



UNIVERSITAT  
POLITÀCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Diseño de una arquitectura IoT para el control domótico basado en Arduino

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Jorge Pérez Fernández

**Tutor:** Sergio Sáez Barona

2016-2017

## Resumen

Este proyecto se centra en crear los elementos necesarios para el control domótico de una casa. Por lo cual se desarrollarán tres roles diferentes los cuales tendrán diferentes funciones para posibilitar el control domótico.

Un cliente que va a ser una aplicación móvil creada para poder ser ampliada a diferentes dispositivos. Un servidor con diferentes funcionalidades y servicios para poder dar soporte a los diferentes roles del escenario. Y para finalizar unos actuadores basándose en una modalidad de Arduino.

Uno de los objetivos añadidos a este proyecto va a ser que los componentes sean lo suficientemente sencillos de instalar para que cualquier tipo de usuario pueda hacerlo. Además del objetivo anterior se añadirá que pueda ser usado con la menor restricciones posibles.

**Palabras clave:** Arduino, Android, Servicios REST, Domótica.

## Summary

This project focuses on creating the necessary elements for the home control of a house. So, it develops three different roles which will have different functions to enable home control.

A client that will be a mobile application created to be able to be extended to different devices. A server with different functionalities and services to be able to support the different roles of the scenario. And to finish some actuators based on an Arduino mode.

One of the objectives added to this project will be that the components are simple enough to install so that any type of user can do it. In addition to the above objective it was added that it can be used with as few restrictions as possible.

Key words: Arduino, Android, Services REST, Home automation

## Agradecimientos

Me gustaría agradecer a todos mis compañeros la ayuda aportada en toda la etapa universitaria y el apoyo que he recibido de ellos.

Además, también quiero agradecerle a mi madre el esfuerzo que ha hecho para que llegue hasta aquí.

# TABLA DE CONTENIDOS

## Contenido

|       |  |    |
|-------|--|----|
| 1.    | Introducción.....                        | 5  |
| 2.    | Estudio de mercado.....                  | 7  |
| 2.1   | Apple home .....                         | 7  |
| 2.2   | Mi Home .....                            | 8  |
| 2.3   | Aspectos destacables.....                | 9  |
| 3     | Especificación requisitos software ..... | 10 |
| 3.1   | Introducción .....                       | 10 |
| 3.2   | Propósito.....                           | 10 |
| 3.3   | Alcance.....                             | 10 |
| 3.4   | Descripción general .....                | 11 |
| 3.5   | Restricciones.....                       | 11 |
| 4     | Análisis.....                            | 13 |
| 4.2   | Especificación Diseño .....              | 14 |
| 4.3   | Casos de uso.....                        | 15 |
| 5     | Diseño.....                              | 18 |
| 5.2   | Capa Persistencia .....                  | 18 |
| 5.3   | Capa Presentación .....                  | 21 |
| 5.4   | Capa Lógica.....                         | 26 |
| 6     | Implementación .....                     | 28 |
| 6.2   | Herramientas utilizadas .....            | 28 |
| 6.3   | Lenguajes utilizados .....               | 29 |
| 6.4   | Tecnologías usadas .....                 | 29 |
| 6.5   | Arquitectura.....                        | 30 |
| 6.6   | Comunicación .....                       | 30 |
| 6.7   | Programa de control .....                | 31 |
| 6.7.1 | Mule.....                                | 31 |
| 6.8   | Nodos.....                               | 33 |
| 6.9   | Aplicación Android .....                 | 34 |
| 6.10  | Servidor.....                            | 34 |
| 6.11  | Librerías usadas.....                    | 35 |
| 7     | Pruebas.....                             | 36 |
| 8     | Conclusiones .....                       | 47 |
| 9     | Bibliografía.....                        | 48 |



# ÍNDICE DE IMÁGENES

|                                   |    |
|-----------------------------------|----|
| Apple Home 2-1 .....              | 7  |
| Mi home 2-2 .....                 | 8  |
| Esquema del sistema 4-1 .....     | 13 |
| Esquema del diseño 4-2 .....      | 14 |
| Esquema base de datos 5-1 .....   | 18 |
| Tabla temperaturas 5-2 .....      | 19 |
| Tabla sensores 5-3 .....          | 19 |
| Tabla habitaciones 5-4 .....      | 20 |
| Tabla Arduinos 5-5 .....          | 20 |
| Menú Principal 5-6 .....          | 22 |
| Vista estancias 5-7 .....         | 22 |
| Vista acciones 5-8 .....          | 23 |
| Vista información 5-9 .....       | 23 |
| Vista ajustes 5-10 .....          | 24 |
| Vista ajustes wifi 5-11 .....     | 24 |
| Vista mule 6-1 .....              | 31 |
| Acciones 6-2 .....                | 32 |
| Nodemcu 6-3 .....                 | 33 |
| Menú principal 7-1 .....          | 36 |
| Menú ajustes 7-2 .....            | 37 |
| Cambiar ip 7-3 .....              | 37 |
| Mensaje de confirmación 7-4 ..... | 38 |
| Añadir estancia 7-5 .....         | 40 |
| Seleccionar bluetooth 7-6 .....   | 41 |
| Formulario wifi 7-7 .....         | 42 |
| Banner de carga 7-8 .....         | 42 |
| Selección de estancia 7-9 .....   | 43 |
| Confirmación 7-10 .....           | 43 |
| Alias sensor 7-11 .....           | 44 |
| Acciones 7-12 .....               | 45 |
| Información 7-13 .....            | 47 |

# 1. Introducción

---

En el mundo se está generando la tendencia de convertir los objetos tradicionales en Smart o inteligente, esto se puede resumir dándoles más funcionalidad que las que tienen tradicionalmente. Todo esto se hace para así hacernos la vida más sencilla y cómoda. Esta idea es hoy posible gracias a abaratamiento en la tecnología y la facilidad de obtención de estos componentes, además como móviles y demás dispositivos del estilo.

Podemos encontrar un ejemplo de esta tendencia en el mundo del automovilismo que a los coches de siempre se les está intentando añadir más funcionalidades conectándolos a los móviles y mediante eso también a internet.

Bajo ese contexto se plantea este proyecto por el cual se intenta dar esa funcionalidad a los objetos habituales en una casa como por ejemplo en persianas, ventanas, iluminación etc. Y así satisfacer las necesidades que se crean en el mundo actual.

Uno de los requisitos que se proponen es facilitar la agregación de estas funcionalidades a los objetos adaptados sin requerir conocimientos informáticos o técnicos, ya que una vez hecha la instalación el usuario podrá fácilmente cambiar los nodos.

Por lo tanto, los nodos que se han diseñado son fácilmente configurables y así simplemente con el asistente que tenemos en el dispositivo controlador estos se puedan configurar.

Los componentes y tecnologías que se han decidido usar están buscando orientarse al mundo de licencia libre o de estándares para que así se puede implementar en diversos dispositivos así se garantiza la compatibilidad y por otro lado al usar licencia libre podemos modificarlos para las necesidades de cada momento.

## ***ESTRUCTURA DE LA MEMORIA***

A continuación, se comentará lo que se va a ver en cada punto de la memoria

En el punto segundo se encuentra el estudio de mercado donde se comparan dos de las diferentes soluciones encontradas para satisfacer las mismas necesidades que se quiere en este proyecto, cada una de estas aplicaciones a comparar pertenecen a las dos plataformas más usadas en dispositivos móviles y tables.

En el punto tercero se encuentra la especificación de requisito software en este punto se describirá los diferentes requisitos que tiene que cumplir la aplicación, así como los que no puede cumplir y las posibles restricciones de estas.

En el cuarto punto se encuentra el Análisis se describe cada uno de los roles que habrá en el proyecto a parte de la explicación de cada uno de los diferentes casos de uso. A parte de esto también se describirá la unión entre el software y el hardware.

En el quinto punto podemos encontrar el diseño se presenta los tres capas o niveles en los cuales se ha disgregado el desarrollo del proyecto.

En el penúltimo punto el número sexto se encuentra la implementación. En esta sección se encuentra las diferentes decisiones que se han tomado para el desarrollo del proyecto, así como algunas explicaciones de sus funcionalidades.

Para finalizar esta el punto séptimo pruebas en este punto mediante capturas de pantalla se va a explicar algunos de los casos de uso que se han implementado, así como una breve descripción de lo que se va viendo en cada uno de los ejemplos.

## 2. Estudio de mercado

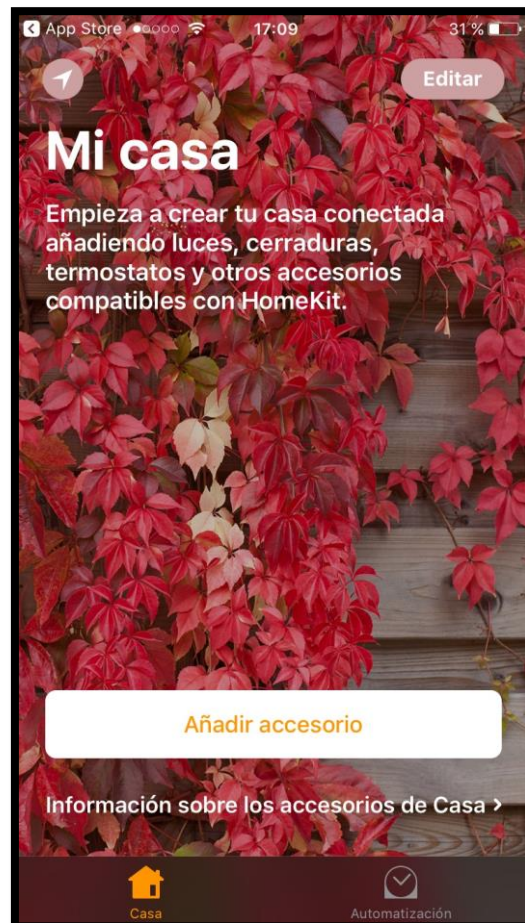
---

Un factor importante a la hora de ver en que se puede mejorar o cuáles son los déficits que hay en el mercado, es simplemente viendo lo que hay en este mercado actualmente. Por lo tanto, lo que se ha realizado es una busca de las variantes que hay actualmente en el mercado.

En este punto solo se ha indicado un ejemplo para móviles procedentes de iOS y otro ejemplo para dispositivos de Android.

### 2.1 Apple home

Apple home es la solución de Apple para el control domótico. Desde esta aplicación se puede controlar la casa con los diferentes dispositivos que tengamos conectada a ella, algunas de los déficits que se han encontrado son por ejemplo que solo los dispositivos con versión mínima de iOS 10 tiene la posibilidad de usar Apple home.



