paciente, afectando tanto la operación hospitalaria como la seguridad de los pacientes.

Objetivo

Desarrollar e implementar un sistema inteligente que integre sensores e inteligencia artificial para facilitar la gestión de inventarios, el monitoreo ambiental y el seguimiento de pacientes, mejorando así la eficiencia operativa del hospital, garantizando el control preciso de los suministros médicos y fortaleciendo la calidad de la atención sanitaria.

ODS impactados

3 - Salud y Bienestar: Consideramos que nuestros sensores realizarán un impacto en este ODS, ya que al cambiar la misma infraestructura de los hospitales con los cuales iríamos a trabajar, mejorando la eficacia de estos mismos

9 - Industria, Innovación e Infraestructura: Dado que los cambios propuestos requieren de una instalación dentro de las facilidades, esto se tendría que dar a cabo con un cambio hacia la infraestructura de los lugares donde trabajáramos.



Definición de requerimientos

Requisitos funcionales:

- Monitorización en tiempo real del paciente: Usa sensores para rastrear la presencia del paciente y sus signos vitales.
- Gestión automatizada del inventario: Cámaras con IA capturan imágenes de los suministros y actualizan los niveles de existencias.
- Control ambiental: Sensores y actuadores regulan la temperatura y la humedad en las áreas de almacenamiento médico.

- Alertas automáticas: Notifica al personal sobre escasez de inventario, condiciones críticas de pacientes y riesgos ambientales.
- Interacción con el usuario: Un chatbot con IA permite añadir o eliminar artículos del inventario manualmente.

Requisitos no funcionales:

- Escalabilidad: Soporta la integración de más sensores y dispositivos.
- Fiabilidad: Ofrece datos precisos y con mínimo tiempo de inactividad.
- Seguridad: Protege los datos de pacientes e inventario, accesibles sólo por personal autorizado.
- Usabilidad: La interfaz es intuitiva y fácil de usar por el personal hospitalario.
- Eficiencia: Procesa y actualiza los datos en tiempo real con baja latencia.

Descripción del proyecto

Equipar un hospital con un sistema inteligente de monitoreo y gestión automatizada para optimizar el acceso a información crítica, mejorar la administración de suministros y fortalecer la atención al paciente en tiempo real. En este caso se incluirían 4 sensores, 3 actuadores y se usaría la cámara de un smartphone para la implementación de la IA.

Sensores

- Sensor de pulso cardiaco (ARD-366): Monitoreo del pulso cardíaco del paciente en tiempo real, alertando sobre anomalías cuando la frecuencia cardíaca es inferior a 60 bpm o superior a 100 bpm.
- Sensor de humedad y temperatura (DHT11): Regular la zona de medicamentos e inventario, activando un ventilador encargado de su regulación de temperatura. Se considera un rango óptimo de temperatura entre 22 °C y 25 °C.
- Sensor presión (MF01): Sensor que detecta presencia humana en camillas, mostrando así un mapa del hospital con las que se encuentran disponibles. Debe de estar programado de tal forma que al detectar una presión de 30 Kg, la camilla en la que esta se considera ocupada.

- Sensor de código autorizado (RFID): Proporciona un acceso seguro a las zonas restringidas, concretamente al almacén. Además mantiene un registro de los usuarios con acceso.

