



Recopilatorio de Práctica Calificada 1 (PC1), Examen Parcial y Práctica Calificada 2 (PC2)

Desarrollo de Soluciones IoT (Universidad Peruana de Ciencias Aplicadas)



Escanea para abrir en Studocu

SI572 – DESARROLLO DE SOLUCIONES IOT
PRÁCTICA CALIFICADA 1
2024-2



Profesor: Velásquez Núñez, Ángel Augusto

Duración: 100 minutos

Indicaciones:

1. El examen consta de 4 preguntas en base a un caso, y tendrá 100 minutos para resolverlas.
2. Las preguntas son en relación a un Caso y la entrega de su respuesta es a través de envío del archivo de PowerPoint adjunto, conteniendo su solución. Utilice el documento de PowerPoint **upc-pre-202402-si572-pc1-file_v1.pptx** para responder, tanto a nivel de texto para las preguntas de redacción como a nivel de las preguntas relacionadas con diagramas. Coloque contenido en cada hoja de respuesta según el título e indicaciones.

Enunciado:

Caso Smart Agro, inc.

Su cliente, Smart Agro, inc. fue fundada en 2008, por un grupo de inversionistas con la finalidad de desarrollar en los desiertos del norte del país una gran zona agrícola. Beneficiándose de el uso de tecnologías innovadoras y sostenibles.

Para ello se embarcaron en un proyecto de Precision Agriculture con el propósito de ahorrar costes y tiempo en la producción agropecuaria, con un uso óptimo de los recursos.

Para lograr sus objetivos de negocio, Smart Agro, inc. ha llegado a implementar unos 32 puntos de monitoreo que incluyen IoT devices como sensores de Temperatura, Humedad y Conductividad del Terreno, todos ellos con una precisión de +/- 2%. También ha instalado una serie de actuators (actuadores) con un rango de precisión de +/- 10%. Todos puntos de monitoreo están ubicados a una distancia máxima entre nodos de 50 metros, distribuidos en su área de sembrado.

El problema que presentan actualmente es que, debido a cambios de gestión, no se siguió un plan de escalamiento con una visión integral, por lo que en muchos casos, dispositivos instalados en momentos diferentes trabajan con protocolos de comunicación no compatibles entre todos los elementos de la red. Debido a que la solución está implementada en una zona rural, no se cuenta con facilidades para el abastecimiento de energía eléctrica en las áreas de sembrado.

En sus reuniones preliminares, el equipo de Smart Agro, inc. le manifiesta el deseo de desarrollar un proyecto que integre los 32 puntos de monitoreo en el terreno, permitiendo que se puedan comunicar y que se procese la data recolectada on-the-edge (en el Edge Gateway más cercano) para brindar soporte a una toma de decisiones rápida (lo que implica un time-delay no mayor de 100 milisegundos). La información procesada en el Edge Layer se enviaría al cloud para que se ejecuten tareas avanzadas de AI y ML. El objetivo de dicho proceso es generar patrones mejorados para los actuators, lo que conlleve a actualizaciones en el nodo de Edge Computing.

El procesamiento de la data recolectada de los diferentes sensores requiere el procesamiento de algoritmos complejos de cálculo en el nodo de Edge Computing.

La solución debe considerar que hay un equipo de trabajo en campo, conformado por dos supervisores de campo (que alternan en turnos) y 6 operadores de campo. Los supervisores cuentan con tablets y smartphone, mientras que los operadores de campo cuentan con smartphones. Dicho equipo es responsable del monitoreo de la correcta operación de la solución y mantenimiento de los IoT devices y equipos de Edge computing desplegados, por lo deben tener a mano información de los sensores de temperatura, humedad y conductividad para detectar valores atípicos. Otro equipo de trabajo se encuentra en la oficina de la estación central, los cuales cuentan con computadoras de escritorio. Este equipo se encarga de revisar también el estado de operación de la solución, pero además revisa los analíticos generados por los procesos de AI y la revisión de los patrones de actuators identificados

generados por los procesos de ML, para administrar las actualizaciones del nodo de Edge Computing. El equipo de gestión de alto rango desea tener acceso desde sus smartphones a key performance indicators y gráficos de resúmenes analíticos de apoyo a la toma de decisiones sobre la IoT Solution sobre todos los aspectos de la solución, incluyendo resultados de estado de operación, datos procesados, analíticos que proporcionan los componentes de la solución basados en IA y ML.

Los representantes de Smart Agro, Inc. le solicitan que, bajo el marco de los 12 pasos del IoT System Design Steps (ver Anexo A), presente para su siguiente reunión un avance de la definición de requisitos para su propuesta de mejora para el IoT Solution.

Pregunta 1 (4 p.)

Elabore el Definition of system requirements (los requisitos del sistema) en términos de Power Supply (Capacidades de suministro de energía) y restricciones de time-delay.

Pregunta 2 (5 p.)

Elabore el Definition of physical layer requirements (los requisitos para el Physical Layer de la IoT Solution). Considere: a) número y tipos de nodos sensores y actuators; b) Target uncertainty (incertidumbre objetivo), relacionada con las cantidades físicas medidas por cada sensor; c) Target accuracy and precision (Exactitud y precisión objetivo) de los actuadores; d) Processing Power (Esfuerzo computacional) para los algoritmos de procesamiento de datos que se implementarán en el Edge node.

Pregunta 3 (6 p.).

Elabore el Definition of information layer requirements (requisitos de la capa de información). Considere: a) Definición de usuarios finales; b) Definición de número y tipos de servicios que deben proporcionarse a cada usuario final; c) Definición de las necesidades de información integrada para implementar cada servicio.

Pregunta 4 (5 p.)

Elabore un diagrama de Containers de C4 Model para su propuesta de Smart Agro, inc. Explique y sustente sus decisiones de diseño.

Solución: Nota de la PC1 con esta solución = 17.0

Pregunta 1:

Elabore el Definition of system requirements (los requisitos del sistema) en términos de Power Supply (Capacidades de suministro de energía) y restricciones de time-delay

System Requirement	Especificación de requisitos
Capacidades de suministro de energía	Energía Solar: Como Smart Agro, Inc opera en una zona desértica del norte del país, se puede aprovechar la energía solar para alimentar a todos los dispositivos IoT. Por tanto, la instalación de paneles solares en cada punto de monitoreo puede garantizar un suministro constante. Baterías para paneles solares: Para asegurar que haya constantemente suministro de energía durante la noche o en días nublados, se podría agregar baterías solares de Litio de 48V que almacenen energía solar. Estas baterías deben tener suficiente capacidad para mantener a todos los dispositivos operativos durante el tiempo que no haya luz solar.
Restricciones de time-delay	Monitoreo en Tiempo Real: Para que el sistema sea efectivo, es importante que el tiempo de respuesta para procesar y comunicar los datos recolectados del Edge Gateway no debe exceder los 100 milisegundos.

Pregunta 2:

Elabore el Definition of physical layer requirements (los requisitos para el Physical Layer de la IoT Solution). Considere: a) número y tipos de nodos sensores y actuators; b) Target uncertainty (incertidumbre objetivo), relacionada con las cantidades físicas medidas por cada sensor; c) Target accuracy and precision (Exactitud y precisión objetivo) de los actuadores; d) Processing Power (Esfuerzo computacional) para los algoritmos de procesamiento de datos que se implementarán en el Edge node.