APPLE HOME 2-1

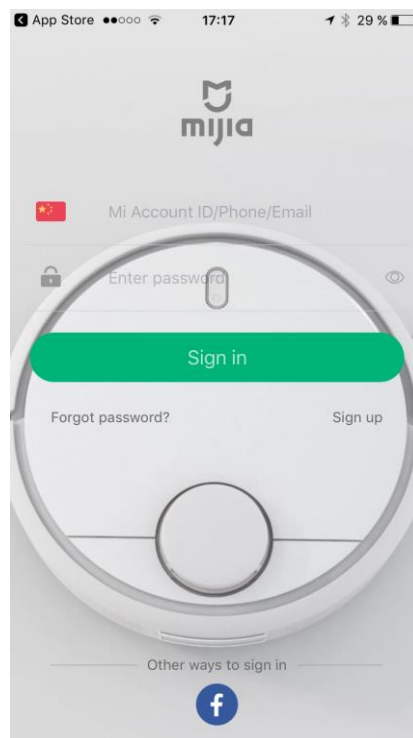
También se considera que las siguientes características son un inconveniente para el usuario final, por ejemplo, que solo se puedan conectar los dispositivos que cumplan el estándar home kit.

Además, los dispositivos los tienes que vincular con una cuneta de iTunes y hacer uso de sus servicios.

Una de las grandes desventajas que podemos destacar, es que solo se pueden ser controlados desde dispositivos de esta marca por lo tanto estamos limitando los posibles “controladores”.

## 2.2 Mi Home

Mi Home es la solución de Xiaomi para el control domótico.



**MI HOME 2-2**

Esta solución está pensada para dispositivos de que sean manejado por iOS o Android. Este es un punto favorable ya que podemos tener distintos controladores sin que nada nos restrinja el acceso.

Pero uno de los grandes inconvenientes que podemos encontrar es que la aplicación de control no está adaptada a todos los países y por lo tanto el idioma es un gran problema, pero no solo la aplicación no está adaptada a otros idiomas si no que los complementos están pensados para el país de origen, por lo tanto, para el resto de países es necesario hacer uso de diferentes adaptadores.



## 2.3 Aspectos destacables

Como se ha podido observar en los datos aportados anteriormente podemos encontrar que existen algunas deficiencias en el mercado sobre todo en el aspecto de la compatibilidad y la multiplataforma.

Por lo tanto, lo que se busca es conseguir un programa que sea multiplataforma y que además sea fácilmente adaptable los componentes a cada país.

Por lo tanto, lo que se ha intentado es libertad a la hora de elegir la plataforma de control independientemente del sistema operativo que se posea.

Además, se ha intentado usar en la mayor manera posible tecnologías y productos basados en licencias abiertas, ya que al usar este tipo de licencias se puede reducir los costes de estos mismos y dar la posibilidad de ser modificados para que se adapten a nuestras necesidades de cada momento.

Además, también se da la libertad de que diferentes desarrolladores implanten el control de la manera que ellos convengan más adecuado y de la misma manera los nodos lo único que tienen que cumplir es con un api de comunicación estándar.

## 3 Especificación requisitos software

---

A continuación, se va a realizar una especificación de requisitos software que deberá de cumplir la aplicación para dar el visto bueno a esta fase de desarrollo, para el desarrollo de este apartado se ha baso en el documento *IEEE Recommended Practice for Software Requirements Specifications*, se podrá encontrar más información sobre este documento en el apartado de bibliografía.

### 3.1 Introducción

En este apartado lo que se indicara cada uno del punto que se van a tratar este índice que da en el contexto del Apartado de Especificación de requisitos software

- 3.2 Propósito
- 3.3 Alcance
- 3.4 Descripción general
  - 3.4.1 Perspectivas del producto
  - 3.4.2 Funciones del producto
- 3.5 Restricciones
  - 3.5.1 Políticas reguladoras
  - 3.5.2 Limitaciones de hardware
  - 3.5.3 Interacciones con otras aplicaciones
  - 3.5.4 Requisitos de lenguaje
  - 3.5.5 Requisitos de fiabilidad
  - 3.5.6 Seguridad

### 3.2 Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales y del sistema para el desarrollo del control domótico tanto las aplicaciones que lo van a controlar como las aplicaciones o nodos que realizaran estas acciones que se pidan.

### 3.3 Alcance

El sistema domótico tendrá tres roles por lo cual los tres roles podrán comunicarse entre si para poder hacer así las tareas designadas. La aplicación de control será una aplicación móvil que funcionará de forma nativa en diferentes dispositivos y nos dará apoyo a los siguientes procesos:

- Añadir nuevo nodo
- Mandar acciones a los nodos
- Añadir estancias
- Ver información estadística y actual del sistema

El servidor nos suministrar la persistencia deseada y la comunicación entre los diferentes elementos.

El ultimo rol va a ser el de los nodos que serán un elemento físico que realizara las tareas deseados por el usuario además de recolectar datos para el sistema.

Con este esquema tendremos una flexibilidad de acceso mayor ya que al estar en diferentes plataformas móviles y poder tener diferentes controladores podremos

realizar las tareas más cómodamente, y además poder ser controlado desde fuera de la casa.

Otro de los puntos que cumplirá es que los diferentes nodos se comunicaran con el servidor de forma inalámbrica. Para así poder conseguir mejor flexibilidad y comodidad a la hora de la instalación o de la sustitución de uno de ellos.

A través de la aplicación no se podrá eliminar un nodo o estancia ya añadida al sistema, por lo tanto, una vez el sistema se empiece a configurar solo podemos añadir nunca modificar o eliminar elementos.

## 3.4 Descripción general

Como se va a ver se describirá de manera genérica lo que la aplicación podrá hacer y las diferentes posibilidades que abarca. Así como lo que la aplicación no va a poder hacer.

### 3.4.1.1 Perspectivas del producto

El sistema domótico será un producto diseñado para para trabajar en múltiples plataformas lo que permitirá adquirir la que más nos convenga, además será un sistema centralizado. También permitirá la coexistencia de diferentes nodos y de diferentes aplicaciones controladoras.

### 3.4.2 Funciones del producto

En este apartado se va a ver las funcionalidades que va a tener nuestro proyecto. Por lo cual se ha creado dos agentes distintos, por un lado, tenemos el agente usuario que va a ser la persona que controlara de forma manual la casa y por otro lado tenemos el agente sistema, que se encargará de las tareas automáticas que estén programadas además de añadir opciones al agente usuario, además de esto el agente sistema será el encargado de unir al agente usuario a los nodos para poder realizar las tareas correspondientes.

#### **Usuario**

- El usuario podrá encender remotamente dispositivos de iluminación.
- El usuario podrá tener información sobre temperaturas de diferentes estancias.
- El usuario podrá añadir nuevos dispositivos al Sistema y ponerlos en una estancia concreta.
- El usuario podrá añadir nueva estancia.
- El usuario podrá configurar aspectos básicos del sistema.

#### **Sistema**

- El Sistema obtendrá información automáticamente.
- El sistema dará la posibilidad de ser controlado desde el exterior de la casa.
- El sistema será capaz de auto armarse en caso de reinicio.

## 3.5 Restricciones

En este apartado se indicará las restricciones a la cual está sometida las aplicaciones en todo su conjunto no solo a la aplicación que controla el usuario si no también las restricciones que afecta a los diferentes roles que tenemos en el sistema completo.

### 3.5.1 Políticas reguladoras

La aplicación se desarrollará mediante software y hardware de licencias abierta por lo tanto no se deberá pagar por el uso de: Base de datos (MYSQL), elementos actuadores(Arduinos).

A excepción del controlador del servidor que va a ser el único elemento por el cual se tendrá que pagar ya que no va a tener una licencia GPL.

### 3.5.2 Limitaciones de hardware

Para cada uno de los diferentes roles que tenemos en el sistema será necesario cumplir unos requisitos hardware que se van a detallar a continuación.

#### **Controladores:**

Serán necesarios elementos que tengan la capacidad de comunicación mediante bluetooth y wifi

#### **Servidores:**

Necesitamos un dispositivo que se le puedan instalar un MySQL, desplegar el proyecto mule y tenga alguna comunicación del protocolo Ethernet.

### 3.5.3 Interacciones con otras aplicaciones

Ya que el sistema una de las grandes características que tiene es que es capaz de comunicarse con otros agentes de diferente plataforma se ha establecido que las conexiones sean basadas en estándares de comunicación.

### 3.5.4 Requisitos de lenguaje

Todo el sistema tiene que estar en el mismo idioma y para poder ser instalada en ese país tiene que estar traducida a su totalidad en el lenguaje dominante de este.

### 3.5.5 Requisitos funcionales

El sistema permitirá al usuario indicar y cambiar la ip en el controlador referente al servidor

El sistema recolectará datos dado un cierto intervalo de tiempo.

A cada Arduino se le asignará un id.

Cada ip solo estará asociada a un arduino.

Cada sensor debe estar vinculado con un Arduino.

El software servidor podrá ser usado en sistemas operativos Windows, Linux y OSX

### 3.5.6 Requisitos de fiabilidad

Todo el sistema tiene que ser capaz de rearmarse en caso de reinicio sin la interacción del usuario.

### 3.5.7 Seguridad

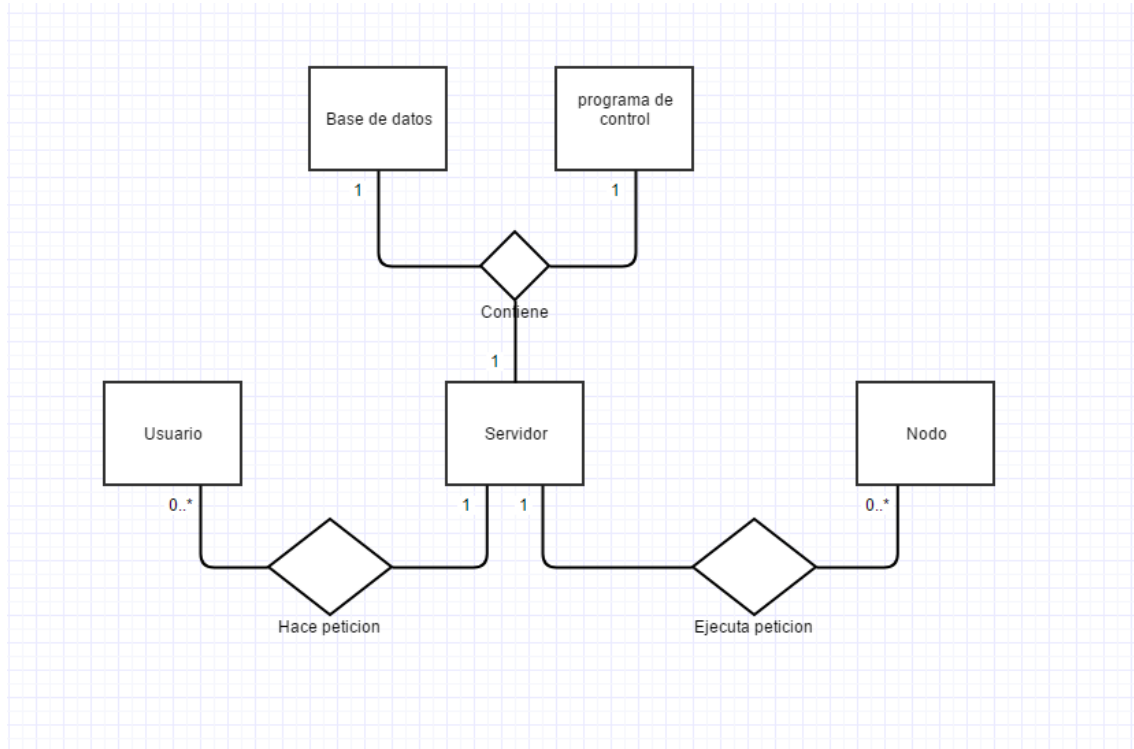
El tema de la seguridad se le pasará a cargo de la tecnología usada para la comunicación de esta en el caso del control desde dentro de la casa, en el caso del control desde la exterior queda fuera de este primer objetivo implantar la seguridad desde esta modalidad.

