



Práctica Calificada 2 SI 572: Desarrollo de Soluciones IoT 2024-2

Desarrollo de Soluciones IoT (Universidad Peruana de Ciencias Aplicadas)



Escanea para abrir en Studocu



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

Sección: SW71
Profesor: León Baca, Marco Antonio

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 con 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.
-

Enunciado:

Protech Innovations, inc.

Protech Innovation es una empresa de tecnología fundada en 2012, especializada en soluciones de seguridad industrial y residencial. Su misión es proteger a las personas y bienes mediante dispositivos de detección temprana de riesgos, apoyándose en tecnologías IoT para crear entornos seguros y monitoreados en tiempo real. Uno de los productos destacados de su línea SecureWatch es el GLP SecureSense Pro, un dispositivo diseñado específicamente para monitorear la concentración de gas licuado de petróleo (GLP) en espacios cerrados o de alto riesgo.

Descripción del GLP SecureSense Pro

El GLP SecureSense Pro está equipado con un sensor de gas MQ-2, el cual es capaz de detectar concentraciones de GLP en el aire. El dispositivo cuenta con un sistema de señalización visual mediante LEDs que cambian de color según el nivel de concentración de GLP, ofreciendo una alerta rápida y visible para que los trabajadores o residentes puedan tomar medidas de precaución. Adicionalmente, el dispositivo puede conectarse a una plataforma de monitoreo en la nube, que envía notificaciones a dispositivos móviles y sistemas de alarma en caso de niveles peligrosos de GLP.

Sistema de Alerta del GLP SecureSense Pro

El sistema de señalización se basa en tres LEDs (verde, amarillo y rojo), que indican el nivel de riesgo de acuerdo con la concentración de GLP medida en PPM (partes por millón):

Concentración de GLP (ppm)	Estado	Led
Menor a 200 PPM	Nivel Seguro	Verde
Entre 200 y 500 PPM	Nivel Moderado	Amarillo
Mayor a 500 PPM	Nivel Crítico, Riesgo de Explosión.	Rojo

Features:

Actualmente cuentan con un prototipo en wokwi:

<https://wokwi.com/projects/412404352853061633>

En dicho prototipo existen aún features por ajustar, mejoras al sensor de Gas, la implementación de los leds indicadores de estado (seguro, moderado, crítico), así como la correcta visualización en el display de la concentración de GLP en ppm y el estado de la medición (seguro, moderado, crítico). El prototipo actual utiliza ESP32, un LCD 1602 (<https://docs.wokwi.com/parts/wokwi-lcd1602>). También le solicitan que su propuesta mejore la experiencia con el device, brindando mejor información y uso del espacio a través del display. Para ello debe utilizar el display LCD 20024 (<https://docs.wokwi.com/parts/wokwi-lcd2004>) el cual brinda mayor espacio.

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 *sensors* 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 Protech Innovation, 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-202402-si572-wx71-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 final.

Referencias

Wokwi Parts:

<https://docs.wokwi.com/guides/esp32/>
<https://docs.wokwi.com/parts/wokwi-lcd2004>

Wokwi Libraries:

(buscar con ese nombre en Wokwi Library Manager)

LiquidCrystal I2C

Wokwi reference projects:

<https://wokwi.com/projects/412404352853061633>

En base al caso:

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 Smart Vitalmonitor BeyondLock.