Physical Layer Requirement	
Número y tipos de nodos sensores y actuators	<ul style="list-style-type: none">Se debe implementar 32 nodos sensores distribuidos en el campo. Lo sensores especificados son: sensor de temperatura, sensor de humedad y sensor de conductividad.Se implementará actuadores en cada nodo sensor para permitir la respuesta automática a las condiciones ambientales. Los actuadores podrían ser válvulas de riego u otros actuadores.
Target uncertainty (incertidumbre objetivo), relacionada con las cantidades físicas medidas por cada sensor	<ul style="list-style-type: none">Los IoT Devices como el sensor de temperatura, sensor de humedad y sensor de conductividad del terreno deben contar con una precisión de +/- 2%. Lo que permite tener una medición precisa y confiable.
Target accuracy and precision (Exactitud y precisión objetivo) de los actuadores	<ul style="list-style-type: none">La serie de Actuators (actuadores) debe tener un rango de precisión de +/- 10%.
Processing Power (Esfuerzo computacional) para los algoritmos de procesamiento de datos que se implementarán en el Edge node	<ul style="list-style-type: none">El procesamiento de datos recolectados de sensores de temperatura, humedad y conductividad implica la ejecución de algoritmos complejos en los nodos.Se debe implementar algoritmos de generación de datos analíticos generador por la Inteligencia Artificial (IA) y se debe desarrollar algoritmos de revisión de los patrones de los actuators generados por procesos de Machine Learning, con el fin de actualizar el nodo de Edge Computing.Los datos smartphones debe tener un algoritmo capaz de mostrar un Key Performance Indicators y gráficos de resúmenes analíticos, para mejorar la toma de decisiones de la IoT Solution como resultados de estado de operación, datos procesados y analíticos que generan los algoritmos de Inteligencia Artificial y Machine Learning.

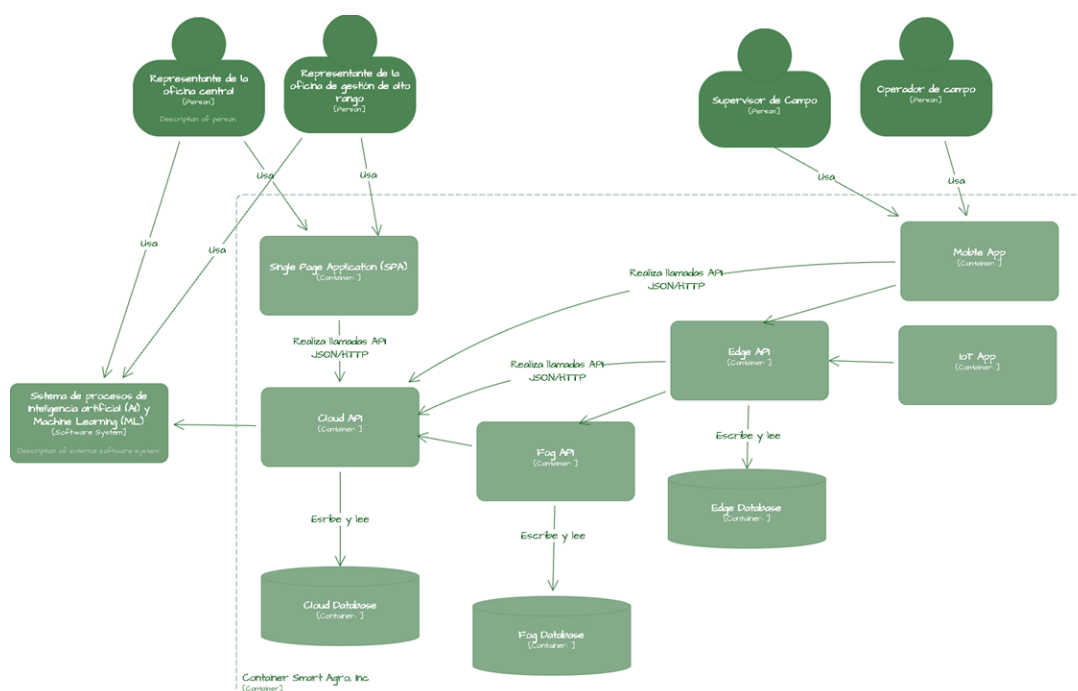
Pregunta 3:

Elabore el Definition of information layer requirements (requisitos de la capa de información). Considere: a) Definición de usuarios finales; b) Definición de número y tipos de servicios que deben proporcionarse a cada usuario final; c) Definición de las necesidades de información integrada para implementar cada servicio.

Information Layer Requirement	Especificación de requisitos
Definición de usuarios finales	<ul style="list-style-type: none"> 2 Supervisores de campo 6 Operadores de campo Equipo de trabajo en la oficina central Equipo de gestión de alto rango
Definición de número y tipos de servicios que deben proporcionarse a cada usuario final	<p>Supervisores y Operadores de Campo:</p> <ul style="list-style-type: none"> Monitoreo en tiempo real de sensores de temperatura, humedad y conductividad. Alertas sobre valores atípicos. Mantenimiento de los IoT Devices y equipos Edge Computing. <p>Equipo de Oficina Central:</p> <ul style="list-style-type: none"> Revisión del estado de operación de la solución. Acceso a analíticos generados por procesos de AI y revisión de patrones de actuadores. <p>Equipo de Gestión de Alto Rango:</p> <ul style="list-style-type: none"> Acceso a indicadores clave de rendimiento (Key Performance) y gráficos analíticos como resultados de estado de operación, datos procesados y analíticos brindados por IA y ML desde sus smartphones.
Definición de las necesidades de información integrada para implementar cada servicio	<ul style="list-style-type: none"> Información procesada de sensores para detectar anomalías. Datos en tiempo real sobre el estado de los dispositivos IoT. Informes analíticos sobre el rendimiento de la solución y patrones de actuadores. Resúmenes analíticos y Key Performance sobre el estado de operación y resultados de la solución IoT.

Pregunta 4:

Pegue aquí el diagrama de contenedores para la IoT Solution, considerando todas las características del caso en su propuesta de solución



SI572 – DESARROLLO DE SOLUCIONES IOT
EXAMEN PARCIAL
2024-2



Profesores: León Baca, Marco Antonio
Velásquez Núñez, Ángel Augusto

Duración: 170 minutos

Indicaciones:

1. El examen consta de 5 preguntas en base a un caso, y tendrá 170 minutos para resolverlas.
2. Las preguntas son en relación con un Caso y la entrega de su respuesta es a través de envío del archivo de PowerPoint adjunto, conteniendo su solución. Utilice el documento de PowerPoint **upc-pre-202402-si572-examen-parcial_v1-file.pptx** para responder, tanto a nivel de texto para las preguntas de redacción como a nivel de las preguntas relacionadas con diagramas. Coloque contenido en cada hoja de respuesta según el título e indicaciones.
3. Puede utilizar como referencia los materiales publicados en el aula virtual, los sitios web referenciados en el enunciado, así como sitios web de documentación de frameworks o tecnologías referenciadas.