### 3.5.8 Atención y dependencias

En el caso que se quiera controlar la casas desde el exterior hay que configurar un reenvío de puertos desde el router al servidor en este caso sería del puerto 8180.

## 4 Análisis

En este punto se va a describir de forma genérica los diferentes roles que hay en el escenario y como estos se relacionan entre si, además. También indicando cuantos son el número máximo y mínimo que puede haber en el sistema.



**ESQUEMA DEL SISTEMA 4-1**

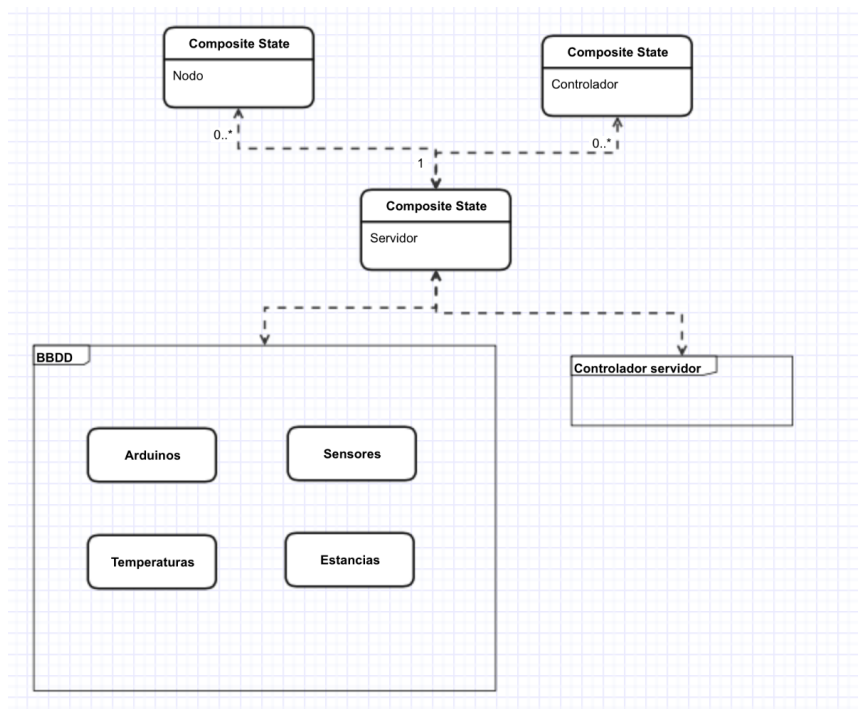
Los nodos son los elementos que actúan sobre el medio físico y además también tienen la posibilidad de recolectar información del medio donde están ubicados. Por lo tanto, van ser uno de los componentes más abundantes en el escenario ya que se podrán colocarse varios y en distintos lugares.

El servidor es el nexo de unión entre los diferentes elementos. También brinda con la característica de la persistencia y la posibilidad de control externo de la casa. Uno de los aspectos que hay que tener en cuenta es que solo puede haber uno, ya que el sistema no está diseñado como un sistema distribuido si no como uno centralizado.

Para finalizar tenemos el controlador que es el medio por el cual el usuario podrá interactuar con la casa tanto mandando ordenes como recibiendo información del estado actual de esta. Este tipo de controladores habrá distintos simultáneamente y no solo esto, sino que además pueden ser multiplataforma.

## 4.2 Especificación Diseño

Lo que vamos a ver a continuación es la unión entre el *software* y el *hardware*, desde un punto más genérico más adelante se verá con mayor detalle.



ESQUEMA DEL DISEÑO 4-2

Como se aprecia en el diagrama, dentro del servidor podemos diferenciar dos aplicaciones o servicios.

El primero que vamos a ver es la base de datos, más adelante se describirá con más detalle la implantación y la tecnología usadas para la creación de esta, en la base de datos encontramos cuatro tablas tanto para el control del sistema como la persistencia de datos.

Estas tablas son necesarias ya que así podemos tener una representación de la casa más dinámica a la hora de la configuración y por otro lado también mantener un historial de los datos recogidos por los distintos sensores.

En segundo lugar, encontramos controlador servidor que es el programa que nos proporciona servicios de conexión entre los dispositivos y suministra información a quien se lo pida, pero este servicio lo veremos más detalladamente en un punto posterior.

Eso referente al apartado software que posee el servidor.

Ahora referente del software dentro los nodos también tendremos un programa para la ejecución de las tareas deseadas que se verá con más detalle en la sección de implementación.

Para finalizar el controlador en este escenario será un programa creado para una plataforma en concreto que será el encargado de acercar al usuario al sistema además de también de mostrarle información.

### 4.3 Casos de uso

En los casos de uso vamos a ver como se desarrollarían en el sistema los puntos especificados en la especificación de requisitos software. Todos estos puntos se ha creado las acciones que tendría que realizar el usuario y el sistema para realizar estas acciones.

|                |  |  |
|----------------|--|--|
| Especificación | El usuario podrá encender remotamente dispositivos de iluminación.   |  |
| Roles          | Usuario  | Sistema  |
| Procedimiento  | <ul style="list-style-type: none"> <li>→ Acciones</li> <li>→ Estancia</li> <li>→ En esa vista seleccionar la luz que desea encender</li> </ul> | <ul style="list-style-type: none"> <li>→ Dar las diferentes estancias</li> <li>→ Dar las acciones que están vinculadas con esa estancia</li> <li>→ Mandar al nodo correspondiente la acción</li> </ul> |
| Observaciones  | Dependiendo que tipo de luz en la interface le puede aparecer o un interruptor o un slider.  |  |

|                |  |   |
|----------------|--|---|
| Especificación | El usuario podrá tener información sobre temperaturas de diferentes estancias  |   |
| Roles          | Usuario  | Sistema   |
| Procedimiento  | <ul style="list-style-type: none"> <li>→ Información</li> <li>→ Estancia</li> <li>→ En esa vista se mostrará la información deseada</li> </ul> | <ul style="list-style-type: none"> <li>→ Dar las diferentes estancias</li> <li>→ Dar la información que esta vinculadas con esa estancia</li> </ul> |
| Observaciones  |  |   |

Diseño de una arquitectura IoT para el control domótico basado en Arduino

|                |   |   |
|----------------|---|---|
| Especificación | El usuario podrá añadir nuevos dispositivos al Sistema y ponerlos en una estancia concreta.   |   |
| Roles          | Usuario   | Sistema   |
| Procedimiento  | <ul style="list-style-type: none"> <li>→ Ajustes</li> <li>→ Nuevo dispositivo</li> <li>→ Seleccionar el nombre bluetooth del dispositivo</li> <li>→ Vincular con el nodo</li> <li>→ Introducir Wifi y contraseña</li> <li>→ Continuar con el asistente</li> </ul> | <ul style="list-style-type: none"> <li>→ Mandar la información de ese nodo</li> <li>→ Almacenar la información que el asistente a recopilado del usuario</li> </ul> |
| Observaciones  | Para la conexión inicial necesario que el nodo indicado tenga la posibilidad de comunicación bluetooth.   |   |

|                |   |         |
|----------------|---|---------|
| Especificación | El usuario podrá añadir nueva estancia.   |         |
| Roles          | Usuario   | Sistema |
| Procedimiento  | <ul style="list-style-type: none"> <li>→ Ajustes</li> <li>→ Nueva Estancia</li> </ul>                                   |         |
| Observaciones  | Esta estancia no está visible en la sección de estancias hasta que coloquemos algún nodo en ella mediante el asistente. |         |

|                |   |         |
|----------------|---|---------|
| Especificación | El usuario podrá configurar aspectos básicos del sistema.                       |         |
| Roles          | Usuario   | Sistema |
| Procedimiento  | <ul style="list-style-type: none"> <li>→ Ajustes</li> <li>→ Nueva Ip</li> </ul> |         |
| Observaciones  | Para la ip solo se ha considerados IPs de tipo c.                               |         |



|                |  |
|----------------|--|
| Especificación | El Sistema obtendrá información automáticamente.   |
| Roles          | Sistema  |
| Procedimiento  | Cada 5 minutos el sistema obtendrá mediante un barrido toda la información sobre los sensores. |
| Observaciones  |  |

|                |  |
|----------------|--|
| Especificación | El sistema dará la posibilidad de ser controlado desde el exterior de la casa.   |
| Roles          | Usuario  |
| Procedimiento  | Tendría que el usuario redirigir el puerto 8081 del router de entrada al servidor de la red ya que si no hace este paso no podrá ser controlado desde el exterior de la vivienda |
| Observaciones  |  |

|                |   |
|----------------|---|
| Especificación | El sistema será capaz de auto armarse en caso de reinicio.  |
| Roles          | Sistema   |
| Procedimiento  | Todos los datos para el rearme están en memorias no volátiles y configurables.  |
| Observaciones  | Este punto está más pensado sobre todo en los nodos ya que son los elementos menos “inteligente” de los demás componentes de la infraestructura. Por lo tanto, en el momento que vuelva a recibir alimentación tiene que ser capaz de volver a un sistema estable sin la ayuda del usuario. |

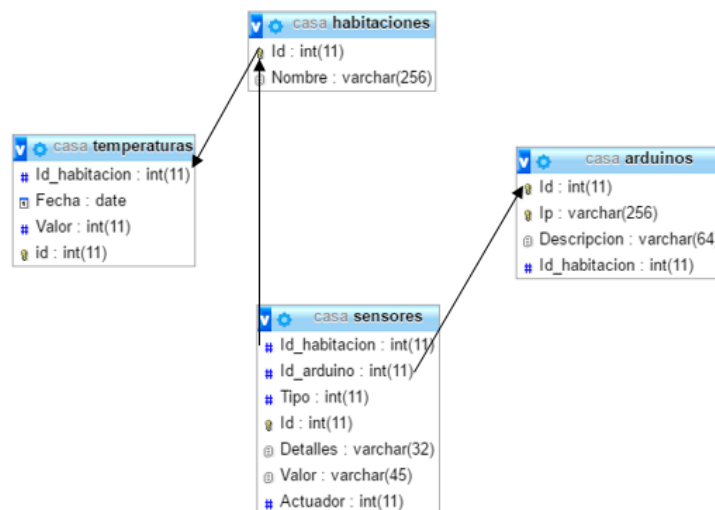
## 5 Diseño

En esta sección se mostrar los aspectos referentes al diseño que se ha tomado para la realización del proyecto abarcando diferentes aspectos como la persistencia, presentación y lógica cada una de estas secciones se detallará el motivo por el cual se ha decidido usar y como se han diseñado para ser usadas

### 5.2 Capa Persistencia

Referente a la persistencia cabe destacar que esta característica su principal sustentador es el servidor ya que en este es donde mayor número de datos van a ser almacenados y demás, pero no solo van a estar en el servidor toda la persistencia una pequeña parte de la persistencia la tendrán los nodos como se indicara en un punto más adelante.

