



DOMOHOME

Proyecto Final DAM de Francisco Salvador
González

Índice

Descripción	2
Android	2
Raspberry Pi	3
Justificación	4
Alcance	4
Alternativas	4
Stack Tecnológico	5
Objetivos	5
Requisitos	6
Requisitos funcionales	6
Requisitos no funcionales	6
Restricciones técnicas	6
Requisitos de interfaz	7
Casos de uso	7
Modelo entidad-relación	9
Modelo relacional	9

Descripción

DomoHome es un proyecto de domótica que permite controlar y automatizar diferentes aspectos de una casa o edificio.

La domótica se encarga de la automatización y el control de dispositivos y sistemas eléctricos y electrónicos, para aumentar la eficiencia energética, la seguridad y el confort del hogar.

Este proyecto de domótica utilizará diferentes tecnologías, entre ellas Android y Raspberry Pi, para crear una solución completa de automatización del hogar.

Android

En la parte de Android del proyecto, se diseñará una aplicación móvil que actuará como intermediario entre el usuario y la Raspberry Pi. Esto es debido a que configurar un sensor o un dispositivo inteligente, pudiendo además añadir rutinas para estos dispositivos, puede llegar a ser bastante complicado para un usuario promedio. Esta aplicación permitirá al usuario controlar y automatizar los diferentes sensores y dispositivos inteligentes de su hogar desde cualquier lugar y en cualquier momento.

Para comenzar a utilizar la aplicación, el usuario deberá registrarse en la aplicación a través de un formulario de registro. Tras rellenar el formulario, se guardará toda la información del usuario (Datos que se han rellenado en el formulario de registro). En cambio, si el usuario ya está registrado en la aplicación, existirá una opción para iniciar sesión en la que solo tendremos que rellenar 2 campos: el usuario y la contraseña. Esta limitación se puede incorporar como mejora de los usuarios es que cada usuario solo podrá tener una casa configurada (Se configura cuando el usuario se registra en la aplicación).

Tras haberse registrado o iniciado sesión, la aplicación redirigirá al usuario a la pantalla principal. Dicha pantalla contendrá el estado general de la casa, como por ejemplo la temperatura y humedad, si las luces están encendidas o apagadas, etc.

Uno de los apartados de la aplicación será para configurar un nuevo dispositivo inteligente. Un dispositivo inteligente es un dispositivo electrónico, por lo general conectado a otros dispositivos, en este caso a la Raspberry Pi, o a la red. Todos los nuevos dispositivos inteligentes quedarán guardados en la Base de Datos.

En el siguiente apartado, podremos observar y controlar los distintos dispositivos inteligentes que tengamos configurados. Los dispositivos que hayamos configurado con anterioridad, se cargarán en la aplicación.

Otro de los apartados será el de configuración. En este apartado podremos editar varios parámetros de un usuario, como la contraseña, el nombre de la casa del usuario, etc. También se podrán configurar varios apartados de la propia aplicación.

Otro de los apartados será el de programación de rutinas para los dispositivos inteligentes. Una rutina se encarga de realizar tareas repetitivas según estén programadas.

El último apartado será el de "Acerca de". Este apartado proporciona información sobre la aplicación en sí misma. Esta sección contendrá detalles sobre el propósito de

la aplicación, la versión actual, la fecha de lanzamiento, el desarrollador responsable de la aplicación, la política de privacidad, los términos de uso, entre otros casos.

Para la navegación entre apartados se usará un menú desplegable.

En un principio la app solo permitirá configurar los siguientes sensores y dispositivos inteligentes:

- **Sensores**
 - Sensores de temperatura y humedad
- **Dispositivos inteligentes**
 - Bombillas inteligentes
 - Televisiones inteligentes
 - Relés

En futuras actualizaciones se añadirá compatibilidad con multitud de sensores y dispositivos inteligentes, aunque esta primera versión sólo incluye estos.

En DomoHome existirán 2 roles de administrador: Usuario y Administrador.

- **Usuario:** Este rol podrá acceder a todas las funcionalidades de la aplicación, pero solo podrá editar parámetros de su propio usuario y solo sus sensores/dispositivos inteligentes. Por defecto todos los usuarios que se registren tendrán este rol.
- **Administrador:** Los usuarios que tengan este rol, podrán acceder y editar todas las configuraciones de todos los usuarios. Por motivos de seguridad, para agregar a un usuario este rol, se tendrá que editar de forma manual el usuario de la base de datos.