Caso Eventify

Eventify (<https://eventify.io/>) es una empresa de gestión de eventos que tiene como objetivo revolucionar la industria de la planificación de eventos. Con un fuerte enfoque en la atención al detalle, un servicio al cliente excepcional y un compromiso con la excelencia, se esfuerzan por crear eventos que se destaquen de la competencia, asegurándose de que cada uno sea memorable y verdaderamente inolvidable.

Son un equipo de profesionales experimentados con una pasión por crear eventos perfectos e inolvidables. La plataforma de eventos está diseñada para simplificar la planificación y ejecución de eventos, lo que le permite concentrarse en lo que más importa: crear experiencias significativas para sus invitados.

Los mejores organizadores de eventos están incorporando las últimas tecnologías de IoT para mejorar su desempeño. Como resultado, el uso de IoT en la planificación de eventos está generando una nueva ola de revolución.

RFID Bands

Entify ofrece una experiencia de los eventos mejorada con tecnología RFID. Desde un ingreso más rápido hasta pagos sin efectivo, las pulseras RFID mejoran la participación y la comodidad de los asistentes. Quieren que sus clientes digan adiós a las largas colas y den la bienvenida a datos más completos y patrocinadores más comprometidos (ver la sección de Referencias).

Smart Food Dispensers

Entify pone disposición de sus clientes Organizadores equipos conocidos como Smart Food Dispenser con control de temperatura. Este equipo permite un control total sobre el inventario del proveedor en un entorno refrigerado y con control de temperatura para una máxima

longevidad, seguridad y frescura del producto. El sistema de refrigeración del dispositivo garantiza que los productos se mantendrán frescos hasta que estén listos para usarse. Estos productos pueden incluir bocadillos, productos nutricionales o sándwiches. Esta máquina dispensadora refrigerada respalda el cumplimiento de las normas regulatorias y garantiza que los productos mantengan su frescura y usabilidad, lo que le permite ahorrar miles de dólares anuales en costos de reemplazo. Entre su configuración se incluye un panel con un módulo lector de RFID RC522, un 4x4 Membrane KeyPad y un display LCD 1602. Adicionalmente cuentan en su interior con 4 Sensores de temperatura DHT-22, junto con 20 Micro Servo Motors para controlar la parte mecánica de dispensación. El sistema de control del dispositivo se basa en un microcontrolador ESP32 (ver sección de Referencias). El dispositivo soporta la conexión a la red WiFi del evento, con el fin de recibir actualizaciones de configuración o proporcionar información sobre el stock de productos para dispensar o notificar cada vez que un asistente al evento hace uso de su RFID Band para consumir un producto. El Dispenser está conectado a una toma de fuente de energía eléctrica estándar de 220 V. Para eventos que ofrecen esta facilidad, Eventify recomienda un Dispenser por cada 50 registrados.

Smart Coffee Makers

La Smart Coffee Maker es una máquina que prepara café muy parecido al café hecho en casa en solo un par de segundos presionando un botón. La máquina está completamente automatizada con puerta modular. Según los requisitos, el asistente puede elegir diferentes sabores de la máquina. Las máquinas son fáciles de mantener y se ven fantásticas para eventos. Tiene un bajo consumo de energía. Al usar esta máquina, puede obtener el sabor del café preparado en casa en el evento, al dar la mezcla adecuada de premezcla y agua caliente. Los asistentes pueden elegir el sabor según su elección y el producto estará listo para servir en un minuto. Es posible configurarlos para ofrecer dos opciones, tres opciones hasta cuatro opciones de tipos de café a los asistentes al evento. También ofrecen una pantalla que se puede personalizar con la marca del evento. Entre su configuración se incluye un panel con un módulo lector de RFID RC522, un 4x4 Membrane KeyPad y un display LCD 1602. Adicionalmente cuentan en su interior con 1 Sensor de temperatura DHT-22, junto con 4 Micro Servo Motors para el módulo de dispensación. El sistema de control del dispositivo se basa en un microcontrolador ESP32 (ver sección de Referencias). El dispositivo soporta la conexión a la red WiFi del evento, con el fin de recibir actualizaciones de configuración o proporcionar información sobre el stock de insumos para dispensar o notificar cada vez que un asistente al evento hace uso de su RFID Band para consumir un producto. El Dispenser está conectado a una toma de fuente de energía eléctrica estándar de 220 V. Para eventos que ofrecen esta facilidad, Eventify recomienda un Coffee Maker por cada 50 registrados.

Event Organizer Platform

El sistema permite al Event Organizer la configuración de información del evento, analíticos, así como la carga masiva de información de personas registradas.

Event Registration and Check-In Platform

Eventify ofrece a las personas registradas en un evento una aplicación móvil que facilita el proceso de check-in. Al momento del ingreso, la persona registrada presenta un código QR generado por la aplicación y es validado en el puesto de control que cuenta con varias estaciones cada una con una aplicación móvil de control de asistencia y un lector de QR. Dicha aplicación también permite la activación y asignación de las RFID Bands disponibles para el evento. Una vez confirmado, la persona registrada pasa a ser asistente al evento, recibiendo a continuación un

RFID Band que le brinda acceso a las salas y facilidades dentro del evento. Eventify recomienda a los organizers una estación en el puesto de control por cada 60 registrados en el evento.

Event Food Partner Management Platform

La solución de Evently incluye una aplicación web y una aplicación móvil para usuarios que sean parte del Partner designado por el Event Organizer para el abastecimiento de insumos y productos para los Smart Food Dispensers y Smart Coffee Makers. Con ello el Food Partner User puede revisar el estado de inventarios de productos e insumos de los Food Dispensers y Coffee Makers distribuidos en las instalaciones del evento, a fin de mantenerlos abastecidos y en operación normal. La plataforma cuenta con un RESTful API para que el Partner pueda integrar la información con sus sistemas internos.

Event Device Maintenance Partner Management Platform

Eventify ofrece a sus proveedores de mantenimiento de equipos y dispositivos una plataforma web y aplicación móvil que brinda acceso a información de estado en tiempo real sobre los dispositivos desplegados en cada evento de los clientes de Eventify, a fin de que cada partner asignado a uno o más eventos cercanos, pueda brindar mantenimiento preventivo antes de colocar un equipo o dispositivo en un evento y atención correctiva durante el evento. La aplicación móvil está destinada a los Agentes de Soporte Técnico, mientras que la aplicación web está orientada a personal en las oficinas de cada partner. La plataforma cuenta con un RESTful API para que el Partner pueda integrar la información con sus sistemas internos.

Eventify desea una configuración de arquitectura que brinde los mejores tiempos de respuesta, estabilidad, tanto a las operaciones que se realizan en cada evento como a las operaciones de gestión centralizada vía cloud para Eventify a nivel global. Ellos le indican que la estación de control de cada evento cuenta con un servidor de procesamiento de datos que centraliza la información del evento, está conectado a la red WiFi del evento, pero también tiene conexión vía internet con la plataforma cloud de Eventify, para envío y sincronización de información relevante.

En los últimos meses, Eventify ha visto un incremento en su demanda y sus clientes recientes suelen organizar eventos presenciales para niveles de asistencia entre 500 y 1000 asistentes.