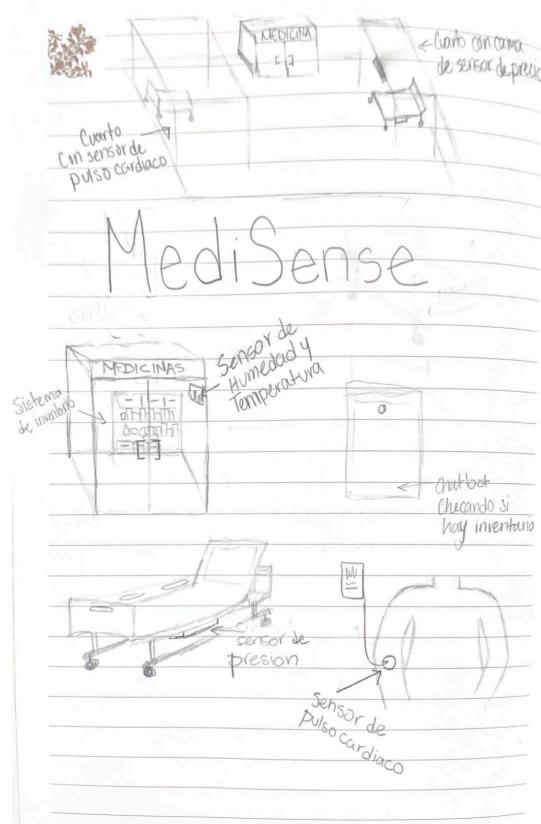
Actuadores

- Ventilador: Se activa cuando se detectan niveles de humedad y temperatura anormales en el inventario. Ayuda a regular el ambiente del mismo para asegurar el almacenamiento seguro y eficiente de los medicamentos.
- Luces led: Se encienden para indicar visualmente niveles anormales que detecten los sensores.
- Buzzer: Activa una alarma indicando que se ha detectado un pulso cardíaco muy alto o muy bajo en los pacientes.

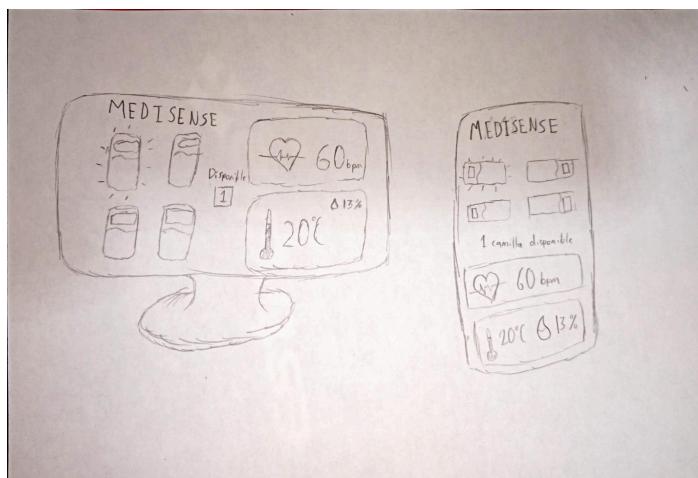
Inteligencia artificial

- Se implementará una inteligencia artificial capaz de llevar control sobre el inventario, indicándonos mediante fotografías la cantidad de medicamentos relacionados con la fotografía.