Raspberry Pi

Para el funcionamiento óptimo de nuestra Raspberry Pi, hemos decidido utilizar el sistema operativo Raspbian, una distribución de GNU/Linux basada en Debian. Este sistema operativo está orientado para la enseñanza de programación y electrónica. Raspbian cuenta con una licencia GPL, lo que significa que está disponible de forma gratuita para su uso y modificación por parte de la comunidad.

La versión específica de Raspbian que hemos elegido para nuestro proyecto es la que se lanzó el 21 de febrero de 2023. Usaremos la versión que cuenta con un entorno de escritorio para facilitar su uso y accesibilidad.

En cuanto a base de datos, hemos decidido utilizar MySQL, un popular gestor de base de datos de código abierto que es conocido por su fiabilidad y flexibilidad. En ella almacenaremos la información de los usuarios registrados en nuestra aplicación, la configuración de los sensores y dispositivos inteligentes, la casa, los valores de temperatura y humedad recopilados en diferentes momentos del día y las rutinas configuradas para automatizar el hogar.

Para acceder a la base de datos, hemos optado por utilizar una API REST programada en Java, utilizando el framework Spring. La API permitirá a la aplicación interactuar con la base de datos de manera segura y eficiente, asegurando la privacidad y confidencialidad de los datos.

Finalmente, para el control de los sensores y dispositivos inteligentes, hemos decidido crear scripts en Python. Estos scripts estarán preparados para recibir parámetros específicos para controlar un sensor o dispositivo inteligente en particular. De esta manera, podremos gestionar y automatizar el hogar de forma más precisa y

personalizada. Dichos scripts los ejecutaremos desde Java con los parámetros que enviemos desde la aplicación de Android.

Justificación

Durante estos últimos años, el ámbito tecnológico ha dado un gran avance. Esto hace que la domótica llegue cada vez a más casas gracias a los dispositivos inteligentes. Con nuestra App, el cliente podrá ver información sobre su casa gracias a los sensores, como, por ejemplo, la temperatura y la humedad de una habitación; y poder controlar varios dispositivos y electrodomésticos, como por ejemplo la luz de las habitaciones, la televisión, etc.

Alcance

DomoHome estará dirigida sobre todo a gente un poco más experimentada en el mundo de la domótica, ya que, si quiere añadir un nuevo sensor, solo en caso de que el sensor se conecte por cable, deberá conectarlo a unos pines en específicos y después configurarlo. Aunque también podrán usarlo gente poco experimentada, ya que también se podrán configurar distintos dispositivos inteligentes a partir de la aplicación.

Alternativas

Existen diversas alternativas a DomoHome, como Mi Home, para dispositivos inteligentes de Xiaomi; SmartThings, para dispositivos inteligentes Samsung; Tuya Smart, para dispositivos inteligentes de diferentes marcas. Algunas de estas aplicaciones también ofrecen la opción de establecer rutinas, pero en ninguna de ellas viene incorporada la opción de configurar sensores a partir de una Raspberry Pi.

Stack Tecnológico

En este proyecto estamos usando diferentes dispositivos y componentes:

- **Raspberry Pi:** Usaremos una Raspberry Pi 4 modelo B para este proyecto. Tendrá las siguientes características:
 - Procesador Broadcom BCM2711 SoC (System on Chip) de 64 bits A72 (ARM v8)
 - SDRAM 8GB LPDDR4
 - Bluetooth 5.0
 - Gigabit Ethernet
 - 2 puertos USB 2.0 y 2 puertos USB 3.0
 - Conector GPIO de 40 contactos (Para conectar los sensores, ventilador, etc.)
 - Ranura para insertar tarjeta microSD para usarla como almacenamiento (En este caso usaremos una tarjeta de 64GB)
- **Sensor DHT 11:** Este es un sensor que sirve para medir la temperatura y humedad relativa en el aire. Es un sensor digital y usa un solo pin de datos para la comunicación. El sensor DHT11 tiene una precisión de $\pm 2^{\circ}\text{C}$ en la medición de temperatura y del $\pm 5\%$ en la medición de humedad. El sensor cuenta con un rango de temperatura de 0°C a 50°C y de humedad relativa de 20% a 90%.