Los representantes de Eventify le solicitan que, bajo el marco de los 12 pasos del IoT System Design Steps (ver Anexo A), Domain-Driven Design y Object-Oriented Software Design, presente para su siguiente reunión un avance de su propuesta de requisitos para el diseño de la actual IoT Solution.

Pregunta 1 (2 p.)

Elabore el Definition of system requirements (los requisitos del sistema) en términos de Power Supply (Capacidades de suministro de energía) y restricciones de time-delay.

Pregunta 2 (3 p.)

Elabore el Definition of physical layer requirements (los requisitos para el Physical Layer de la IoT Solution). Considere: a) número y tipos de nodos sensores y actuators; b) Target uncertainty (incertidumbre objetivo), relacionada con las cantidades físicas medidas por cada sensor; c) Target accuracy and precision (Exactitud y precisión objetivo) de los actuadores; d)

Processing Power (Esfuerzo computacional) para los algoritmos de procesamiento de datos que se implementarán en el Edge node.

Pregunta 3 (5 p.).

Elabore el Definition of information layer requirements (requisitos de la capa de información). Considere: a) Definición de usuarios finales; b) Definición de número y tipos de servicios que deben proporcionarse a cada usuario final; c) Definición de las necesidades de información integrada para implementar cada servicio.

Pregunta 4 (5 p.)

Elabore un diagrama de Containers de C4 Model para su propuesta de Eventify. Explique y sustente sus decisiones de diseño.

Pregunta 5 (5 p.)

En base al caso, elabore un Class Diagram para el Object-Oriented Embedded Application para el Smart Food Dispenser. Eventify requiere que su diseño sea con nomenclatura en inglés, considere los atributos y métodos principales en las clases. Tome en cuenta también que en base a su diseño, posteriormente otro equipo de desarrollo distinto estará encargado de realizar la implementación en Arduino Sketch con C++ (archivo .ino, junto con los archivos .h y .cpp que sean necesarios). Explique y sustente sus decisiones de diseño.

Referencias

Diagrams

<https://c4model.com/>

<https://www.uml-diagrams.org/class-diagrams-overview.html#domain-model-diagram>

Microcontroller

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

RFID

<https://rfid.it/en/content/10-rfid-ics-specs>

<https://www.idcband.com/en-us/blog/how-do-rfid-wristbands-work/>

Components

<https://arduinomodels.info/ky-004-key-switch-module/> <https://www.sunrom.com/p/matrix-keypad-4x4-membrane-type-self-adhesive>

<https://components101.com/wireless/rc522-rfid-module> <https://protosupplies.com/product/lcd1602-16x2-i2c-blue-lcd-display/>

<https://docs.wokwi.com/parts/wokwi-lcd1602> <https://cityos-air.readme.io/docs/4-dht22-digital-temperature-humidity-sensor> <https://docs.wokwi.com/parts/wokwi-dht22>

<https://vishaworld.com/products/sg90-servo-9-gms-mini-micro-servo-motor>

https://cdn.shopify.com/s/files/1/0672/9409/files/Introduction_to_Servo_Motors_Arduino.pdf

Solución: Nota del Examen Parcial con esta solución = 16.0

Pregunta 1:

Elabore el Definition of system requirements (los requisitos del sistema) en términos de Power Supply (Capacidades de suministro de energía) y restricciones de time-delay

System Requirement	Especificación de requisitos
Capacidades de suministro de energía	RFID Bands: : Existen tres tipos de etiquetas para los RFID como pasivo, activo y asistida por batería. El pasivo utiliza la energía transmitida por el lector para obtener energía y no tienen una batería incorporada. Para este caso se usará el RFID con etiqueta pasiva ya que es la mejor solución para eventos a gran escala. La información fue obtenida en la siguiente ruta: https://www.idcband.com/en-us/blog/how-do-rfid-wristbands-work/ Smart Food Dispensers y Smart Coffee Makers: Conectados a una fuente de energía eléctrica estándar de 220V. Estación de control del evento: Debe tener suministro de energía para el servidor de procesamiento de datos y proveer energía para los equipos de red WiFi del evento.
Restricciones de time-delay	RFID Bands: Deben permitir un ingreso más rápido al evento, minimizando el tiempo de lectura y validación, además, debe facilitar pagos sin efectivo con tiempos de respuesta rápidos. Smart Food Dispensers y Smart Coffee Makers: Deben procesar y notificar cada uso de RFID Band para consumir un producto en tiempo real. Event Registration y Check-In Platform: El proceso de validación del código QR y asignación de RFID Band debe ser ágil para manejar el flujo de 60 registrados por estación. Conectividad: La sincronización de datos entre el servidor local del evento y la plataforma cloud de Eventify debe ser en tiempo real. Event Food Partner Management Platform y Event Device Maintenance Partner Management Platform: Deben proporcionar información de estado en tiempo real sobre los dispositivos y niveles de inventario.

Pregunta 2:

Elabore el Definition of physical layer requirements (los requisitos para el Physical Layer de la IoT Solution). Considere: a) número y tipos de nodos sensores y actuators; b) Target uncertainty (incertidumbre objetivo), relacionada con las cantidades físicas medidas por cada sensor; c) Target accuracy and precision (Exactitud y precisión objetivo) de los actuadores; d) Processing Power (Esfuerzo computacional) para los algoritmos de procesamiento de datos que se implementarán en el Edge node.

Physical Layer Requirement	Especificación de requisitos
Número y tipos de nodos sensores y actuators	RFID Bands: Uno por cada asistente registrado, en este caso son entre 500 a 1000 por evento. Smart Food Dispensers: 1 por cada 50 registrados, es decir, entre 10 y 20 por evento. Este incluye 1 módulo lector RFID RC522, 4 sensores de temperatura DHT-22, 20 Micro Servo Motors y 1 Display LCD 1602. Smart Coffee Makers: 1 por cada 50 registrados. Este incluye 1 módulo lector RFID RC522, 1 sensor de temperatura DHT-22, 4 Micro Servo Motors y 1 Display LCD 1602. Estaciones de control de ingreso: 1 por cada 60 registrados. Este incluye 1 Lector de código QR.
Target uncertainty (incertidumbre objetivo), relacionada con las cantidades físicas medidas por cada sensor	Sensores de temperatura DHT-22: Humedad 0-100%RH y Temperatura de -40 °C - 80 °C. Módulo lector RFID RC522: leer datos de etiquetas pasivas que funcionan en 13,56 MHz.
Target accuracy and precision (Exactitud y precisión objetivo) de los actuadores	Micro Servo Motors: Temperatura de funcionamiento: -30 a 60 °C, Grado de Rotación de 180° y la velocidad de funcionamiento a 4,8 V es de 10 sec/60°.

