

Actividad #3 **#Smart City**

línea corta

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA EN CIENCIAS Y SISTEMAS

Notas:

1. Donde encuentres el símbolo “#” deberás sustituirlo por el numero de la semana en el que te encuentres.
2. En el apartado de Definición de conceptos la “n” la sustituirás por cada nuevo concepto que agregues, por ejemplo, si agregas tres conceptos y te encuentras en la semana 5, deberán verse tres recuadros, el primero identificado como 5.1.1, el segundo como 5.1.2 y el tercero como 5.1.3
3. De la misma manera en el apartado de objetivos la “n” deberás sustituirla por cada objetivo que agregues, si te encuentras en la semana 2 y agregas dos objetivos el primer recuadro se denominará Objetivo 2.2.1 y el segundo se denominará 2.2.2

|  |  |
| --- | --- |
|  | |
| #.1 | **DEFINICIÓN DE CONCEPTOS** |

|  |  |
| --- | --- |
| Definición #.1.1 |  |
| Un sistema embebido es un sistema de computación diseñado para realizar funciones específicas, muchas veces en tiempo real, dentro de un sistema mayor. Se caracteriza por su bajo consumo, tamaño compacto y operación autónoma. | |

|  |  |
| --- | --- |
| Definición #.1.2 |  |
| * **ARM**: Arquitectura usada en la mayoría de Raspberry Pi. Optimizada para bajo consumo energético y eficiencia. * **x86**: Arquitectura usada en PCs tradicionales y mini PCs como Intel NUC. Generalmente más poderosa, pero con mayor consumo. | |
| Definición #.1.3 |  |
| Capacidad de enviar datos a servidores remotos mediante protocolos como HTTP, MQTT, o plataformas como AWS IoT, Azure IoT Hub, Blynk o ThingSpeak. | |

|  |  |
| --- | --- |
|  | |
| #.2 | **COMPETENCIA(S) DEL EJEMPLO A IMPARTIR** |

|  |  |
| --- | --- |
| Competencia #.2.1 |  |
| El estudiante Identifica los componentes esenciales de un sistema IoT mediante el análisis de documentación técnica y catálogos de hardware para describir las funcionalidades básicas de cada elemento del sistema | |

|  |  |
| --- | --- |
|  | |
| #.3 | **SOFTWARE O MATERIALES NECESARIOS** |

|  |  |
| --- | --- |
| Descripción |  |
| Practica en clase  **Ejercicio Integrador Alternativo: Diseño de un Prototipo de Ciudad Inteligente (Smart City) con Raspberry Pi y Servicios IoT**  **Objetivo:** Diseñar un módulo funcional para una ciudad inteligente utilizando Raspberry Pi, sensores, actuadores, programación en Python, y servicios en la nube. El módulo debe enfocarse en uno de los sistemas: iluminación pública inteligente, monitoreo de calidad del aire, o recolección inteligente de residuos.  **Parte 1: Enunciado de Diseño (Proyecto Conceptual + Prototipo Técnico)**  **Escenario:**  Tu equipo es parte de una empresa que desarrolla soluciones IoT para ciudades inteligentes. Se les ha encargado diseñar y prototipar uno de los siguientes módulos:   * 🟡 **Iluminación inteligente:** sistema que enciende/apaga farolas basado en luz ambiental y movimiento. * 🟢 **Calidad del aire:** monitoreo de gases contaminantes y temperatura en diferentes zonas. * 🔵 **Gestión de residuos:** sensor de nivel en un contenedor que notifica cuándo debe vaciarse.   **Parte 2: Actividades por Tema**   1. **Control del Hardware de Raspberry Pi con Python**    * Controla entradas (sensor de movimiento, sensor de gas o ultrasonido) y salidas (LED o buzzer) con Python.    * Crea una rutina que automatice respuestas según condiciones específicas. 2. **Control de Sensores y Actuadores con Python**    * Utiliza sensores como LDR, MQ-135, DHT22 o HC-SR04 según el módulo elegido.    * Activa actuadores (luces, ventiladores, alarmas) desde el script en Python según el estado del sensor. 3. **Raspberry Pi y la Nube**    * Implementa una conexión a Internet vía Wi-Fi o Ethernet.    * Usa una API REST para enviar datos a la nube (por ejemplo: **ThingSpeak**, **IFTTT**, **Firebase Realtime Database**, **MQTT con HiveMQ**). 4. **Principales Servicios para Sistemas IoT**    * Investiga y justifica el uso de una plataforma IoT moderna (como **Blynk**, **Node-RED**, **AWS IoT**, etc.).    * Muestra una visualización de los datos del sensor en tiempo real o por histórico.   **Parte 3: Desarrollo y Simulación**   * Diagrama del sistema con sensores, actuadores y flujos de datos. * Código Python modular y comentado. * Simulación del sistema si no se cuenta con hardware (opcional: usar **Node-RED + simulador MQTT**, o gráficos con Matplotlib en Python). * Demostración de envío/recepción de datos desde/hacia la nube.   **Parte 4: Análisis y Proyección**   * ¿Cómo podría escalarse este sistema a una ciudad completa? * ¿Qué problemas de seguridad, energía o red podrían surgir? * ¿Qué ventaja ofrece Raspberry Pi frente a otros microcontroladores (como ESP32) en este tipo de aplicaciones?   **✅ Entregables**   * Documento o presentación técnica con:   + Descripción del módulo elegido.   + Diagrama del sistema.   + Código en Python.   + Capturas del funcionamiento en la nube. * (Opcional) Video demo. * Rúbrica de evaluación: innovación, integración, funcionalidad, presentación. | |