Objetivos

- Controlar y automatizar diferentes aspectos de una casa o edificio a partir de la domótica.
- Usar Android y Raspberry Pi para crear un proyecto sobre la automatización del hogar.
- Diseñar una aplicación para dispositivos Android que permita al usuario controlar y automatizar los diferentes sensores y dispositivos inteligentes desde cualquier lugar y en cualquier momento.
- Facilitar la configuración de los sensores y dispositivos para que lo pueda realizar un usuario promedio.
- Permitir al usuario registrarse o iniciar sesión en la aplicación y tener acceso a sus sensores y dispositivos inteligentes, así como poder editar su perfil de usuario.
- Tener un sistema de roles que permita a la aplicación tener diferentes niveles de acceso a la aplicación.

Requisitos

Requisitos funcionales

- REQF-1. La aplicación debe permitir Registrar un nuevo usuario
- REQF-2. La aplicación debe permitir Iniciar sesión con un usuario ya existente
- REQF-3. Tras el Registro/inicio de sesión, la aplicación debe redirigir al apartado principal
- REQF-4. La aplicación debe permitir navegar entre los diferentes apartados de la aplicación
- REQF-5. La aplicación debe mostrar la temperatura y la humedad de la habitación del usuario
- REQF-6. La aplicación debe permitir configurar un nuevo dispositivo inteligente
- REQF-7. La aplicación debe mostrar los dispositivos inteligentes ya configurados
- REQF-8. La aplicación debe permitir editar parámetros del usuario
- REQF-9. La aplicación debe permitir configurar rutinas para automatizar los sensores/dispositivos inteligentes
- REQF-10. Los administradores deben poder ver la información de todos los usuarios, así como editarlos.

Requisitos no funcionales

- REQNF-1. El usuario debe haber introducido todos los datos para registrarse o iniciar sesión
- REQNF-2. El usuario debe haber introducido todos los datos para configurar el nuevo dispositivo inteligente
- REQNF-3. La contraseña no se guardará en texto plano en la base de datos, sino que se guardará en un hash con el algoritmo SHA-1

Restricciones técnicas

- RT-1. La parte de Android se desarrollará en Android Studio
- RT-2. La versión de Android mínima en la que debe funcionar la aplicación es Android 7
- RT-3. El Sistema Gestor de Bases de Datos usado será MySQL versión 8.0.32
- RT-4. La API Rest se desarrollará en IntelliJ y estará alojada en la Raspberry Pi (En la Raspberry solo se ejecutará el JAR)
- RT-5. El Sistema Operativo que usará la Raspberry Pi será Raspbian de 64 bits con entorno de escritorio. La versión será la que se lanzó el 21 de febrero de 2023.
- RT-6. Los scripts de control de sensores se desarrollarán en Python. La versión de Python será Python 3.11.2

Requisitos de interfaz

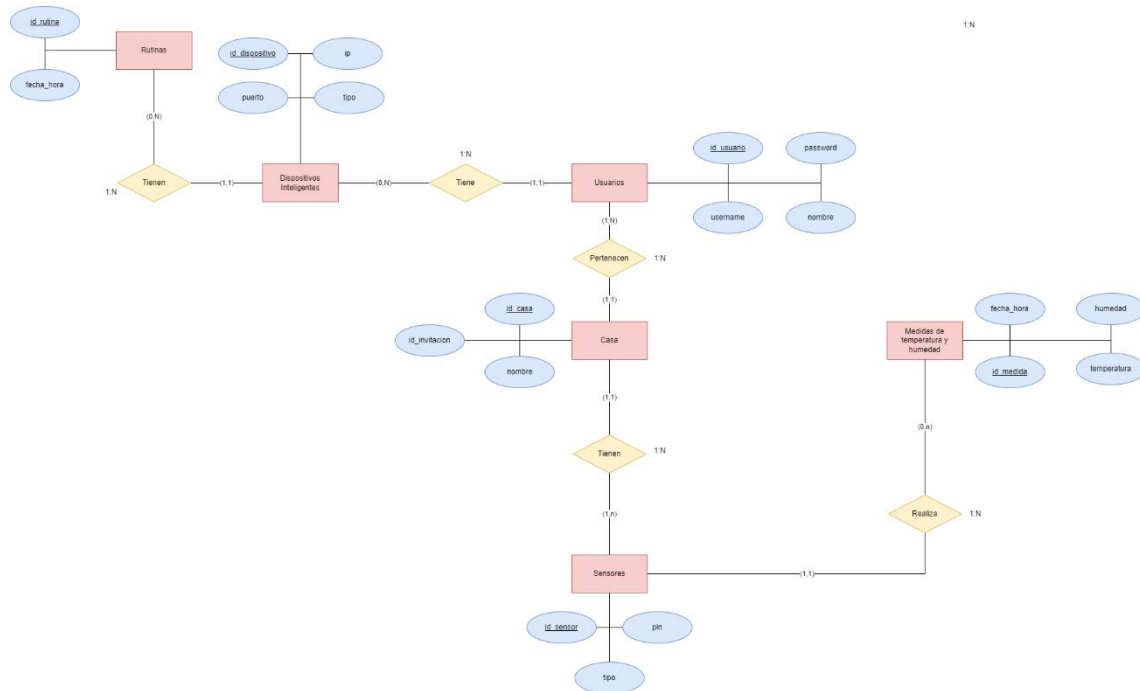
- REQUI-1. La interfaz solo tendrá colores agradables a la vista y que tengan buen contraste.
- REQUI-2. En el apartado de registro aparecerán los campos de texto necesarios junto a sus títulos. También aparecerán los botones Registrarse e iniciar sesión.
- REQUI-3. Para navegar entre los diferentes apartados de la aplicación, tendremos un botón arriba a la izquierda de la aplicación que abrirá un menú
- REQUI-4. En el apartado iniciar sesión aparecerán los campos de texto necesarios junto a sus títulos. También aparecerá el botón iniciar sesión.
- REQUI-5. En el apartado de “Acerca de” aparecerá un texto con la información correspondiente.