Diseño inicial del proyecto



Diseño inicial del dashboard



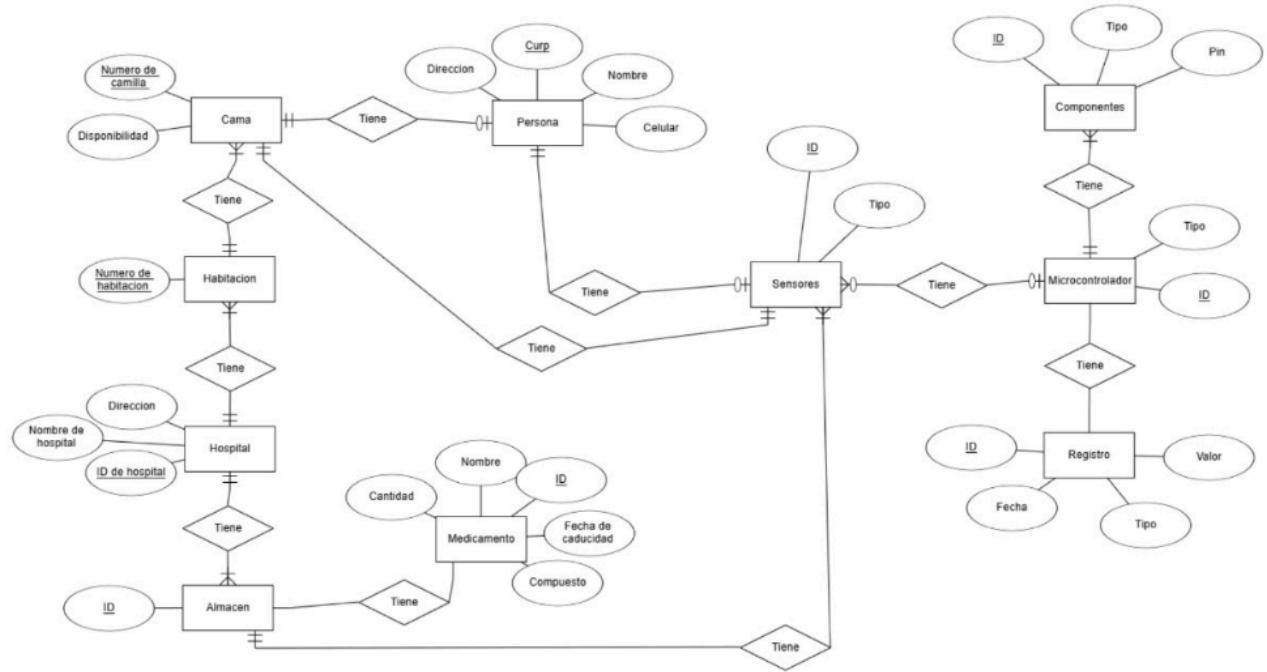
Modelo Entidad Relación

El proyecto consta de las siguientes entidades con sus respectivos atributos:

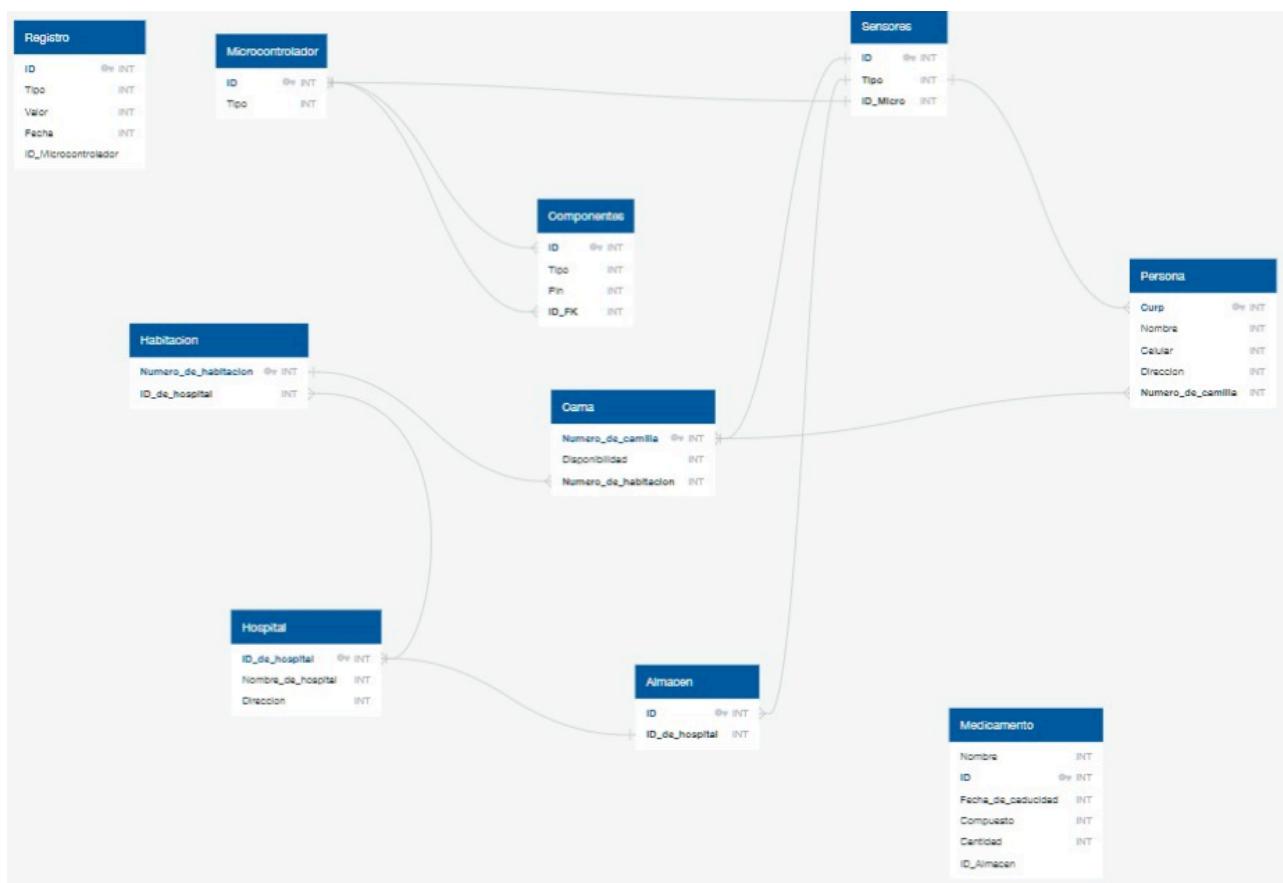
- Cama
 - Número de camilla
 - Disponibilidad