|  |  |
| --- | --- |
| Herramientas |  |
| Redacción y Proteus | |

   
Sin necesidad



|  |  |
| --- | --- |
|  | |
| #.4 | **EJERCICIOS EN CLASE** |

|  |  |
| --- | --- |
| Descripción |  |
| Practica en clase  **Ejercicio Integrador Alternativo: Diseño de un Prototipo de Ciudad Inteligente (Smart City) con Raspberry Pi y Servicios IoT**  **Objetivo:** Diseñar un módulo funcional para una ciudad inteligente utilizando Raspberry Pi, sensores, actuadores, programación en Python, y servicios en la nube. El módulo debe enfocarse en uno de los sistemas: iluminación pública inteligente, monitoreo de calidad del aire, o recolección inteligente de residuos.  **Parte 1: Enunciado de Diseño (Proyecto Conceptual + Prototipo Técnico)**  **Escenario:**  Tu equipo es parte de una empresa que desarrolla soluciones IoT para ciudades inteligentes. Se les ha encargado diseñar y prototipar uno de los siguientes módulos:   * 🟡 **Iluminación inteligente:** sistema que enciende/apaga farolas basado en luz ambiental y movimiento. * 🟢 **Calidad del aire:** monitoreo de gases contaminantes y temperatura en diferentes zonas. * 🔵 **Gestión de residuos:** sensor de nivel en un contenedor que notifica cuándo debe vaciarse.   **Parte 2: Actividades por Tema**   1. **Control del Hardware de Raspberry Pi con Python**    * Controla entradas (sensor de movimiento, sensor de gas o ultrasonido) y salidas (LED o buzzer) con Python.    * Crea una rutina que automatice respuestas según condiciones específicas. 2. **Control de Sensores y Actuadores con Python**    * Utiliza sensores como LDR, MQ-135, DHT22 o HC-SR04 según el módulo elegido.    * Activa actuadores (luces, ventiladores, alarmas) desde el script en Python según el estado del sensor. 3. **Raspberry Pi y la Nube**    * Implementa una conexión a Internet vía Wi-Fi o Ethernet.    * Usa una API REST para enviar datos a la nube (por ejemplo: **ThingSpeak**, **IFTTT**, **Firebase Realtime Database**, **MQTT con HiveMQ**). 4. **Principales Servicios para Sistemas IoT**    * Investiga y justifica el uso de una plataforma IoT moderna (como **Blynk**, **Node-RED**, **AWS IoT**, etc.).    * Muestra una visualización de los datos del sensor en tiempo real o por histórico.   **Parte 3: Desarrollo y Simulación**   * Diagrama del sistema con sensores, actuadores y flujos de datos. * Código Python modular y comentado. * Simulación del sistema si no se cuenta con hardware (opcional: usar **Node-RED + simulador MQTT**, o gráficos con Matplotlib en Python). * Demostración de envío/recepción de datos desde/hacia la nube.   **Parte 4: Análisis y Proyección**   * ¿Cómo podría escalarse este sistema a una ciudad completa? * ¿Qué problemas de seguridad, energía o red podrían surgir? * ¿Qué ventaja ofrece Raspberry Pi frente a otros microcontroladores (como ESP32) en este tipo de aplicaciones?   **✅ Entregables**   * Documento o presentación técnica con:   + Descripción del módulo elegido.   + Diagrama del sistema.   + Código en Python.   + Capturas del funcionamiento en la nube. * (Opcional) Video demo. * Rúbrica de evaluación: innovación, integración, funcionalidad, presentación. | |

|  |  |
| --- | --- |
| Entrega |  |
| Entrega de un PDF dando respuesta a todas las preguntas planteadas y el archivo generado por Proteus | |

|  |  |
| --- | --- |
|  | |
| #.1 | **VALORES Y ACTITUDES** |

|  |  |
| --- | --- |
| Aplicación de Valores |  |
| * Excelencia: Aplicación práctica: La comparación detallada entre Raspberry Pi y mini PC x86 muestra una búsqueda de la mejor solución técnica para el cliente, enfocándose en criterios como consumo, escalabilidad y conectividad. * Compromiso: Aplicación práctica: El análisis se desarrolló pensando en resolver una necesidad real de un cliente, demostrando empatía y disposición para brindar una solución efectiva. | |