	<p>Display LCD 1602: LCD azul LCD1602 16x2 I2C proporciona una pantalla LCD de 16 caracteres x 2 líneas con interfaz I2C para un fácil control mediante un microcontrolador. Además, la intensidad y contraste de retroiluminación son ajustables.</p>
Processing Power (Esfuerzo computacional) para los algoritmos de procesamiento de datos que se implementarán en el Edge node	<p>Microcontrolador ESP32: Wifi 802.11n (2,4 GHz) hasta 150 Mbps, Bluetooth 4.2 BR/EDR y controlador de modo dual Bluetooth LE, microprocesador(es) LX6 de 32 bits de núcleo simple/doble Xtensa®, puntuación CoreMark® de 1 núcleo a 240 MHz y 2 núcleos a 240 MHz. Además, cuenta con 448 KB ROM, 520 KB SRAM, QSPI admite múltiples chips flash/SRAM, oscilador de cristal externo de 32 kHz para RTC con calibración, 34 GPIO programables, 10 sensores táctiles y LED PWM de hasta 16 canales.</p> <p>Todo ello para lograr lectura de sensores de temperatura, Control de servomotores, Interfaz con RFID y teclado, Actualización de display LCD, Comunicación WiFi para envío de datos y recepción de configuraciones, Procesamiento de lógica de negocio para dispensación y notificaciones.</p>

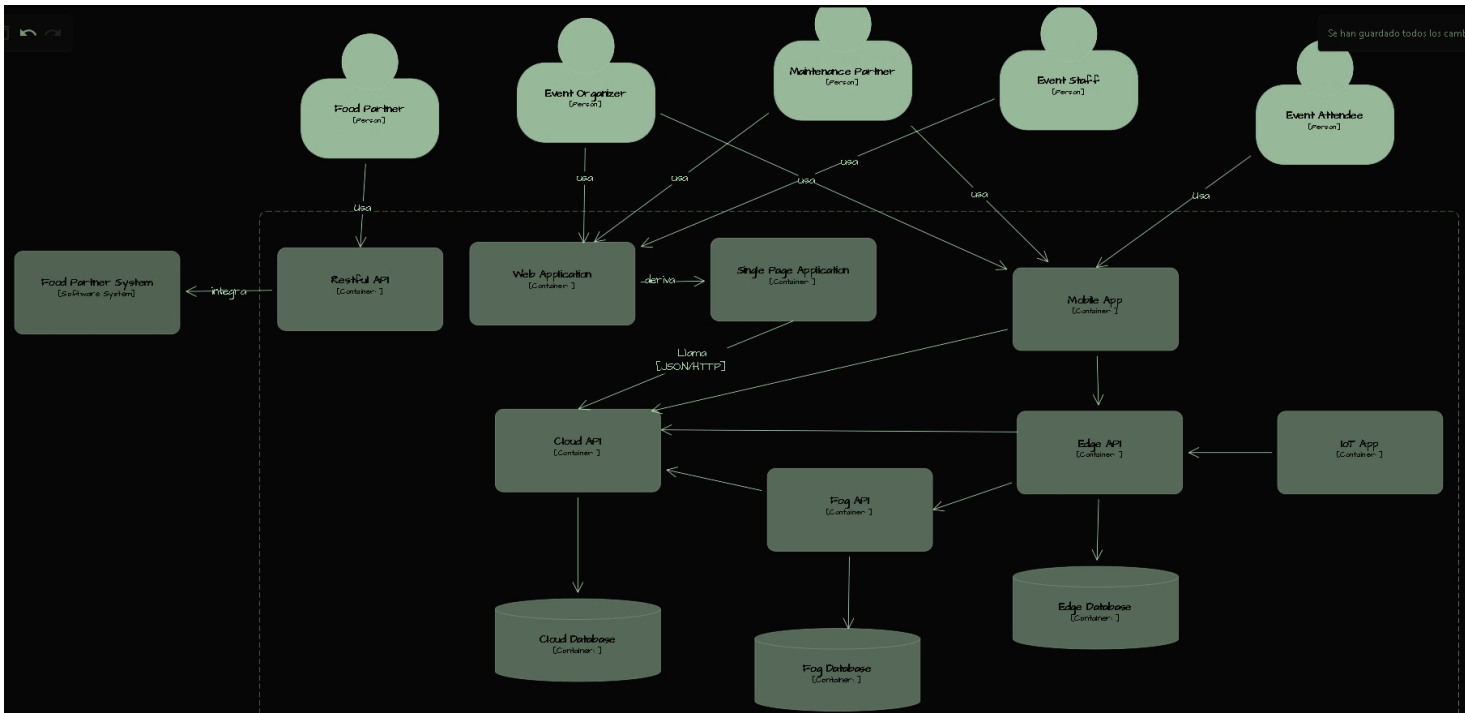
Pregunta 3:

Elabore el Definition of information layer requirements (requisitos de la capa de información). Considere: a) Definición de usuarios finales; b) Definición de número y tipos de servicios que deben proporcionarse a cada usuario final; c) Definición de las necesidades de información integrada para implementar cada servicio.

Information Layer Requirement	Especificación de requisitos
Definición de usuarios finales	<p>Asistentes al evento: Personas registradas que utilizan la aplicación móvil para check-in y la RFID Band durante el evento.</p> <p>Event Organizers: Personal encargado de la configuración y gestión general del evento.</p> <p>Event Registration and Check-In Staff: Personal en las estaciones de control de ingreso.</p> <p>Food Partner Users: Personal encargado del abastecimiento de insumos y productos para Smart Food Dispensers y Smart Coffee Makers.</p> <p>Device Maintenance Partner Users: Agentes de Soporte Técnico para usuarios de aplicación móvil y Personal en oficinas para la aplicación web.</p>
Definición de número y tipos de servicios que deben proporcionarse a cada usuario final	<p>Asistentes al evento entre 500 a 1000 por evento: Ellos tendrán servicio de registro y generación de código QR, servicio de check-in y asignación de RFID Band, servicio de acceso a salas y facilidades, servicio de pagos sin efectivo y servicio de dispensación de alimentos y bebidas mediante RFID Band.</p> <p>Event Organizers: Ellos tendrán servicio de configuración de información del evento, servicio de análisis de datos del evento y servicio de carga importante de información de personas registradas.</p> <p>Event Registration and Check-In Staff : Ellos tendrán servicio de validación de códigos QR y Servicio de activación y asignación de RFID Bands.</p> <p>Food Partner Users: Ellos tendrán servicio de monitoreo de inventario de Smart Food Dispensers y Smart Coffee Makers y servicio de gestión de abastecimiento.</p> <p>Device Maintenance Partner: Servicio de monitoreo de estado de dispositivos en tiempo real y servicio de gestión de mantenimiento preventivo y correctivo.</p>
Definición de las necesidades de información integrada para implementar cada servicio	<p>Servicio de registro y generación de código QR: Se necesitarán los datos personales del asistente y la información del evento.</p> <p>Servicio de check-in y asignación de RFID Band: Se necesita validación de código QR y asociación entre el asistente y RFID Band asignada.</p> <p>Servicio de pagos sin efectivo: se necesitará el saldo asociado a cada RFID Band e integración con sistema de facturación.</p> <p>Servicio de monitoreo y gestión de mantenimiento de dispositivos: Se necesita el estado operativo en tiempo real de todos los dispositivos, historial de mantenimiento por dispositivo, programación de mantenimiento preventivo y un sistema para el mantenimiento correctivo.</p>