#### Servidor:




ESQUEMA BASE DE DATOS 5-1

Se ha decidido usar un sistema gestor de bases de datos de tipo SQL. El gestor que se ha instalado para controlar esta base de datos es MYSQL, con esta base de datos no solo se guardará información, sino que también se almacenará parámetros de configuración para su uso.

El motivo por el cual se ha decidido usar una base de datos de tipo SQL está relacionado con el tipo de infraestructura que tenemos ya que al solo tener un servidor con un esquema centralizado no interesa la característica de base de datos descentralizada también se ha decido por este tipo ya que al ser la más probable conseguir mayor integración que con otro tipo de base de datos.

Como podemos observar disponemos de 4 tablas que se explicara cada uno de sus parámetros a continuación. Todas las tablas tienen un id único ya que esto puede llegar a agilizar las tareas de búsqueda.

### Tabla temperaturas:

| #                        | Nombre | Tipo   | Cotejamiento | Atributos | Nulo | Predeterminado | Comentarios | Extra          |
|--------------------------|--------|--|--------------|-----------|------|----------------|-------------|----------------|
| <input type="checkbox"/> | 1      | Id_habitacion  | int(11)      |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> | 2      | Fecha  | date         |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> | 3      | Valor  | int(11)      |           | No   | Ninguna        |             |                |
| <input type="checkbox"/> | 4      | id  | int(11)      |           | No   | Ninguna        |             | AUTO_INCREMENT |

**TABLA TEMPERATURAS 5-2**

En esta tabla se almacenará las diferentes temperaturas de las distintas estancias de la casa. Para sacar sus datos haremos una AVG(media) de los valores, pero previamente lo que se hará es agrupar por fechas. Este es una tabla que no almacenara ningún tipo de configuración solamente almacenara información sobre un tipo concreto de sensor que en este caso es el de temperatura.

### Tabla sensores:

| Id_habitacion | Id_arduino | Tipo | Id | Detalles | Valor | Actuador | Metodo   |
|---------------|------------|------|----|----------|-------|----------|----------|
| 1             | 1          | 1    | 1  | Ventana  | 1     | 1        | encender |
| 2             | 1          | 0    | 2  | comedor  | 1     | 0        | encender |
| 3             | 3          | 1    | 3  | Luz mesa | 1     | 0        | encender |

**TABLA SENSORES 5-3**

Esta es una de la tabla más importante que podemos encontrar entre las diferentes tablas ya que esta tabla relaciona datos entre las diferentes tablas, como podemos observar en las dos primeras columnas podemos encontrar la relación de las tablas Habitaciones mediante su *id* y la *tabla Arduinos* también por su parámetro *id*. Cabe destacar que el objetivo de esta es el de almacenar los sensores /actuadores de nuestro sistema ya que en el caso que no estén en esta tabla en caso práctico es como si o existieran. Para todo esto necesita también almacenar parámetros de configuración y datos como severa a continuación.

Para conseguir identificar a que arduino pertenece y a que habitación hace referencia usamos *Id\_arduino*, *Id\_habitacion* respectivamente.

Para poder diferenciar entre si es de tipo sensor hacemos uso del campo *Tipo* que en el caso que este colocado un cero hace referencia a que será de tipo sensor sin embargo si en la celda hay colocado un uno hace referencia a un actuador. En el caso que sea un actuador puede ser de dos tipo binario o gradual.



En el caso que sea binario en la celda *Actuador* está colocado un uno, por lo tanto, la acción referente a este actuador solo puede ser encendido o apagado.

Pero en el caso que sea de tipo gradual en la celda *Actuador* estará colocado un cero, por lo tanto, la acción referente a este actuador puede ser gradual.

Para finalizar usamos la columna *Método* para identificar cual es la función que tiene que ejecutar el Arduino internamente ya que será necesaria en el caso que tenga varias funciones.

### Tabla Habitaciones:

| Id | Nombre           |
|----|------------------|
| 1  | Dormitorio       |
| 2  | Comedor          |
| 3  | Cocina           |
| 4  | Dormitorio Niños |
| 5  | salon            |

TABLA HABITACIONES 5-4

En esta tabla se encuentran las diferentes estancias que se han añadido mediante la aplicación, cabe destacar que esto no significa que vayan a salir acciones referentes a estas ya que en el caso que estas estancias no tengan asignado algún arduino más concretamente ningún sensor no saldrán en el menú de acciones o de información, en el único sitio donde se mostrará es en la vista de añadir sensor ya que para que se pueda añadir un sensor se tiene que vincular con un estancia previamente creada.

### Tabla Arduinos

| Id | Ip          | Descripcion           | Id_habitacion |
|----|-------------|-----------------------|---------------|
| 1  | 192.168.1.2 | Dentro del dormitorio | 1             |
| 2  | 192.168.1.3 |                       | 2             |
| 3  | 192.168.1.4 |                       | 3             |

TABLA ARDUINOS 5-5

En esta tabla podemos ver los diferentes Arduinos que hay por la casa y sus direcciones ip. Como podemos ver tenemos el atributo *Id\_habitacion* que nos sirve para saber en que estancia está situado este arduino, referente a la ip como se ha indicado anteriormente está pensado para una red de clase c por lo tanto tendrá un máximo aproximado de 253 de Arduinos en el caso que solo tengamos una aplicación controladora.

Esta tabla también nos ayuda a la hora de asignar IPs a nuevos nodos ya que se basa en la última ip dada para dar la siguiente al nuevo nodo.

## Nodos:

En los Node es necesario que se almacenen de forma estática pero configurable una cierta cantidad de datos, ya que como se ha comentado anteriormente en caso de reinicio es necesario que el nodo por si mismo se capaz de recuperarse por lo tanto lo que el nodo va a almacenar va a ser lo siguientes datos Nombre la de wifi, Contraseña del Wifi y para finalizar Ip que tiene asignada.

Se ha decidido abogar por la siguiente técnica ya que por los detalles de implementación es la que mejor se adecua a los requisitos del proyecto.

En la primera posición guardamos el tamaño de la siguiente palabra, a continuación, vamos poniendo carácter a carácter esa palabra en los sucesivos huecos hasta completarlo, así lo haríamos hasta acabar con todos los datos. Hay que tener en cuenta que puede darse el caso que el nombre o la contraseña tengan más de 9 por lo tanto no podremos indicarlo con un solo número tampoco que su puede obligar a que lea el siguiente dígito y si es número que lo concatene con el anterior ya que en este caso pondríamos la limitación al usuario de que ningún de estos parámetro pueda empezar por un numero por lo tanto lo que se ha decidido hacer es almacenar el número de dígitos que tiene la longitud de los siguientes parámetros y con esta solución se puede almacenar aunque desperdiciemos dos huecos pero no es una desventaja ya que tenemos más de sobra para esto Ejemplo para la configuración 192.168.1.6 ono256 Casa de almacenaría así:

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 1 | 9 | 2 | 3 | 1 | 6 | 8 | 1 | 1 | 1 | 6 | 1 | 6 | O | N | O | 2 | 5 | 6 | 1 | 4 | C | A | S | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Además de los datos de conexión en la parte final del segmento de almacenamiento, se almacena una palabra para comprobar si este nodo ya ha sido configurado o no, ya que así en el caso que no esté configurado este saltara directamente a la escucha.

Se ha decido poner un límite de 60 caracteres ya que la cadena que indica que está configurado esta almacenado al final y además de este motivo también es porque no todos los nodos tienen la misma capacidad de almacenamiento por lo tanto así se asegura de que todos tiene suficiente espacio.

Como ayuda se ha seguido el tutorial Read and writing esp8266 que se indica en la bibliografía para la ayuda del uso de la EEPROM

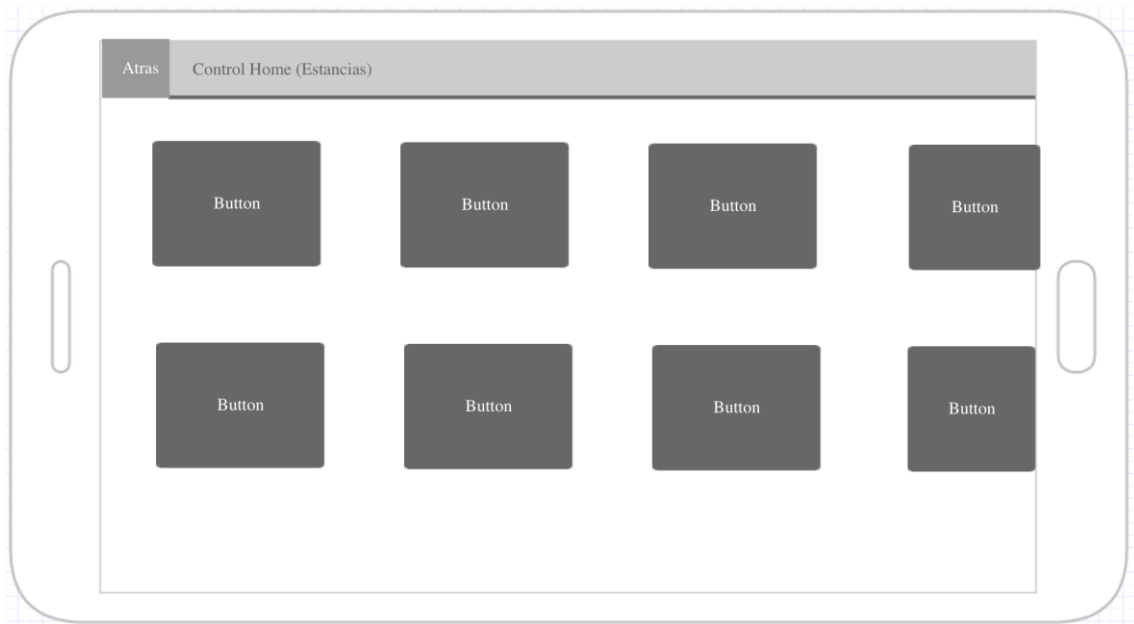
## 5.3 Capa Presentación

Esta capa esta implementada en la App creada para la Tablet ya que mediante esta herramienta el usuario podrá interactuar con la casa y obtener información relevante a esta.

A continuación, se podrá observar algunas de las imágenes que se crearon en esta fase para evaluar el diseño si era el correcto.



