PROYECTO (ABP)

Módulo: Programador

ÍNDICE

Situación Profesional

La empresa ficticia SmartHome Solutions los ha contratado para desarrollar un programa capaz de gestionar dispositivos inteligentes dentro de una vivienda. Este programa permitirá a los usuarios interactuar con diversos dispositivos como luces, termostatos, cámaras de seguridad y electrodomésticos, todo desde una interfaz centralizada. El proyecto estará basado en programación orientada a objetos, el uso de estructuras de datos con conexión a base de datos.

La empresa SmartHome Solutions busca desarrollar una solución que no solo sea funcional, sino también ética y responsable. La empresa adhiere a los principios del AWS Well-Architected Framework para garantizar que su sistema de hogar inteligente sea seguro, eficiente, confiable, optimizado en costos y sostenible. El proyecto deberá considerar las implicaciones éticas del manejo de datos personales sensibles recolectados en ambientes íntimos como el hogar, así como la responsabilidad profesional en el desarrollo de sistemas que impactan directamente en la seguridad y privacidad de los usuarios.

Objetivo Principal

Crear un programa que permita:

1. Registro e inicio de sesión.
2. Registrar dispositivos inteligentes para el hogar.
3. Controlar encendido, apagado y ajustes específicos de los dispositivos.
4. Consultar el estado actual de cada dispositivo en tiempo real.
5. Diseñar la arquitectura del sistema siguiendo al menos cuatro de los seis pilares del AWS Well-Architected Framework: Excelencia Operativa, Seguridad, Fiabilidad, Eficiencia del Rendimiento, Optimización de Costos y Sostenibilidad.
6. Incluir una problemática de su elección.

Requerimientos generales funcionales del programa

- **Gestión de Usuarios.** El programa debe permitir registrar nuevos usuarios y almacenar su información. Los usuarios podrán iniciar sesión para acceder a sus datos y gestionar sus dispositivos.
- **Gestión de Dispositivos.** El programa debe permitir registrar dispositivos nuevos con detalles como tipo (luz, cámara, electrodoméstico) y estado inicial (encendido/apagado). Debe permitir modificar las características del dispositivo (por ejemplo, ajustar la temperatura en un termostato).
- **Control de Estado.** Los usuarios podrán ver el estado actual de todos los dispositivos registrados (encendido, apagado, configuraciones específicas)
- **Automatización**. Implementar reglas automáticas como encender luces al anochecer o ajustar el termostato cuando la temperatura exterior cambie.
- **Transparencia Algorítmica.** El sistema debe ofrecer información clara sobre cómo funcionan las automatizaciones y permitir a los usuarios auditar las decisiones automáticas tomadas por el sistema.

Problemáticas extra para elegir (Automatización)

A continuación se listan algunas ideas problemáticas para su elección (**elegir o definir sólo una).**

- **Modo Ahorro de Energía.** Crear una funcionalidad que apague todos los dispositivos no esenciales cuando los usuarios seleccionan el “Modo Ahorro”.
- **Automatización Personalizada.** Permitir a los usuarios programar acciones automáticas, como encender la cafetera a las 7:00 am.
- **Notificaciones Simuladas.** Añadir alertas para informar cambios importantes, como “Se detectó movimiento en la sala” o “la temperatura de la habitación ha excedido el límite configurado”.
- **Escenarios Predefinidos.** Ofrecer opciones de configuración rápida como

"Modo fiesta" (luces encendidas y música activada) o "Modo Noche" (luces apagadas, cámaras activadas).

- **Otra de tu interés**.

Requerimientos no funcionales

- **Modularidad**. El código fuente debe estar estructurado en módulos, clases y funciones independientes para facilitar la escalabilidad, mantenimiento y comprensión.
- **Legibilidad**. El código fuente debe ser claro, con variables, funciones, módulos y clases descriptivas.
- **Eficiencia**. El programa debe optimizar el uso de recursos del sistema como memoria y CPU.
- **Documentación**. Incluir documentación adecuada y completa que explique cómo instalar y configurar y/o **Documentación Ética.** Desarrollar un manual de usuario que explique no sólo cómo usar el sistema, sino también las consideraciones éticas detrás de cada funcionalidad y cómo se protegen los datos del usuario.
- **Usabilidad**. La interfaz de consola debe ser intuitiva y permitir al usuario interactuar fácilmente con el programa.

Actividades Integradoras del Módulo

Actividad Integradora N° 2

Programación I

Basándose en los requerimientos funcionales y no funcionales del proyecto, se solicita inicialmente:

1. Realizar un **programa modular** que cumpla con los requerimientos generales del programa que son:
    a. Gestionar los dispositivos (listar, buscar, agregar y eliminar)
    b. Gestionar automatizaciones (sólo una).

    Nota: Almacenar la información en estructuras de datos (ej. listas, diccionarios)

Sugerencias para la escritura del código fuente *(convenciones de nomenclatura estándar - comunidad de Python):*

- **Archivos y Directorios**
    - Archivos Python: Utilizar el formato snake_case en minusculas. Ejemplo: *main.py, module_example.py.*
    - Directorios: Utilizar nombres en minúsculas sin espacios ni caracteres especiales. Ejemplo: docs, src.
- **Funciones**. Utilizar el formato snake_case en minúsculas comenzando la primera palabra con un verbo en infinitivo. Ejemplo: obtener_usuario, actualizar_datos.
- **Variables**. Utilizar el formato snake_case. Ejemplo: contador, nombre_usuario.
- **Constantes.** Utilizar el formato snake_case en mayúsculas. Ejemplo: MAX_INTENTOS.

Base de Datos

1. **DER:**
    a. Identificar las entidades necesarias para almacenar la información del sistema.
    b. Definir los atributos para cada entidad y sus tipos de datos.
    c. Definir claves candidatas
    d. Realizar diagrama E-R. No olviden definir la cardinalidad.

Ética y Deontología Profesional

1. **Documento de Protección de Datos:**
    a. Desarrollar una política de privacidad y protección de datos para el sistema SmartHome, considerando:
        i. Tipos de datos personales que se recolectarán
        ii. Finalidad del procesamiento de datos
        iii. Medidas de seguridad implementadas
        iv. Derechos de los usuarios sobre sus datos

Entregables

- **Programación I.** Link a repositorio en Github (u otro medio) + Documento PDF que explique la automatización elegida y su funcionamiento.

  *Nota: Se **valorará** código fuente que respeta las nomenclaturas estándar de la comunidad de Python. Cumplimiento de requerimientos no funcionales: modularidad, usabilidad y legibilidad.*

- **Base de Datos.** Documento que contenga las entidades identificadas, con sus respectivos atributos, donde se indique el tipo de dato de cada uno y cuales son claves candidatas. En el mismo documento incluir al final el diagrama E-R. El mismo lo pueden hacer con la herramienta https://app.diagrams.net/ , https://www.lucidchart.com/ o cualquiera que ustedes definan.

- **Ética y Deontología Profesional.** Documento PDF con la política de protección de datos y el plan de gestión de trabajo en equipo (4-5 páginas).