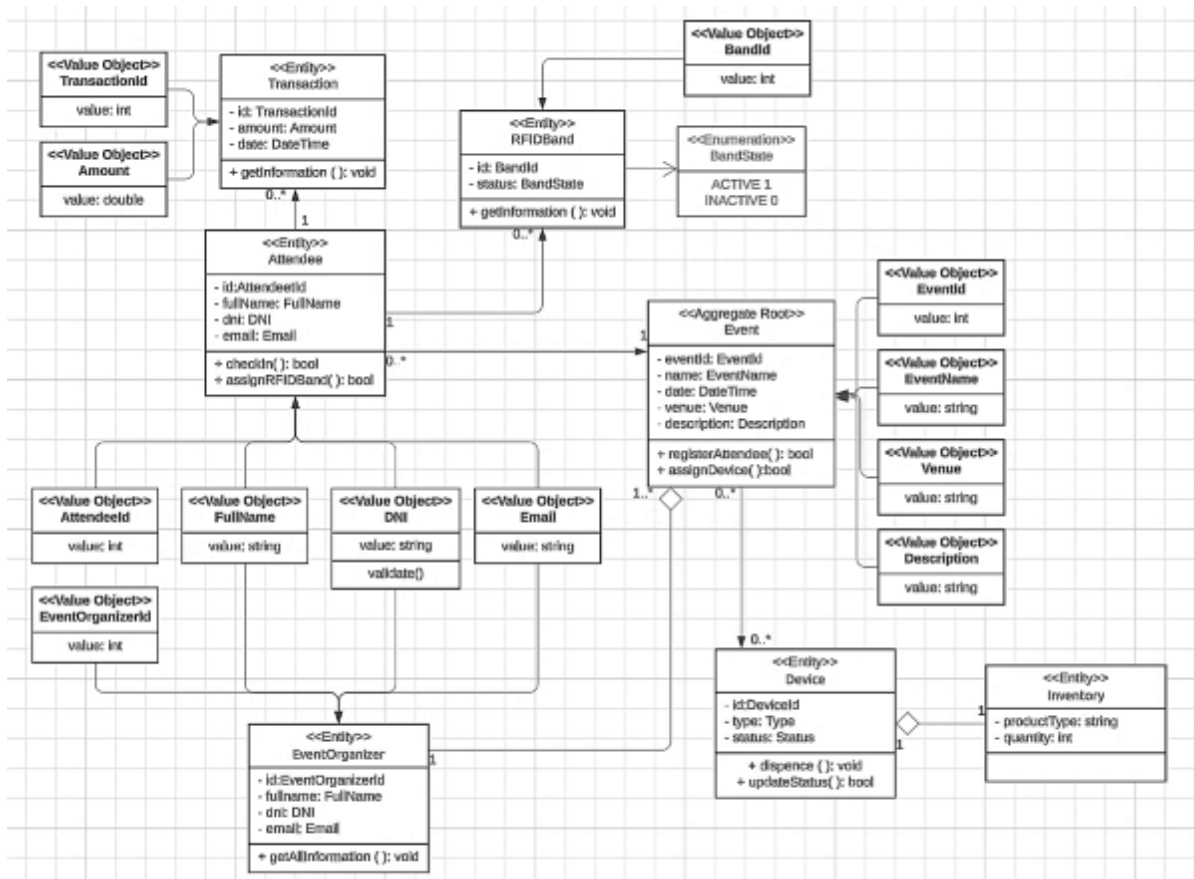
Pregunta 4:

Pegue sobre esta área la imagen capturada con el diagrama C4 Container Diagram elaborado con C4 Model



Pregunta 5:

Pegue sobre esta área la imagen capturada con el diagrama de clases elaborado con UML





Sección: SW74

Profesor: Velásquez Núñez, Ángel Augusto

Duración: 100 minutos Indicaciones:

1. El examen consta de 2 preguntas en base a un caso, y tendrá 100 minutos para resolverlas.
2. Las preguntas son en relación a un Caso y la entrega de su respuesta es a través de envío de dos archivos: un archivo .zip, conteniendo la exportación de su solución a la pregunta desarrollo de software y el documento de PowerPoint upc-pre-202402-si572-pc2-file_v1.pptx con su respuesta para la pregunta de redacción, colocando el contenido en la hoja de respuesta según el título e indicaciones.

Caso Moen, Inc.

En una noche típica de Seattle en 1937, un joven Al Moen estaba limpiando después de una noche de trabajo cuando fue a lavarse las manos en un antiguo grifo de dos manijas. Un repentino chorro de agua caliente lo hizo dar un salto hacia atrás alarmado y ese momento de sorpresa resultó ser la inspiración para el grifo de una sola manija, un invento que revolucionaría la plomería. Al Moen convirtió un problema en una solución, y su compromiso de encontrar una mejor manera sigue siendo tan importante para el trabajo de Moen ahora como lo fue para Al entonces.

Desde grifos inteligentes hasta filtración de agua y seguridad del agua, Moen creó productos innovadores que maximizan el rendimiento y minimizan el desperdicio.

En Moen (<https://www.moen.com/>), su equipo diseña productos con altos niveles de calidad y belleza, pero también para proteger el agua del mañana. Los productos Smart Water Network permiten a los clientes conservar agua rastreando su uso de agua a través de la aplicación Moen Smart Water.

Actualmente desea introducir Cia Steel Faucet, un dispositivo para interiores que consiste en un grifo inteligente con la tecnología MotionSense Wave™ (la capacidad de detección de proximidad y abrir o cerrar la válvula de agua sin contacto).

Features:

El dispositivo basado en ESP32, incluye un Ultrasound distance sensor, un Led y un Relay module conectado a la válvula de agua. El led, de color azul, se enciende cuando el dispositivo se encuentra activo y detecta que se traspasa el threshold de 10 cm. (cuando un usuario acerca sus manos al grifo) y se apaga cuando no detecta proximidad por debajo del threshold (se alejan las manos del usuario). Cuando el sensor de proximidad detecta que se alcanzó el threshold, el Relay Module debe abrir la válvula de agua del grifo durante 5 segundos. El flujo de agua es de 1.2 gpm max (4.5L/min) a 60 psi (pounds per square inch). El dispositivo muestra periódicamente en consola los valores de proximidad (en cm.) y estado de la válvula de agua (open / closed) con una frecuencia de 2.5 segundos. El dispositivo cuenta además con conexión WiFi.

También ofrece una aplicación móvil

(<https://play.google.com/store/apps/details?id=com.moen.smartwater&hl=en> y <https://apps.apple.com/us/app/moen-smart-water-network/id1495094343>) que brinda información de los dispositivos y permite la configuración remota de los mismos, como thresholds. La aplicación se empareja con los dispositivos disponibles en la red WiFi.

Technical constraints:

- El prototipo debe elaborarse con Wokwi y la programación debe ser con Arduino sketch (C++) aplicando Object-Oriented Programming para el Embedded Application, creando clases y

objetos para representar el estado y comportamiento de los diversos tipos de sensores y actuators en el device.