Casos de uso

CU-01	Registrarse en la aplicación
Versión	1.0
Dependencias	N/A
Precondición	El dispositivo donde se encuentre la aplicación debe tener conexión a internet
Descripción	La aplicación deberá comportarse como se describe en el siguiente caso de uso cuando el usuario que use la aplicación solicite crear un nuevo usuario.
Secuencia normal	<ol style="list-style-type: none">1. El usuario debe rellenar los datos del formulario de registro que aparece nada más abrir la aplicación2. El usuario debe pulsar el botón registrarse
Postcondición	El nuevo usuario debe haberse insertado en la Base de Datos y la aplicación debe redirigirse al apartado principal.
Excepciones	<ul style="list-style-type: none">• Si el usuario no ha rellenado todo el formulario y pulsa el botón registrarse, debe saltar un mensaje de error diciendo que debe rellenar todo el formulario• Si el usuario no tiene conexión a internet debe saltar un mensaje de error diciendo que no tiene conexión a internet• Si el usuario ya existe en la base de datos, debe saltar un mensaje de error diciendo que el usuario ya existe en la base de datos

CU-02	Iniciar sesión en la aplicación
Versión	1.0
Dependencias	N/A
Precondición	<ul style="list-style-type: none"> • El dispositivo donde se encuentre la aplicación debe tener conexión a internet • Debe existir el usuario con el que queremos iniciar sesión en la base de datos
Descripción	La aplicación deberá comportarse como se describe en el siguiente caso de uso cuando el usuario que use la aplicación inicie sesión.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario debe pulsar en iniciar sesión, que se encuentra debajo del botón de registrarse en el apartado de registro 2. Una vez estemos en el apartado de iniciar sesión, rellenamos el formulario de inicio de sesión
Postcondición	La aplicación debe consultar si el usuario existe en la base de datos y en el caso de que el usuario exista, la aplicación debe redirigir al apartado principal
Excepciones	<ul style="list-style-type: none"> • Si el usuario no ha rellenado todo el formulario y pulsa el botón iniciar sesión, debe saltar un mensaje de error diciendo que debe rellenar todo el formulario • Si el usuario no tiene conexión a internet debe saltar un mensaje de error diciendo que no tiene conexión a internet • Si el usuario no existe, debe saltar un mensaje de error diciendo que el usuario indicado no existe • Si la contraseña no coincide, debe saltar un mensaje de error diciendo que la contraseña es incorrecta

Modelo entidad-relación



Modelo relacional

USUARIOS (id_usuario, username, password, nombre, **id_casa**)

CASAS (id_casa, nombre)

SENSORES (id_sensor, pin, tipo, **id_casa**)

MEDIDAS_TEMPERATURA_HUMEDAD (id_medida, temperatura, humedad, fecha_hora, **id_sensor**)

DISPOSITIVOS_INTELIGENTES (id_dispositivo, ip, puerto, tipo, **id_usuario**)

RUTINAS (id_rutina, fecha_hora, **id_dispositivo**)