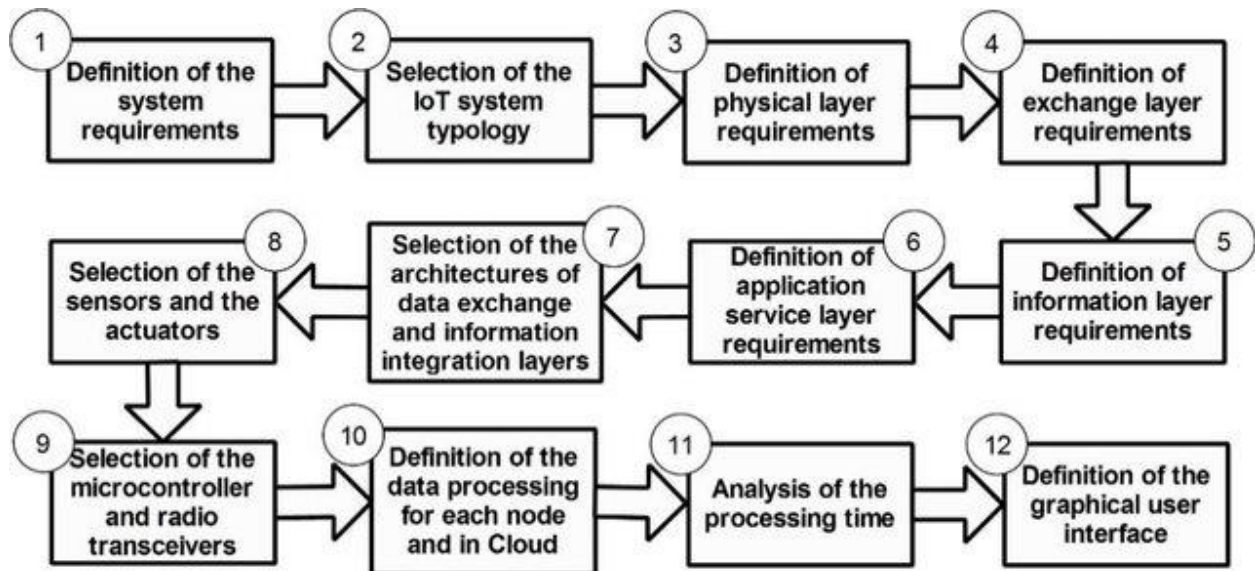
Rúbrica de calificación

Criterio de Calificación	Sobresaliente (S)	Esperado (E)	Necesita Mejorar (M)	Insuficiente (I)	Calificación
C01. Definition of application service layer requirements	Define de forma clara y completa los requisitos relacionados con la capa de servicio de la aplicación, en términos de definición de la interfaz de usuario para cada servicio prestado, estando en todos los casos alineados con las necesidades de la solución de IoT.	Define de forma clara y completa la mayoría de los requisitos relacionados con la capa de servicio de la aplicación, en términos de definición de la interfaz de usuario para cada servicio prestado, estando en todos los casos alineados con las necesidades de la solución de IoT.	Define algunos de los requisitos relacionados con la capa de servicio de la aplicación, en términos de definición de la interfaz de usuario para cada servicio prestado, o en varios casos estos no están alineados con las necesidades de la solución de IoT.	No brinda respuesta o lo especificado no tiene relación con el caso.	
	5.0 puntos	3.5 puntos	1.5 punto	0 puntos	
C02. IoT Device Design	El diseño de prototipo de dispositivo cumple de forma completa con la configuración solicitada.	El diseño del prototipo de dispositivo cumple con la mayoría de los requisitos sobre configuración.	El diseño del prototipo de dispositivo cumple con algunos de los requisitos sobre configuración.	No diseña prototipo de dispositivo o este no cumple con la mayoría de requisitos sobre configuración.	
	5.0 puntos	3.5 punto	1.5 puntos	0 puntos	
C03. Software Features	Se evidencia la funcionalidad de las operaciones solicitadas, proporcionando en cada caso los valores esperados y respondiendo adecuadamente ante las excepciones.	Se evidencia parcialmente la funcionalidad de las operaciones solicitadas, proporcionando en algunos casos los valores esperados y respondiendo de forma parcialmente adecuada ante las excepciones.	La ejecución no cumple con las funcionalidades solicitadas.	No implementa comportamiento para el dispositivo.	
	5.0 puntos	3.5 punto	1.5 puntos	0 puntos	
C04. Quality Attributes	Utiliza el lenguaje de programación Arduino Sketch C++. En todos los casos Incluye información de propósito y autoría como comentarios. La codificación tiene un estilo claro, indentando los bloques de código según los estándares de programación correspondientes al lenguaje, aplicando una lógica consistente en los métodos, condicionales sin escenarios no contemplados, uso adecuado de reutilización de código para evitar redundancia. Aplica patrones de arquitectura y patrones de diseño. Cumple de forma completa con los technical constraints.	Utiliza el lenguaje de programación Arduino Sketch C++. En la mayoría de los casos Incluye información de propósito y autoría como comentarios. La codificación es funcional, aplica en la mayoría de casos los estándares de indentación de bloques de código, ó existen algunas ineficiencias en la codificación: redundancia ó inconsistencias en la lógica de programación. Aplica parcialmente patrones de arquitectura y patrones de diseño. Cumple con la mayoría de technical constraints.	Utiliza el lenguaje de programación Arduino Sketch C++. En algunos casos Incluye información de propósito y autoría como comentarios. La codificación es funcional, pero solo aplica algunos de los estándares de indentación de bloques de código, ó existen muchas ineficiencias en la codificación: redundancia ó inconsistencias en la lógica de programación. Aplica algunos patrones de arquitectura y patrones de diseño. Cumple con solo algunos de los technical constraints.	No utiliza el lenguaje de programación Arduino Sketch C++, ó la codificación es funcional pero no se evidencia aplicación de estándares ó criterios de eficiencia en la codificación, con ausencia de comentarios, ó no aplica patrones de arquitectura ni patrones de diseño, o la codificación no es funcional.	
	3.0 puntos	2.0 punto	1.0 puntos	0 puntos	
C05. Naming standards	El desarrollador aplica en todos los nombres de objetos de programación y base de datos como paquetes, componentes, interfaces, clases, objetos, variables, constantes, métodos, la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como los recursos.	El desarrollador aplica en la mayoría de casos la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como los recursos.	El desarrollador aplica en muy pocos casos la nomenclatura en inglés y la nomenclatura estándar para identificadores de clases, objetos, miembros de programación, así como los recursos.	El desarrollador no aplica nomenclatura en inglés para los objetos de programación ó recursos.	
	2.0 puntos	1.0 punto	0.5 puntos	0 puntos	
Total	20 puntos	13.5 puntos	7.0 puntos	0 puntos	

Lima, 07 de noviembre del 2024

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.