- Habitación
 - Número de habitación
- Hospital
 - Nombre de hospital
 - Dirección
 - ID de hospital
- Almacén
 - ID
- Medicamento
 - Nombre
 - ID
 - Cantidad
 - Fecha de caducidad
 - Compuesto
- Persona
 - CURP
 - Nombre
 - Dirección
 - Celular
- Sensores
 - ID
 - Tipo
- Microcontrolador
 - ID
 - Tipo
- Componentes
 - ID
 - Tipo
 - Pin
- Registro
 - ID
 - Tipo
 - Valor
 - Fecha

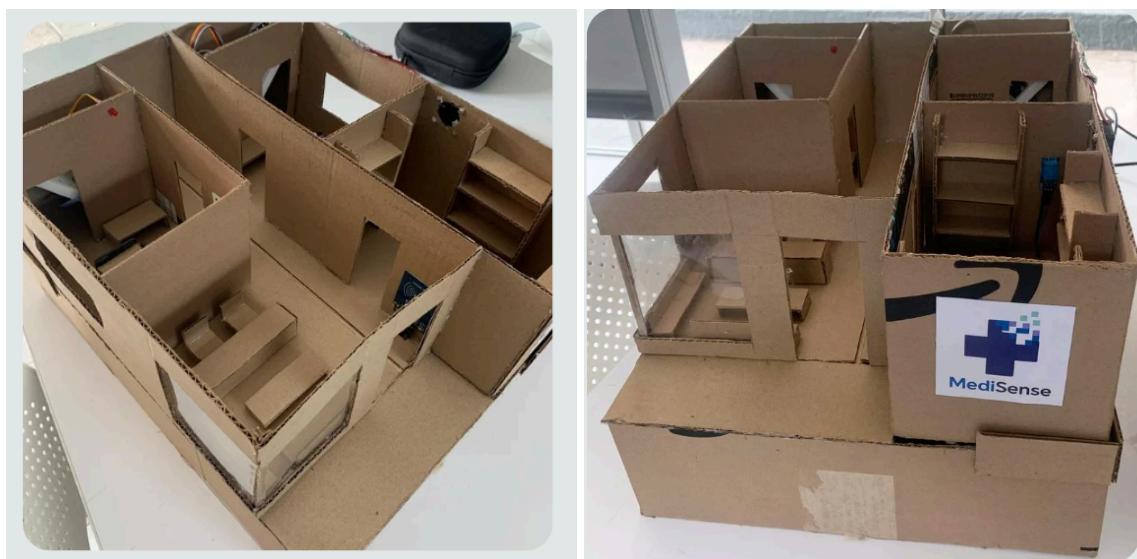
Las relaciones entre las entidades se observan en el siguiente diagrama ER:

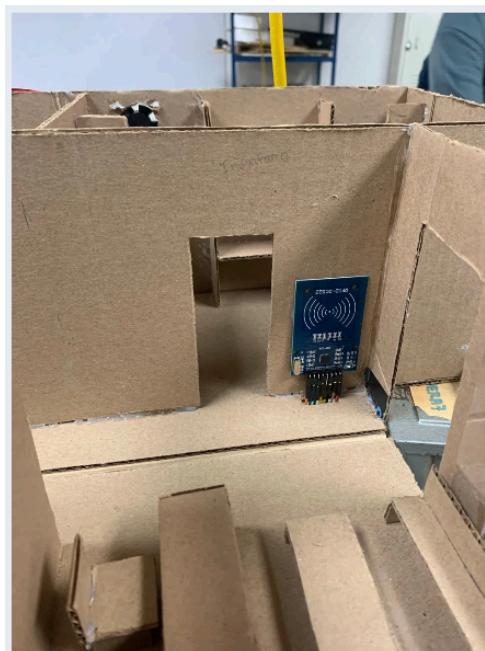


Mapeo ERD (Modelo Relacional)