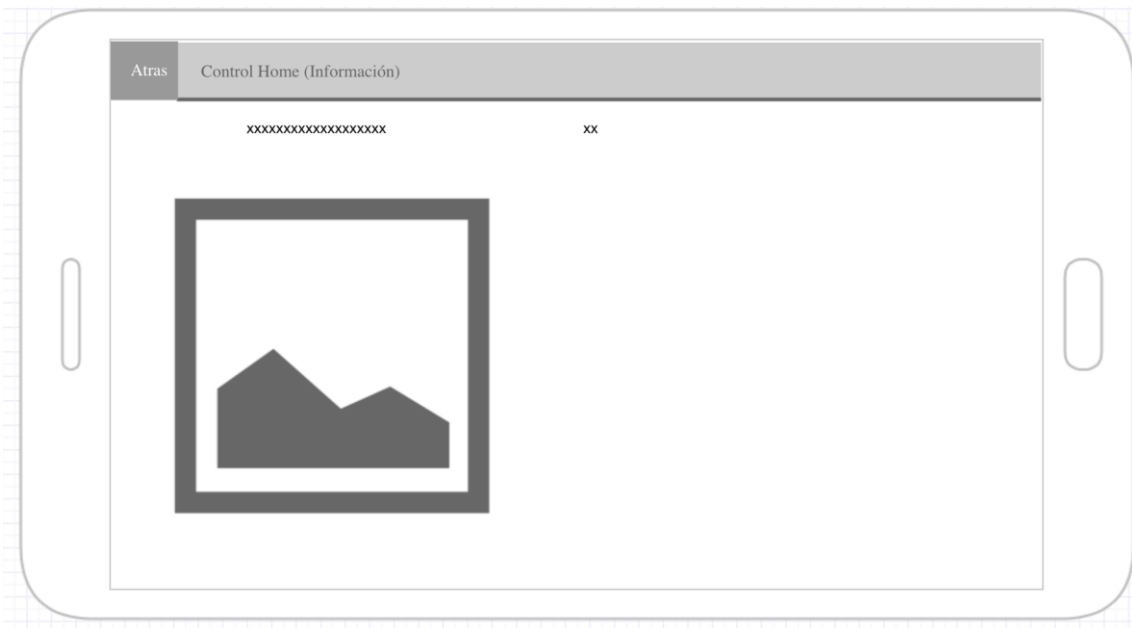
**MENÚ PRINCIPAL 5-6**



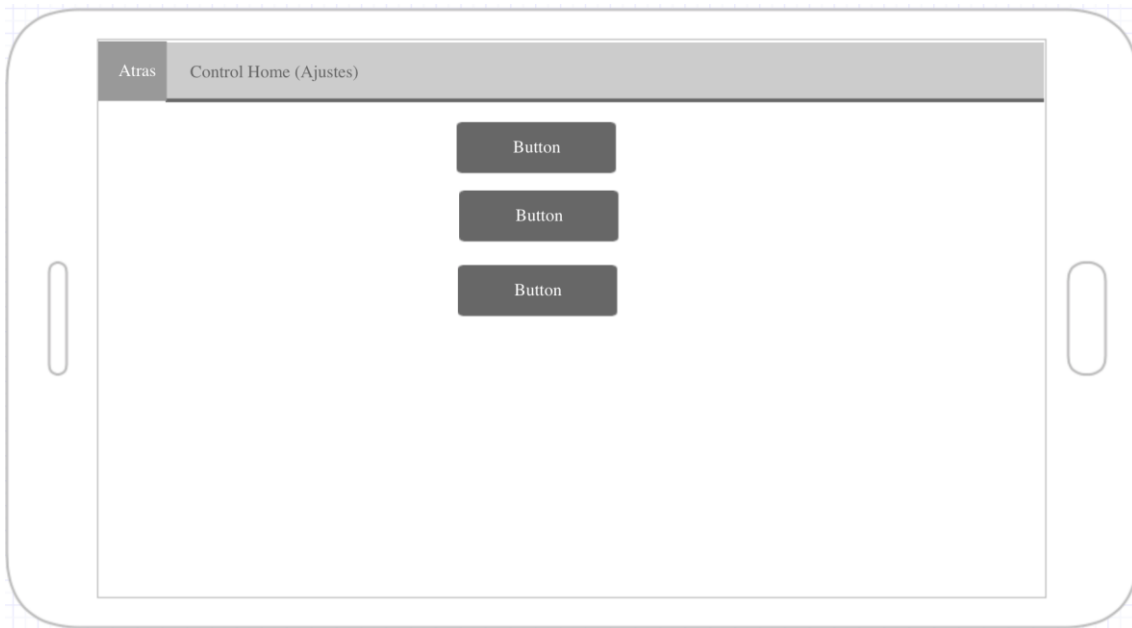
**VISTA ESTANCIAS 5-7**



VISTA ACCIONES 5-8



VISTA INFORMACIÓN 5-9



**VISTA AJUSTES 5-10**



**VISTA AJUSTES WIFI 5-11**

Referente al diseño se ha buscado una interface sencilla para que a la gente que tenga menos habilidades tecnológicas que se manejó peor con dispositivos móviles puedan manejarse con mayor facilidad.

Para conseguir una interface satisfactoria se ha considera que esta tenía que cumplir una serie de requisitos ya que esta parte es importante para el usuario y es la parte visible de todo lo que hay detrás.

A continuación, se exponen los requisitos.

- Que la opción más profunda estuviera a tres pasos desde el menú principal.
- Que el usuario tardara menos de 10 segundos en ejecutar una acción o ver un dato.

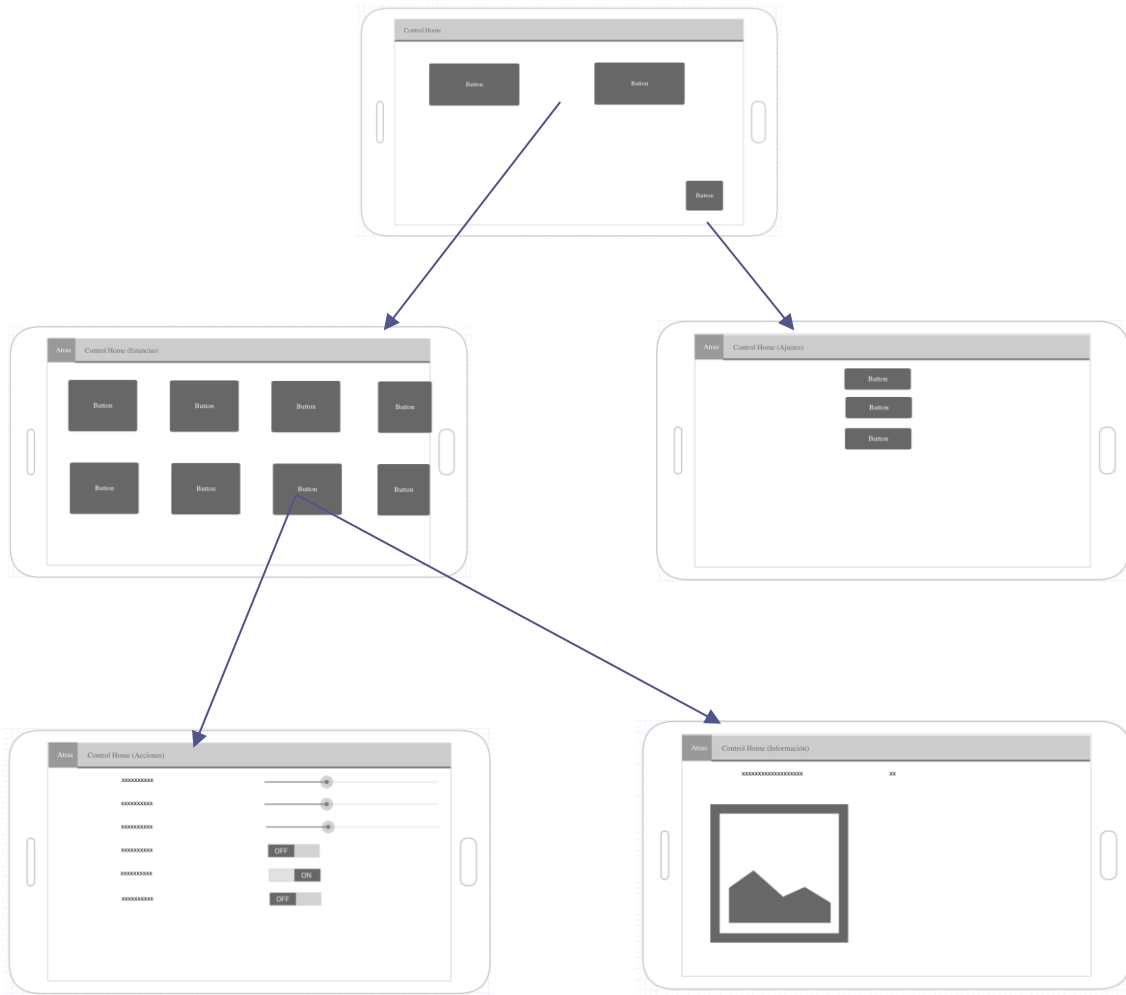


- Que al añadir una nueva acción el tiempo para encontrarla sea de menos de 20 segundos.
- Que siguiera un orden coherente con la información mostrada en cada situación.

En el caso de estudio se ha empleado a diferentes personas de distinta edad y de distintas destrezas tecnológicas, pero de todas formas se ha destacado más el perfil más tecnológico ya que se cree que la persona objetivo de este proyecto puede ser aquella que le gusta la tecnología, ya que al ser una tecnología novedosa no es normal que las personas que tienen poco interés en este tema se dediquen a implantar esta tecnología en sus casas.

Uno de los aspectos que salió en general es que los usuarios se sienten más cómodos cuando la vista cambia lo mínimo posible ya que así cuando se acostumbran y los botones están en el mismo sitio siempre les es más fácil de encontrarlos y por lo tanto navegar más rápido.

## 5.4 Capa Lógica Diagrama de Alcance



Como se puede observar se ha buscado que en los mínimos pasos se llegue al sitio deseado, también se ha intentado reaprovechar la vista ya que así al usuario le será más cómodo de moverse por la interface

Un aspecto que cabe destacar en este diagrama es que la vista central es la misma con información distinta, pero siguiendo el mismo esquema para el botón acciones o para el botón de información.

## 6 Implementación

---

Referente a la implementación se abogó por usar unas tecnologías específicas que se comentarán a continuación. También se indicará las herramientas necesarias tanto para el desarrollo como la posible implantación.

### 6.2 Herramientas utilizadas

Aquí se describirán cada una de las herramientas que han sido utilizadas para el desarrollo del proyecto tanto para el diseño como para la implementación.

#### **Android Studio:**

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Con este entorno de desarrollo se ha creado la aplicación para los dispositivos Android.

#### **SDK**

El SDK de Android es un conjunto de herramientas de desarrollo de software que le permiten al programador crear aplicaciones para un sistema en concreto que en nuestro caso este sdk pertenecerá al kit de Android ya que hemos creado la aplicación para estos dispositivos.

#### **XAMP**

Xamp es la unión de tres servicios (MYSQL, Apache, PHP) por la cual nos permite tener un servidor de páginas web con interpretación de php y un gestor de bases de datos de tipo SQL. Existe diferentes versiones dependiendo cual es el OS del host anfitrión.

#### **Arduino IDE**

Es un entorno de programación multiplataforma pensado para la programación de dispositivos Arduinos. Con este IDE también nos permite transmitir el código para ser ejecutado dentro del hardware.

#### **Mule**

Mule es un framework de mensajería ESB. Es un gestor de objetos escalable y distribuido que puede manejar interacciones con servicios y aplicaciones que usan distintas tecnologías de transporte y mensajería.

#### **GLIFFY**

Gliffy es un programa web que nos permite desde crear mockups hasta crear diagramas SQL aunque es cierto que necesita registro para poder exportarlo, es gratuito la edición web para crear los diagramas.