- El contenido de sketch.ino debería enfocarse en la generación de instancia de un objeto de tipo Device e invocar al comportamiento correspondiente en los métodos setup() y loop().
- Aplique convenciones de nomenclatura en inglés, con principios, patrones y buenas prácticas de diseño de software orientado a objetos.
- Documente con comentarios en inglés sus archivos de código fuente, incluyendo title blocks (ver Arduino Style Guide for Writing Content en la sección de referencias).
- Al inicio de la ejecución debe mostrarse en consola la información de la empresa, así como sus nombres y apellidos como developer miembro del equipo de Moen, inc. Considere en sus objetos y archivo sketch el código necesario para describir y demostrar los modos de operación y actualización de estado requeridos en el IoT Device.
- Exporte el proyecto de Wokwi y empaquete el mismo como un archivo .zip. (único formato válido) con el nombre upc-pre-202402-si572-<sección>-pc2-u<código-estudiante>.zip (por ejemplo, upc-pre-202402si572-sw74-pc2-u201621873.zip).
- Suba dos archivos: El archivo upc-pre-202402-si572-pc2-file_v1.pptx con sus respuestas y el archivo .zip con el proyecto de solución de Wokwi, en la Actividad indicada para el Examen.

Pregunta 1 (5 p.).

Bajo el marco de los 12 pasos del IoT System Design Steps (ver Anexo A), elabore el Definition of application service layer requirements (requisitos relacionados con la capa de servicio de la aplicación). Aquí considere a) Definición de la interfaz de usuario para cada servicio prestado (dependiendo del servicio la aplicación para dicho servicio podría requerir una interfaz gráfica para móvil, o tablet o desktop o más de un tipo).

Pregunta 2 (15 p.)

Cree un nuevo proyecto en Wokwi y desarrolle un prototipo que satisfaga los requisitos planteados para el Cia Steel Faucet.

Referencias

Wokwi Parts:

<https://docs.wokwi.com/guides/esp32/> <https://docs.wokwi.com/parts/wokwi-hc-sr04/>
<https://docs.wokwi.com/parts/wokwi-led> <https://docs.wokwi.com/parts/wokwi-relay-module>

Wokwi reference projects:

<https://wokwi.com/projects/373163766425063425> <https://wokwi.com/projects/290056311044833800>

Referencias adicionales:

Arduino Style Guide for Writing Content
<https://docs.arduino.cc/learn/contributions/arduino-writing-style-guide/>

Solución: [Nota de la PC2 con esta solución = 17.5](#)

Pregunta 1:

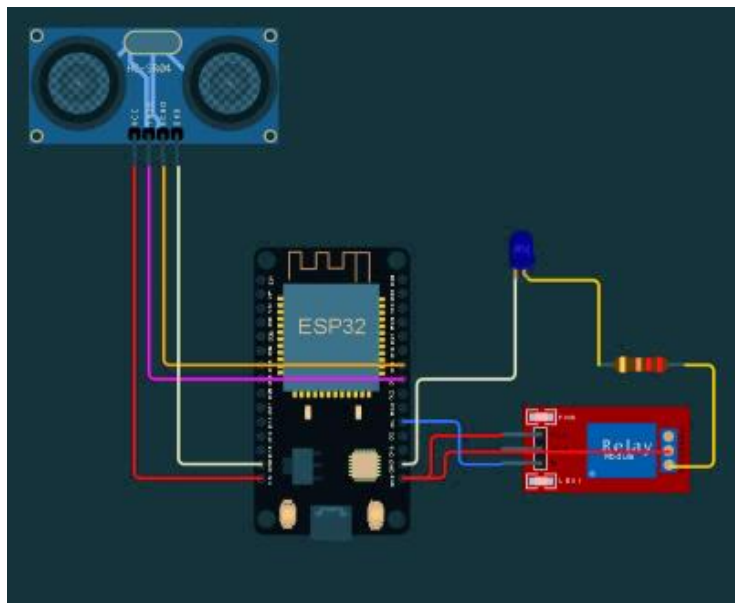
Elabore el Definition of application service layer requirements (requisitos relacionados con la capa de servicio de la aplicación). Aquí considere a) Definición de la interfaz de usuario para cada servicio prestado (dependiendo del servicio la aplicación para dicho servicio podría requerir una interfaz gráfica para móvil, o tablet o desktop o más de un tipo).

Application Service Layer Requirement	Especificación de requisitos
Servicio 1: Servicio de Monitoreo de Uso de Agua	<ul style="list-style-type: none"> - Interfaz gráfica en la aplicación móvil y la aplicación de escritorio para visualizar el uso de agua en tiempo real - Gráficos que muestren el consumo de agua diario, semanal y mensual. - Alertas que notifiquen al usuario sobre patrones inusuales de uso de agua.
Servicio 2: Servicio de Control de Válvula	<ul style="list-style-type: none"> - Interfaz en la aplicación móvil para permitir a los usuarios abrir o cerrar la válvula manualmente. - Indicadores visuales que muestran el estado de la válvula ya sea si está abierta o cerrada y el flujo de agua. - Opción para programar los horarios para el funcionamiento automático del grifo.
Servicio 3: Servicio de Configuración del Dispositivo	<ul style="list-style-type: none"> - Interfaz de configuración desde la aplicación móvil para ajustar el umbral o threshold de proximidad del sensor que detecta cuando el usuario acerca las manos al grifo y que por defecto detecta si traspasa los 10 cm. - Opción para cambiar la duración del flujo de agua que realiza el Relay Module al abrir la válvula de agua (por defecto está en 5 segundos) según las preferencias del usuario. - Configuración de la conectividad WiFi y gestión de la red.
Servicio 4: Servicio de Mantenimiento y Soporte . Servicio 5: Servicio sobre el Uso del Agua	<ul style="list-style-type: none"> - Sección en la aplicación para acceder a manuales de usuario y guías para solucionar problemas recurrentes. - Opción para enviar consultas al soporte técnico directamente desde la aplicación móvil. - Notificaciones sobre actualizaciones de software y mantenimiento. - Interfaz que ofrezca consejos y recomendaciones personalizadas para la conservación del agua basadas en el consumo del usuario. - Gráficos y estadísticas que comparen el uso de agua del usuario con promedios de consumo en su área, solo en caso de que otros usuarios también usen la misma solución IoT. - Challenges para reducir el consumo de agua, con recompensas virtuales o logros para motivar a los usuarios. - Notificaciones periódicas que recuerden a los usuarios la importancia de conservar el agua y cómo el mal uso impacta al medio ambiente.

Pregunta 2:

Coloque aquí el URL del proyecto en Wokwi

<https://wokwi.com/projects/<projectID>>



File: sketch.ino

```
#include "Device.h"

/*
  Cla Steel Faucet

  This sketch controls a smart faucet using an ultrasonic distance sensor,
  a relay module to control the water valve, and an LED indicator.

  The circuit:
  * Ultrasonic sensor connected to pins 5 (trigger) and 18 (echo)
  * Relay module connected to pin 4
  * LED connected to pin

  Created 8 November 2024
  By HOLA MUNDO :)
*/

Device device;

void setup() {
  device.setup();
}

void loop() {
  device.loop();
}
```