Maqueta





La maqueta consta de un hospital de 2 cuartos, un almacén y una recepción. En cada cuarto hay una cama y en cada una de ellas hay un sensor, uno de pulso cardiaco y el otro de presión. Cada cuarto tiene su baño, los cuales se usaron para esconder la circuitería. En el cuarto donde está el sensor de presión hay un led y un buzzer, para indicar cuando se detectan niveles anormales de frecuencia cardiaca. En el almacén está el sensor de humedad y temperatura, así como el ventilador, el cual se activa dependiendo de las condiciones ambientales. Aquí también hay unas repisas donde se guardan los medicamentos y se pueden identificar con la cámara del smartphone utilizando IA. En la parte de afuera de la puerta que da al almacén está el sensor de código autorizado, el cual se usa para un acceso controlado.



Software

Utilizando Arduino, se programaron varios sensores para crear un sistema de monitoreo integrado, cuya visualización se gestiona desde un cuadro de mandos desarrollado en Oracle APEX mediante la herramienta App Builder. Este sistema permite supervisar de forma centralizada el pulso del paciente, la ocupación de la cama, la temperatura, la humedad y el estado de los dispositivos.

Se implementaron sensores específicos para cada función: uno de pulso para medir la frecuencia cardíaca, otro de temperatura y humedad para controlar las condiciones del área de almacenamiento, y un sensor de presión para detectar si la

cama está ocupada. Además, se incluyó control de acceso y monitoreo del estado de dispositivos como una luz LED y un zumbador, los cuales se activan automáticamente cuando se detecta un pulso fuera del rango normal.

Las placas Arduino están programadas para leer continuamente los datos de los sensores y enviarlos a la base de datos. Oracle APEX permite presentar esta información en una aplicación web responsive y fácil de usar, que muestra datos clave como la frecuencia cardíaca, el estado de la cama, condiciones ambientales y el estado de los dispositivos. El panel también incluye funciones interactivas para facilitar la navegación y gestión del entorno hospitalario de forma eficiente.

Monitor serial enviando información de los sensores a Oracle Apex:

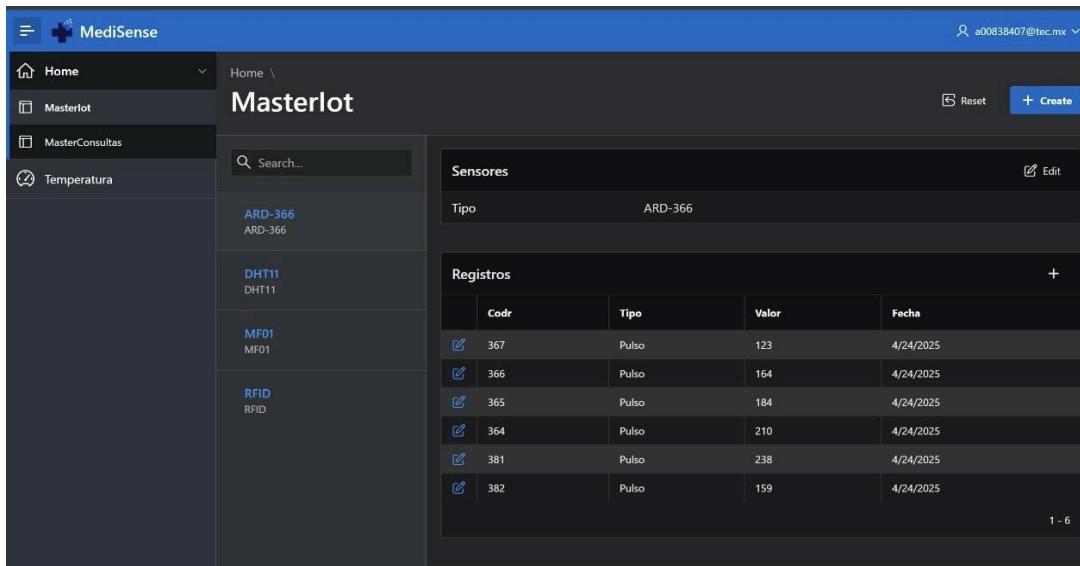


The screenshot shows a terminal window titled "Serial Monitor". The message area contains the following text:

```
Output Serial Monitor X
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM8')
Connection: close
Conexión cerrada
Error leyendo del sensor DHT!
Lectura Analogica = 14
Enviendo GET a: /pls/apox/a00838407/tcc/sendreq/senddata?pcodes=41&ptipo=Fuerza&pvalor=0
ETag: "qVs3YUc6BSrgtaXtpukuhpCt+Ft7eV2ohxfkpBSp0q2GVkQ0XxeG6gn8uCIQwchxO0Z6zZulpjCpE5Diveg=-"
Strict-Transport-Security: max-age=31536000
X-Content-Type-Options: nosniff
Content-Length: 0
Date: Mon, 28 Apr 2025 03:33:20 GMT
Connection: close
Conexión cerrada
⑧ Error leyendo del sensor DHT!
```

Bottom right corner of the window displays: Ln 12, Col 44 NodeMCU 1.0 (ESP-12E Module) on COM8

Base de datos de los sensores:



The screenshot shows a web-based application interface for "MediSense". The left sidebar has navigation items: Home, Masterlot (selected), MasterConsultas, and Temperatura. The main content area is titled "Masterlot". It shows a "Sensores" section with a table:

Tipo	ARD-366
Registros	ARD-366

Below this is a "Registros" table:

Codr	Tipo	Valor	Fecha
367	Pulso	123	4/24/2025
366	Pulso	164	4/24/2025
365	Pulso	184	4/24/2025
364	Pulso	210	4/24/2025
381	Pulso	238	4/24/2025
382	Pulso	159	4/24/2025

Bottom right corner of the table displays: 1 - 6

Dashboard donde se despliega la información en tiempo real:

The screenshot shows a dashboard titled "DashBoard" from the MediSense application. The interface is divided into several sections, each displaying a different metric with its current value and a status indicator (red dot). The sections include:

- Pulso:** 123 BPM
- Temperatura:** 25 °C
- Estado de Cama:** Libre (green)
- Humedad:** 51 %
- Último Acceso:** Persona 1
- Estado de led:** Off (red)
- Estado de buzzer:** Off (red)
- Estado de Ventilador:** Off (red)

Implementación de IA

MediSense utiliza inteligencia artificial a través del modelo Gemini de Google para automatizar el proceso de conteo de medicamentos en un almacén hospitalario. Cuando el usuario toma una foto del cuarto de almacenamiento, la imagen se envía a Gemini, que analiza el contenido visual e identifica las cajas o frascos de medicamentos presentes. Gracias a su tecnología integrada de reconocimiento de texto (OCR), también puede leer etiquetas y nombres para registrar con precisión qué productos hay y cuántos. Toda esta información se interpreta automáticamente y se actualiza en la base de datos del hospital sin necesidad de intervención manual. Esto no solo ahorra tiempo, sino que también reduce errores humanos y permite tener un control mucho más eficiente del inventario médico.



Conclusión

El proyecto MediSense demuestra cómo la integración de sensores, microcontroladores, inteligencia artificial y plataformas de desarrollo web como Oracle APEX puede transformar la infraestructura hospitalaria, mejorando significativamente la eficiencia y calidad de la atención médica. A través de un sistema inteligente de monitoreo en tiempo real de signos vitales, ocupación de camas, condiciones ambientales y control de inventarios, MediSense ofrece una solución innovadora para reducir errores humanos, optimizar el uso de recursos y garantizar la seguridad de los pacientes.

El desarrollo de hardware, software y bases de datos fue realizado de manera efectiva, permitiendo la creación de un entorno hospitalario más automatizado, seguro y reactivo a las necesidades críticas. Además, el proyecto contribuye a los Objetivos de Desarrollo Sostenible, específicamente en los apartados de Salud y Bienestar, e Industria, Innovación e Infraestructura. MediSense representa un paso importante hacia la modernización de los servicios hospitalarios a través de la implementación práctica de tecnologías del Internet de las Cosas (IoT).