## 6.3 Lenguajes utilizados

Aquí se describirán los diferentes lenguajes de programación que se han usado en el desarrollo del proyecto. Cabe destacar que se podrían usar otros lenguajes ya que en el caso que queramos crear la aplicación de control para dispositivos iOS necesitarías el lenguaje Swift.

### **Java-Android**

Este lenguaje es necesario para la creación de la app, aunque ciertamente Android está basado en Java este añade funcionalidades nuevas que hay saber cómo implementarlas.

### **SQL**

Es un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos.

### **Servicios WEB (HTML, PHP)**

Aunque ciertamente usemos Mule para todas estas tareas es importante tener conocimientos de los lenguajes necesarios para la creación de estos servicios, ya que sabiendo cómo funcionan podemos llegar a completar las funciones deseadas.

### **Arduino**

Este lenguaje está basado en C++.

C++ es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

## 6.4 Tecnologías usadas

Se describirá las tecnologías que se ha usado en diferentes fases del proyecto

### **REST**

Es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. Se suele usar para describir cualquier interfaz entre sistemas que utilice directamente HTTP para obtención de datos. Los sistemas que siguen los principios REST se llaman RESTfull.

Algunas de las características de lo definen son, por ejemplo:

Protocolo clientes/servidor sin estados

Un conjunto de operaciones definidas: POST, GET, PUT Y DELETE.

Sintaxis universal para identificar los recursos.



## Json

Es el acrónimo de *JavaScript object Notation*, es un formato de texto ligero para el intercambio de datos.

Una de la característica que tiene es que facilita la estructuración de lo datos para ser interpretados por el receptor.

Por lo tanto, dependiendo de la estructura se podrá mandar listas y objetos independientemente.

## 6.5 Arquitectura

Como se verá a continuación se ha tomado diferentes decisiones referentes a las tecnologías e infraestructuras a la hora de la implementación. Aquí podemos ver un resumen que está más detallado en cada apartado específico.

**Comunicación:** Se ha utilizado la tecnología REST y JSON.

**Persistencia:** Por lado del servidor se ha decidido instalar un gestor MYSQL y por lado del nodo se utiliza la EPROM con una codificación especial.

**Programa de control:** Se ha utilizado el framework mule.

**Nodos:** Para los nodos se ha decidido usar el módulo nodemcu 12-e

**Servidor:** Para el servidor en este caso se utilizará una Raspberry pi-3 pero como se comenta en el punto correspondiente hay más libertad a la hora de elegir este dispositivo

## 6.6 Comunicación

La comunicación entre los diferentes servicios y dispositivos va a ser de tipo REST ya que así podremos conseguir independizarnos de la tecnología por la cual este implementada los diferentes elementos siempre y cuando cumpla con los estándares de HTTP que al ser un estándar toda tecnología capaz de conectarse a internet debería de cumplir.

Para la estructura de los mensajes que se envían será en formato JSON, así de esta forma muchos lenguajes directamente interpretan este formato y nos ayuda a sacar la información deseada.

Se ha decidido usar como canal de comunicación la tecnología Wifi, los motivos por los cuales se ha decidido usar esta tecnología son varios algunos de ellos son que es una tecnología que ya está implantada en la mayoría de los hogares. Otros de los motivos es que realmente es cómodo y barato crear una nueva red Wifi o ampliar el alcance de esta para que llegue a todos los dispositivos.

Un inconveniente a la hora de usar esta tecnología es que si se coloca muy lejos el nodo o algún elemento de la fuente del wifi esta puede sufrir perdida de señal esporádicas y por lo tanto dejar de darnos la funcionalidad o crear diferentes problemas.

La tecnología de comunicación que se ha usado a nivel físico es Wifi ya que con esta tecnología conseguimos gran movilidad de los dispositivos y facilidad de instalación ya

que no es necesario colocar ningún cable de alimentación. La distancia que abarca el wifi es fácilmente ampliable y se puede realizar con conocimientos básicos.

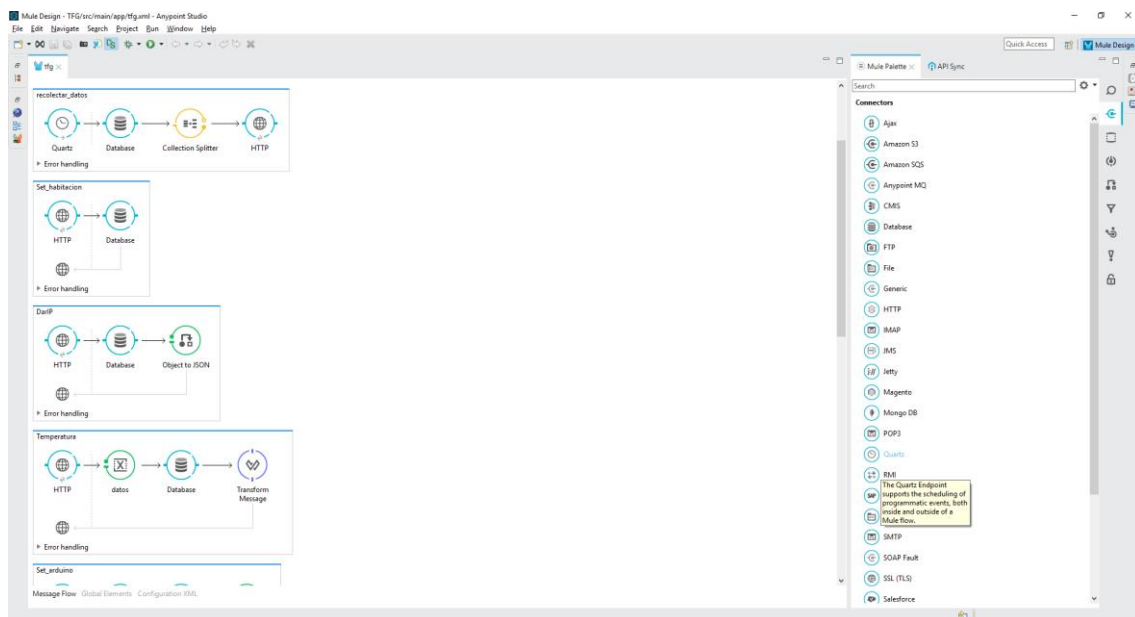
Una de las características del wifi nos ayuda es implementar una mínima seguridad, aunque en el desarrollo del proyecto no se ha contemplado este aspecto al usar esta tecnología indirectamente conseguimos algo de seguridad en la comunicación.

## 6.7 Programa de control

Para el programa de control que se desplegará en el servidor se ha usado el framework Mule ya que este tipo de tecnología está diseñada para propósitos como estos otra alternativa era crear manualmente los diferentes servicios web, pero a la hora de automatizar tareas es un aspecto bueno que tiene este framework.

### 6.7.1 Mule

Mule es un framework de mensajería ESB (Enterprise Service Bus). Es un gestor de objetos escalable y distribuido que puede manejar interacciones con servicios y aplicaciones que usan distintas tecnologías de transporte y mensajería.



VISTA MULE 6-1

Con mule se ha creado dos roles principales por un lado tenemos el de generar servicios web para el usuario y por el otro lado consumimos diferentes servicios este aspecto se usa para automatizar tareas o bien para reaprovechar servicios vinculados entre si.

### Generar servicios web

Cuando proporcionamos un servicio web lo que generamos es un inicializador vía web, por el cual recibimos los datos que se necesitaran para su procesamiento o funciones para ser ejecutadas junto a datos, para todos estos datos se ha seguido la filosofía de servicios REST por lo tanto los datos van embebidos dentro de la URL.

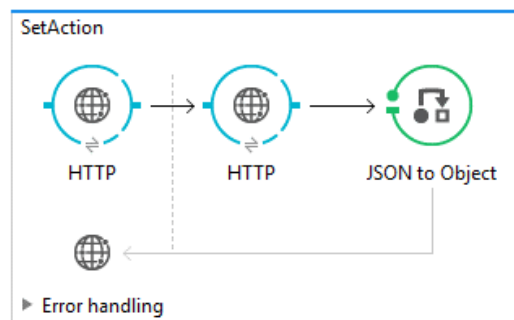


Ejemplo:

```
"/Set_arduino/{Ip}/{Id}"
```

El primer parámetro es la función que llamar y el resto de parámetros son los datos que se necesitan.

La cuando solicitamos alguna información lo que realiza es una consulta SQL a la base de datos que está instalada en el mismo dispositivo y con los datos de la consulta genera un JSON para ser devuelto a la aplicación. Pero cuando queremos modificar el medio físico por lo tanto que un *Arduino* ejecute una tarea llamamos a una función diferente. Esta función lo que realmente hace es una pasarela entre la app y el Arduino deseado.



### ACCIONES 6-2

Como se ha comentado anteriormente el sistema brinda la posibilidad de comunicación con el exterior por lo tanto lo que se ha hecho es aprovechar esta característica / restricciones para crear un método genérico que independientemente la petición venga desde el exterior como del interior del hogar pasara por el mismo hilo de ejecución.

Por lo tanto, lo que hace esta función es trasladar la petición desde la entrada del servidor hacia la entrada del nodo correspondiente por lo tanto lo que se hace es enviarle los siguientes datos:

```
"/Set_Action/{ip}/{method}/{datos}"
```

Con estos datos lo que se hace es generar una petición nueva hacia el nodo correspondiente que es te indicado en el parámetro ip y este ejecutara la acción que está indicado en el campo método adicionalmente se le manda datos que puede que use o puede que no, uno de los aspectos de esta tecnología es que no permite sobre carga de operadores por lo tanto le tenemos que tener que mandar todos los datos especificados, aunque no se usen en todos los casos.

Los datos que nos devuelve el *Arduino* son codificados en JSON\* y estos son devueltos a la App iniciadora.

\*Realmente no llegar a ser un *JSON* ya que las comillas dobles (") generan error por lo tanto lo que hay que hacer es genéralo con simples y posteriormente reemplazarlas.

### Consumir servicios web

Como se ha comentado en la introducción de este apartado a la hora de consumir servicios web teníamos dos motivos uno era reutilizar funciones vinculadas que en nuestro caso de ejemplo no se dan y el otro es el de automatización de tareas que si se realiza. Por lo tanto, lo que se ha hecho es que el inicializar de una automatización es un



reloj programado que se inicie cada cierto tiempo y a continuación hará las tareas que se le haya programado que haga.

## 6.8 Nodos

Para el rol de los nodos se ha decidido usar Arduino ya que al ser una plataforma bajo código libre que nos brinda mayor libertad a la hora de alterar su configuración.

Para ser más concretos se ha decidido usar el módulo NODEMCU 12e. ya que las características de este módulo son que además de incorporar las funcionalidades de un arduino también integra la electrónica necesaria para la conexión wifi.

Ya que todo está interconectado por wifi es un factor en tener en cuenta para descartarse por usar estos módulos. También cabe destacar que tienen un tamaño más pequeño que por ejemplo un Arduino uno o un Arduino mega.



**NODEMCU 6-3**

Un componente que necesita además del wifi es una conexión bluetooth, ya que la tarea de sincronización inicial por la cual el servidor le hace llegar mediante la aplicación los datos necesarios para que este dispositivo se conecte al wifi de la casa, el método implementado para guardar los datos en la EEPROM está indicados en esta sección en el apartado de persistencia y en el sub apartado nodo.