File: diagram.json

```
{
  "version": 1,
  "author": "Hola Mundo :)",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -62.5, "left": 23.8, "attrs": {} },
    {
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": -209.7,
      "left": -128.9,
      "attrs": { "distance": "200" }
    },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": -80.4,
      "left": 186.2,
      "attrs": { "color": "blue" }
    },
    {
      "type": "wokwi-relay-module", "id": "relay1", "top": 48.2, "left": 192, "attrs": {} },
    {
      "type": "wokwi-resistor",
      "id": "r2",
      "top": 13.85,
      "left": 257.8,
      "rotate": 180,
      "attrs": { "value": "220" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [ ] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [ ] ],
    [ "relay1:VCC", "esp:3V3", "red", [ "h-48", "v28.8" ] ],
    [ "relay1:COM", "esp:3V3", "red", [ "h-152.4", "v17.8" ] ],
    [ "relay1:IN", "esp:D4", "blue", [ "h-28.8", "v-29.1" ] ],
    [ "relay1:NO", "r2:1", "gold", [ "h30", "v-69" ] ],
    [ "r2:2", "led1:A", "gold", [ "v0" ] ],
    [ "led1:C", "esp:GND.1", "black", [ "v67.2", "h-66.8", "v57.7" ] ],
    [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v0" ] ],
    [ "ultrasonic1:TRIG", "esp:D5", "magenta", [ "v0" ] ],
    [ "ultrasonic1:ECHO", "esp:D18", "orange", [ "v0" ] ],
    [ "ultrasonic1:GND", "esp:GND.2", "black", [ "v0" ] ]
  ],
  "dependencies": {}
}
```


File: libraries.txt

Wokwi Library List

See <https://docs.wokwi.com/guides/libraries>

WiFi

Time

File: Device.h

```
#ifndef DEVICE_H
#define DEVICE_H

#include <Arduino.h>
#include "SensorManager.h"
#include "ActuatorManager.h"
#include "StateManager.h"

class Device {
public:
    Device();
    void setup();
    void loop();

private:
    SensorManager sensorManager;
    ActuatorManager actuatorManager;
    StateManager stateManager;
};

#endif
```

File: Device.cpp

```
#include "Device.h"

Device::Device() {}

void Device::setup() {
    Serial.begin(9600);
    Serial.println("Moen, Inc. - Cia Steel Faucet Initialized");
    Serial.println("Get Healthier, Better-Tasting Water with Moen.");
    Serial.println("Developer: HOLA MUNDO :)");

    sensorManager.setup();
    actuatorManager.setup();
}

void Device::loop() {
    sensorManager.updateSensors();

    stateManager.setDistance(sensorManager.getDistance()); // Set state in
    StateManager

    actuatorManager.updateActuators(stateManager.getDistance()); // Update
    actuators based on distance

    // Print distance and valve state to console
    Serial.print("Distance: ");
    Serial.print(stateManager.getDistance());
    Serial.print(" cm, Valve: ");
    Serial.println(actuatorManager.getDoorState() == 1 ? "Open" : "Closed");

    delay(2500); // Delay for 2.5 seconds
}
```

File: SensorManager.h

```
#ifndef SENSORMANAGER_H
#define SENSORMANAGER_H

#include <Arduino.h>

class SensorManager {
public:
    void setup();
    void updateSensors();
    double getDistance();

private:
    int triggerPin = 5; // Trigger pin for ultrasonic sensor
    int echoPin = 18;   // Echo pin for ultrasonic sensor
    double distance;    // Distance measured by the sensor
};

#endif
```

File: SensorManager.cpp

```
#include "SensorManager.h"

void SensorManager::setup() {
    pinMode(triggerPin, OUTPUT); // Set trigger pin as output
    pinMode(echoPin, INPUT);     // Set echo pin as input
}

// Update distance sensor readings
void SensorManager::updateSensors() {
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);

    long duration = pulseIn(echoPin, HIGH); // Measure duration of echo
    distance = (duration * 0.034) / 2;      // Convert duration to distance in
    cm
}

double SensorManager::getDistance() {
    return distance;
}
```

File: ActuatorManager.h

```
#ifndef ACTUATORMANAGER_H
#define ACTUATORMANAGER_H
#include <Arduino.h>

class ActuatorManager {
public:
    void setup();
    void updateActuators(double distance);
    int getDoorState();

private:
    int relayPin = 4;    // Relay pin to control the water valve
    int ledPin = 2;      // LED pin to indicate status
    int doorState = 0;   // 0 = Closed, 1 = Open
};

#endif
```

File: ActuatorManager.cpp

```
#include "ActuatorManager.h"

void ActuatorManager::setup() {
    pinMode(relayPin, OUTPUT);
    pinMode(ledPin, OUTPUT);
    digitalWrite(relayPin, LOW);
    digitalWrite(ledPin, LOW);
}

// Update actuators based on distance
void ActuatorManager::updateActuators(double distance) {
    if (distance < 10) {                // Threshold of 10 cm
        doorState = 1;                 // Open door
        digitalWrite(relayPin, HIGH);
        digitalWrite(ledPin, HIGH);
        delay(5000);                   // Keep relay on for 5 seconds
        digitalWrite(relayPin, LOW);
    } else {
        doorState = 0;                 // Close door
        digitalWrite(ledPin, LOW);
    }
}

int ActuatorManager::getDoorState() {
    return doorState;
}
```

File: StateManager.h

```
#ifndef STATEMANAGER_H
#define STATEMANAGER_H

#include <Arduino.h>

class StateManager {
public:
    void setDistance(double distance);
    double getDistance();

private:
    double distance;
};

#endif
```

File: StateManager.cpp

```
#include "StateManager.h"

void StateManager::setDistance(double distance) {
    this->distance = distance;
}

double StateManager::getDistance() {
    return distance;
}
```

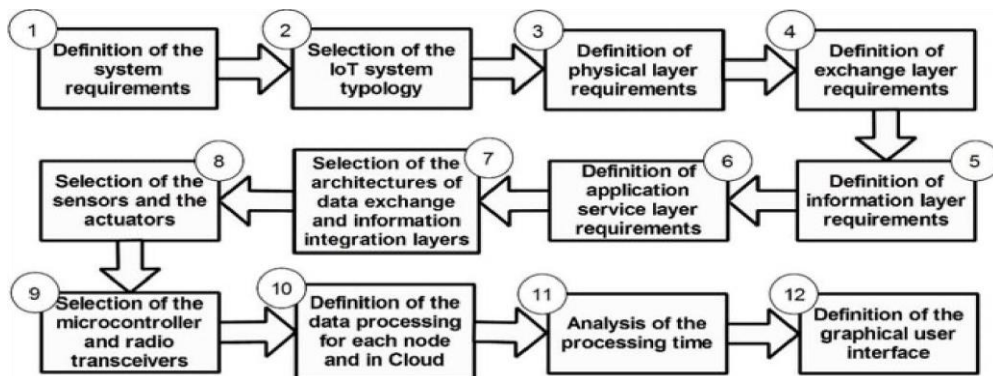
Library Manager



Anexos que estuvieron en todos los exámenes:

Anexo A. IoT System Design Steps

Como se trató en clase, los ingenieros Eulalia Balestrieri, Luca De Vito, Francesco Lamonaca, Francesco Picariello, Sergio Rapuano y Ioan Tudosa de la Universidad de Sannio en Italia, proponen un conjunto de design guidelines para IoT System Design, sintetizados en 12 steps.



Anexo B. Edge-To-Cloud Architecture Layers

Este diagrama que ilustra la relación entre las capas Edge Layer, Fog Layer y Cloud layer.