Código general

```
#include <DHT.h>

#define PRESION_PIN D3 // Sensor de presión (interruptor)
#define PULSO_PIN A0 // Sensor de pulso (analógico)
#define DHT_PIN D0 // Sensor de temperatura y humedad (DHT11 o DHT22)
#define BUZZER_PIN D6 // Buzzer
#define RELAY_PIN D7 // Relay para ventilador
#define LED_PIN D8 // LED indicador del ventilador

#define DHTTYPE DHT11 // Cambiar a DHT22 si es necesario
DHT dht(DHT_PIN, DHTTYPE);

const int umbralPulsoCritico = 700; // Umbral para detectar un pico de pulso (ajustar según el sensor)
const int maxMuestra = 100; // Cantidad de muestras para calcular BPM
unsigned long tiempoUltimoLatido = 0; // Para calcular el tiempo entre latidos
int latidosPorMinuto = 0; // Para almacenar el BPM

// Variables para detectar picos de pulso
int pulsoAnterior = 0;
int pulsoActual = 0;
bool pulsoDetectado = false;

void setup() {
    Serial.begin(115200);

    // Configurar pines de sensores y actuadores
    pinMode(PRESION_PIN, INPUT_PULLUP);
    pinMode(PULSO_PIN, INPUT);
    pinMode(DHT_PIN, INPUT);
    pinMode(BUZZER_PIN, OUTPUT);
    pinMode(RELAY_PIN, OUTPUT);
```

```

pinMode(LED_PIN, OUTPUT);

digitalWrite(BUZZER_PIN, LOW);
digitalWrite(RELAY_PIN, LOW);
digitalWrite(LED_PIN, LOW);

dht.begin();
}

void loop() {
    // Leer estado del sensor de presión [10]
    int presion = digitalRead(PRESION_PIN);
    if (presion == LOW) {
        Serial.println("⚠ Camilla Ocupada ⚠");
    } else {
        Serial.println("✓ Camilla Libre");
    }

    // Leer el pulso actual (valor analógico)
    pulsoActual = analogRead(PULSO_PIN);
    Serial.print("Pulso actual: ");
    Serial.print(pulsoActual);
    Serial.print(", ");

    // Detectar el pico del pulso y calcular el BPM [11]
    if (pulsoActual > umbralPulsoCritico && pulsoAnterior <= umbralPulsoCritico) {
        // Se detecta un pico (cuando el valor supera el umbral y el anterior no lo superaba)
        pulsoDetectado = true;
        latidosPorMinuto++; // Contamos el latido
        Serial.println("Pulso Detectado");
        digitalWrite(BUZZER_PIN, HIGH); // Activar buzzer si el pulso es crítico
    } else {
        pulsoDetectado = false;
        digitalWrite(BUZZER_PIN, LOW); // Desactivar buzzer si no es crítico
    }

    // Mostrar BPM cada 5 segundos
    if (millis() % 5000 == 0) {
        Serial.print("Latidos por minuto (BPM): ");
        Serial.println(latidosPorMinuto);
        latidosPorMinuto = 0; // Reiniciar el contador de BPM para el siguiente ciclo
    }

    // Leer temperatura y humedad [9]
    float temperatura = dht.readTemperature(); // Temperatura en Celsius
    float humedad = dht.readHumidity(); // Humedad relativa (%)

    if (isnan(temperatura) || isnan(humedad)) {
        Serial.println("Error al leer el sensor DHT. Revisar conexiones.");
    } else {
        Serial.print("Temperatura: ");
        Serial.print(temperatura);
        Serial.print(" °C, Humedad: ");
        Serial.print(humedad);
        Serial.println(" %");
    }

    // Comprobar si la temperatura es anormal y activar ventilador
    if (temperatura > 38.0 || temperatura < 20.0) { // Valores de referencia

```

```
Serial.println("⚠ Temperatura Anormal, Activando Ventilador y LED ⚠");
digitalWrite(RELAY_PIN, HIGH); // Prender ventilador
digitalWrite(LED_PIN, HIGH); // Prender LED indicador
} else {
    digitalWrite(RELAY_PIN, LOW); // Apagar ventilador
    digitalWrite(LED_PIN, LOW); // Apagar LED indicador
}

// Guardar el pulso actual como el anterior para la siguiente iteración
pulsoAnterior = pulsoActual;

delay(1000); // Esperar 1 segundo antes de la próxima lectura
}
```