Todos los diferentes nodos tendrán un código base para así disponer de métodos genérico independientemente de las funciones que vayan a desarrollar.

Nada más conectarlo el dispositivo mira a ver si ya está configurado en caso afirmativo intentara conectarse a la red wifi que este almacenada dentro de él, pero en todo momento este dispositivo estará escuchando por si le entra algún dato por el puerto serie ya que por este medio de comunicación es por donde se configurará.

En el caso que no esté configurado directamente pasara al modo escucha del puerto serie para así poder recibir la configuración deseada.

En el momento que este se conecta a la red wifi lanza un servidor web para poder escuchar las peticiones web que le hagan. Un aspecto que destacar es que todo el nodo en primera estancia tiene un rol pasivo, pero en el caso que se necesitara podrían poder a pasar activos ya que así podríamos ampliar sus funcionalidades.

## 6.9 Aplicación Android

En este caso lo que se ha usado para que el usuario puede comunicarse con la casa es una aplicación creada en lenguaje Android para una Tablet.

Desde esta aplicación además de poder controlar la casa podremos añadir más funcionalidades a esta ya que el asistente de añadir nuevos elementos está integrado en a la aplicación.

En un futuro se podría crear aplicaciones para diferentes plataformas y así poder abarcar los déficits que tienen otras aplicaciones similares.

Esta aplicación para su correcto funcionamiento necesita una conexión permanente con internet. Ya que todo lo referente a información, sólo almacena dentro de ella la dirección ip del servidor, el resto de información la consigue consumiendo los servicios web del servidor.

El contenido que muestra en las vistas es dinámico por lo tanto lo que se ha creado diferentes inflates para poder crear la vistas dinámicamente ya que dependiendo de la información que nos suministre el servidor podremos ver diferente tipo de información, pero un aspecto para tener en cuenta es que tiene que mantener en la medida de lo posible la misma distribución

Cuando el usuario lanza la aplicación esta genera una conexión con el servidor que dependiendo las acciones ejecutadas le mandara una serie distintas de datos, pero siempre con la misma base, por ejemplo:

Encender una luz

```
Ip_servidor+ "/Set_Action/" +ip_cliente+ "/" +metodo+ "/" +valor
```

Por lo tanto, al ser un servicio REST se traducirá de la siguiente manera

Base/Método de entrada/Ip del Arduino que queremos mandar la acción/método/valor, este campo puede ser binario o número entre 0 y 100

En el proceso de sincronización la aplicación Android mediante bluetooth. Lo primero que hará es comprobar si el bluetooth está activo y disponible en caso afirmativo continuará, pero en el caso que no este activo este le pedirá al usuario que lo active, le mandará los datos de acceso al nodo, acto seguido el nodo se reiniciará y cargará la nueva configuración para conectarse a la red wifi. Una vez la aplicación establezca conexión mediante red al nodo, la aplicación le mostrara un asistente al usuario para su configuración. Lo datos recolectados mediante el asistente serán volcados en la base de datos para su posterior uso.

## 6.10 Servidor

A la hora de decantarnos por el servidor no tenemos muchos requisitos ya que lo único que necesitamos es algún sistema con una pequeña capacidad de cálculo y que tenga una conexión de red tanto inalámbrica con alámbrica.

Eso referente a requisitos hardware, por otra parte, referente a requisitos software solo necesitamos que se pueda desplegar mule y que puede instalar un sistema MYSQL.

Por lo tanto, se ha decidido usar una *Raspberry pi 3* ya que cumple con los diferentes requisitos que hemos indicado y su pequeño tamaño y bajo consumo eléctrico es uno de los mejores candidatos.

Sobre la *Raspberry pi 3* se ha instalado un OS de tipo Linux más concretamente raspbian, una vez teniendo el sistema operativo utilizaremos un lamp para disponer de los diferentes servicios que necesitamos para ejecutar el entorno además de lo necesario para levantar el proyecto Mule

## 6.11 Librerías usadas

En esta sección se hará un repaso de las tecnologías usadas para poder llevar a cabo este proyecto

En primera estancia vamos a comentar de la librería que hemos añadido a Android para poder añadir más funcionalidades. Por un lado, tenemos volley y por otro lado tenemos MPAndroidChart.

**Volley** es una librería que nos permite agilizar las comunicaciones en red, ya que todo el tema de gestión de la conexión y la concurrencia lo está tratado por esta librería. En el proyecto es un punto muy importante ya que en el proyecto gran la parte de conectividad tiene mucho peso, es importante que este tema este bien tratado ya que un error en esta parte puede generar un grave problema

**MPAndroidChart** En una librería que se utiliza para generar gráficos en este proyecto no es factor muy destacable simplemente se utiliza para que al usuario le sea más fácil ver algunos datos mostrados y ase hacerle más vistosa la interface

A continuación, se comentará algunos aspectos de los nodos.

Como se ha comentado en puntos anteriores nuestro módulo *Arduino* tenía incorporado una conectividad wifi. Está conectividad es más concretamente un módulo ESP8266, al usar este módulo en el propio código de *Arduino* tenemos que importar su librería para poder utilizarlo y así hacer pleno uso de sus características.

También se ha usado la librería EEPROM para poder hacer uso de esta funcionalidad ya que como se ha comentado la EEPROM juega un aspecto importante y destacable n los módulos

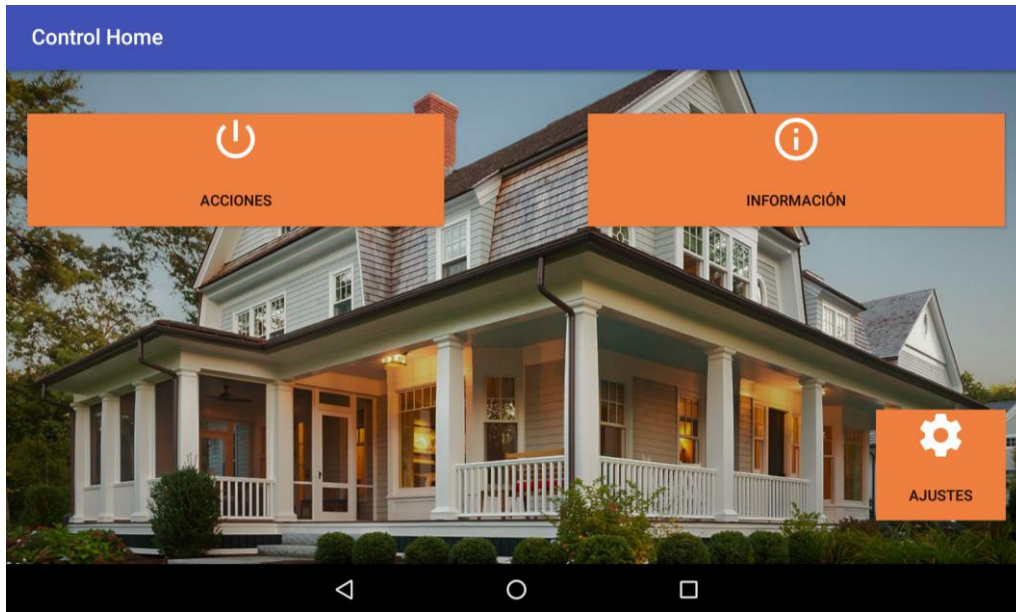
También hay que disponer de la librería para el sensor de temperatura de humedad que se usa para la recolección de datos en esta librería, lo que se quiere hacer referencia es que dependiendo las funcionalidades que dispongo el arduino necesitara diferentes librerías para su uso, pero las únicas que tiene que tener todos es la de EEPROM y la de ESP8266.

## 7 Pruebas

En esta sección se va a mostrar algunos de los aspectos desarrollados en la aplicación indicando en cada momento lo que se está mostrando y por qué hay que realizar algunos pasos.

### Configuración básica

En este punto se va a proceder a la configuración básica del programa.



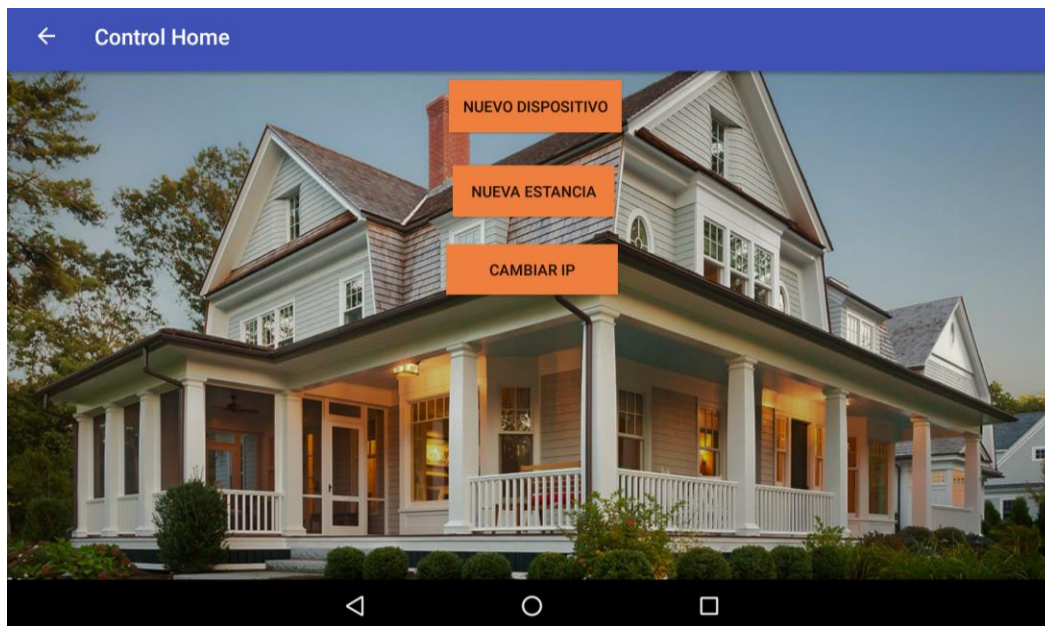
MENÚ PRINCIPAL 7-1

Lo que podemos observar en la imagen es el menú principal que está compuesto de tres botones:

El primer botón que tenemos es el botón de acciones que lo llevará a la vista siguiente (estancias) rellenas solamente con aquellas estancias en las cuales tengamos acciones que realizar.

El segundo botón que tenemos es el botón de información este también llevará a la vista de estancia, pero a diferencia del botón acciones esta vista solo estarán las estancias las cuales tengan información.

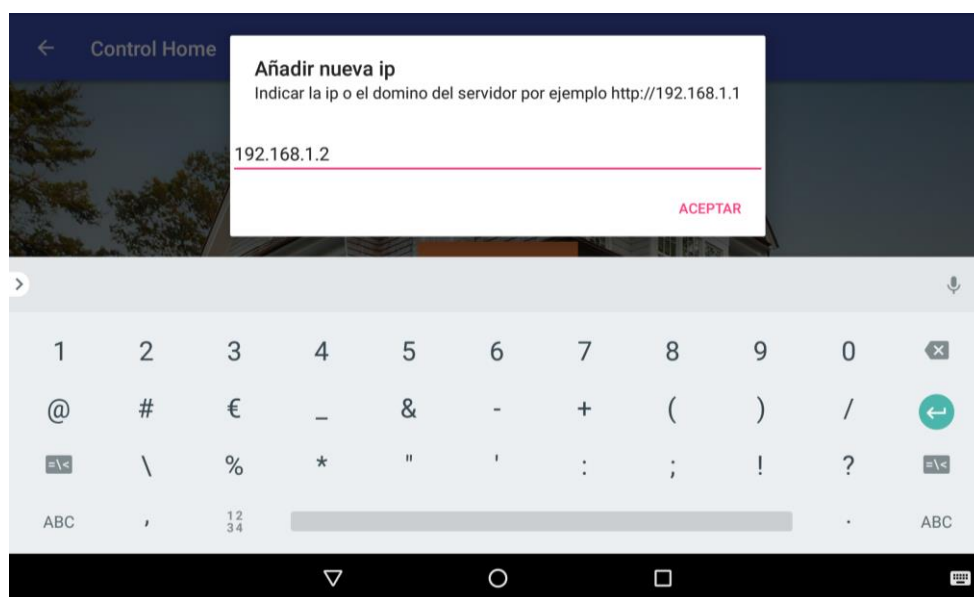
Para finalizar tenemos el botón de ajustes este botón nos llevará al menú de ajustes.



### MENÚ AJUSTES 7-2

Como podemos ver esta vista está compuesta por tres botones en la parte central y uno más en la barra en la parte izquierda que nos llevara a la vista anterior.

Cada una de las opciones que tenemos se irán explicando a mediad que se hagan uso de estas, Para este ejemplo se seleccionara el botón de cambiar ip, Por lo tanto, seleccionáramos el botón de “CAMBIAR IP”.

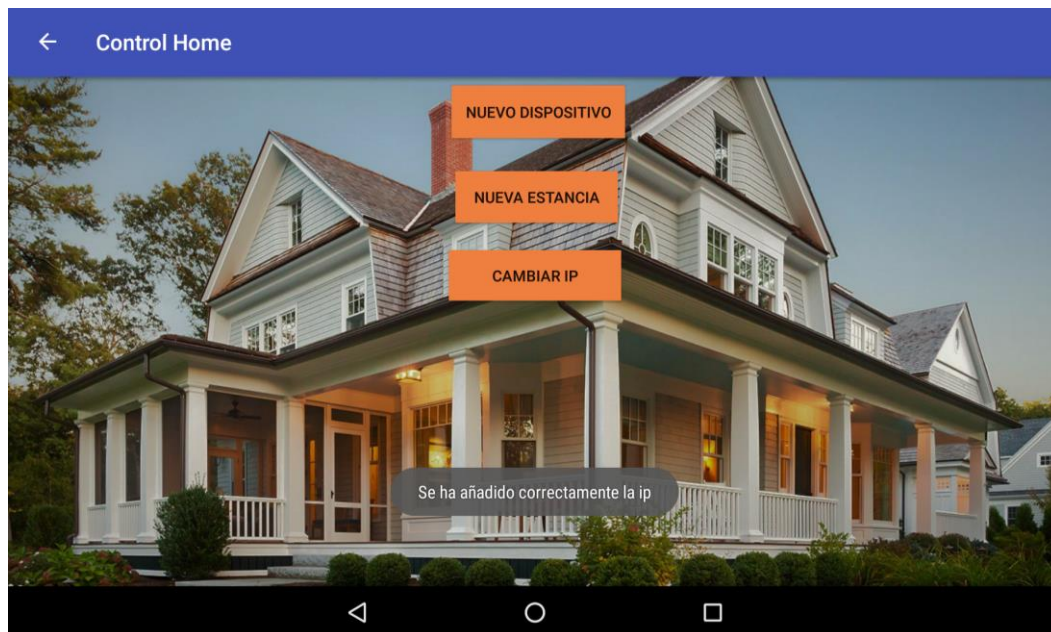


### CAMBIAR IP 7-3

Cabe destacar que el sistema admitiría también url ya que para el control desde el exterior es importante ya que así podemos eliminar el problema de Ip dinámicas

Por lo tanto, indicaremos la Ip la cual está escuchando el servidor de nuestra casa. En el caso que queramos que se puede manejar desde el exterior de la casa es necesario que indiquemos la ip publica o sería mejor hacer uso de un sistema de DNS e indicar la url ya que la ip publica tiende a ser dinámica por lo tanto no sabremos en que momento ha sido cambia y es más tampoco se sabrá a cuál ha sido cambiada.





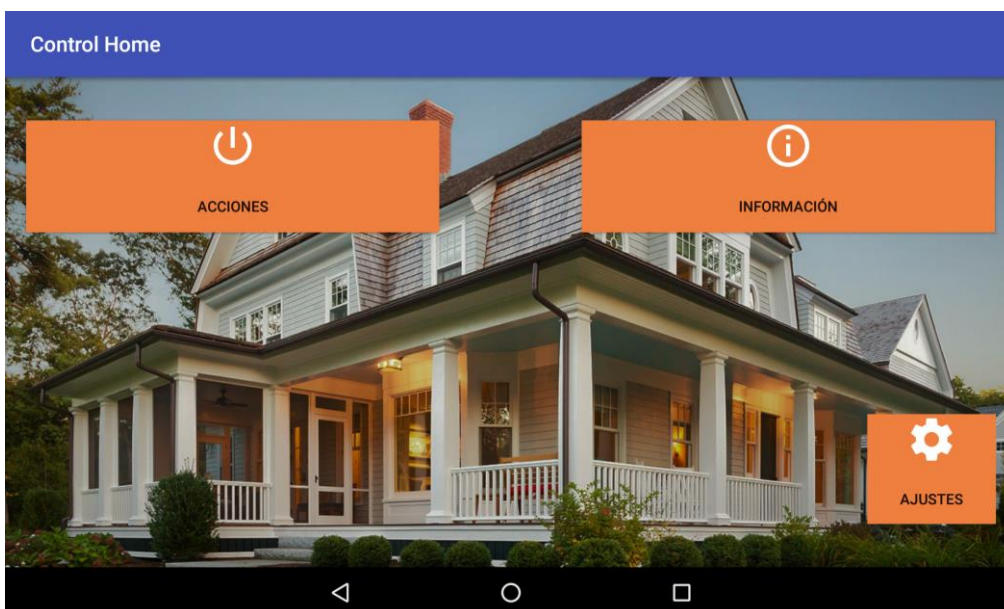
#### MENSAJE DE CONFIRMACIÓN 7-4

Como se puede observar en la imagen en el momento que la ip es cambia se muestra un mensaje indicando que se ha cambiado correctamente y posterior mente saldrá un mensaje indicando cual es la nueva ip que tenemos.

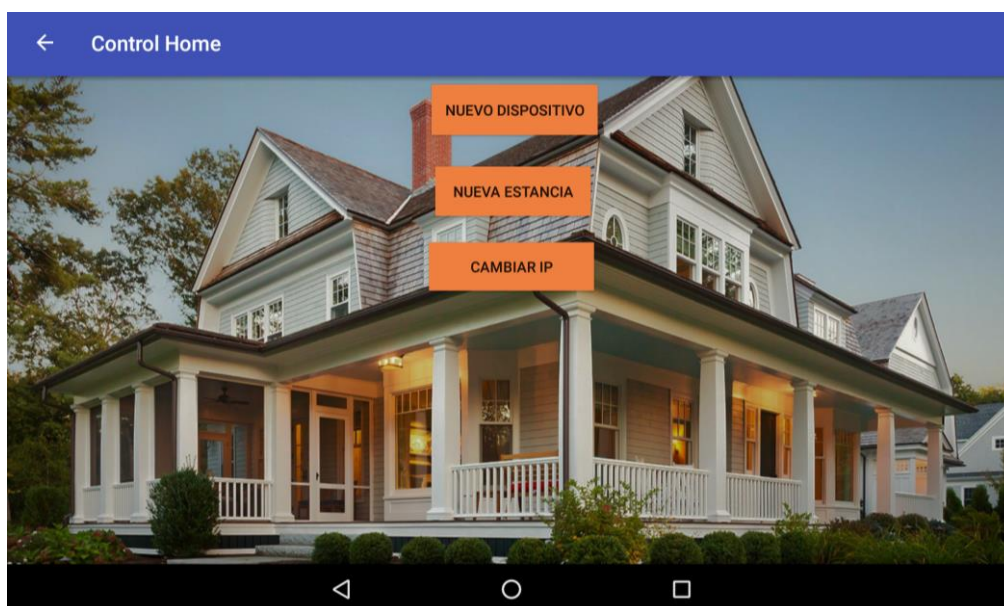
En todos los casos que se verán a continuación contarán con la configuración básica realizada excepto en alguna ocasión que se indicara en el propio ejemplo. Siempre partiremos desde el menú principal con la configuración básica realizada.

### Añadir estancia

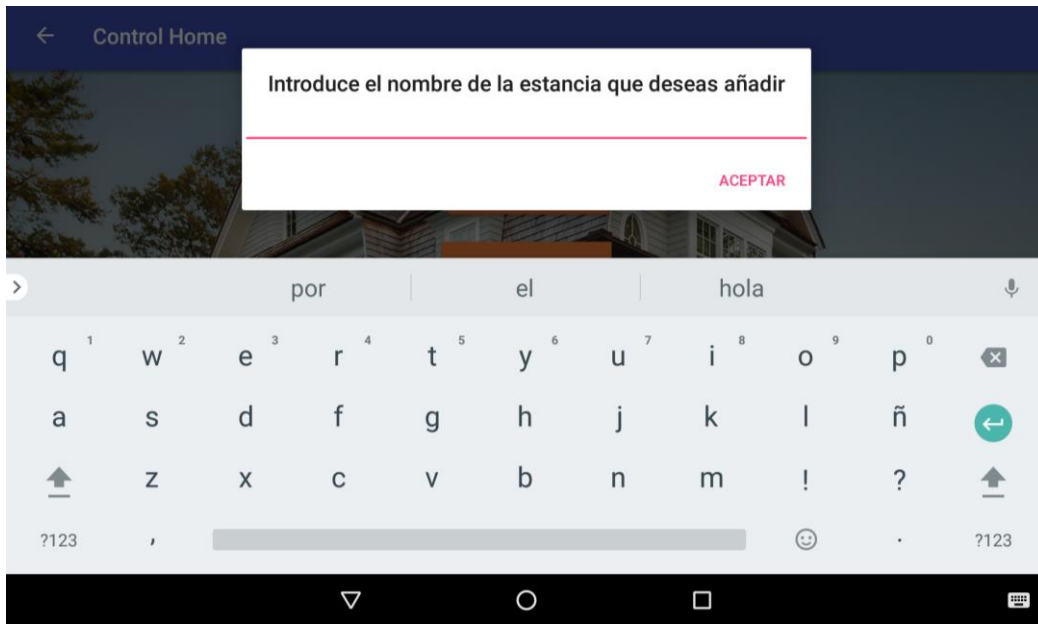
Como se podrá observar a continuación se añadirá una estancia al esquema cabe destacar que hasta que no coloquemos algún sensor o actuador en esta no saldrá para ser seleccionada en las vistas de estancias.



Se ira a Ajustes colocado en la parte inferior derecha



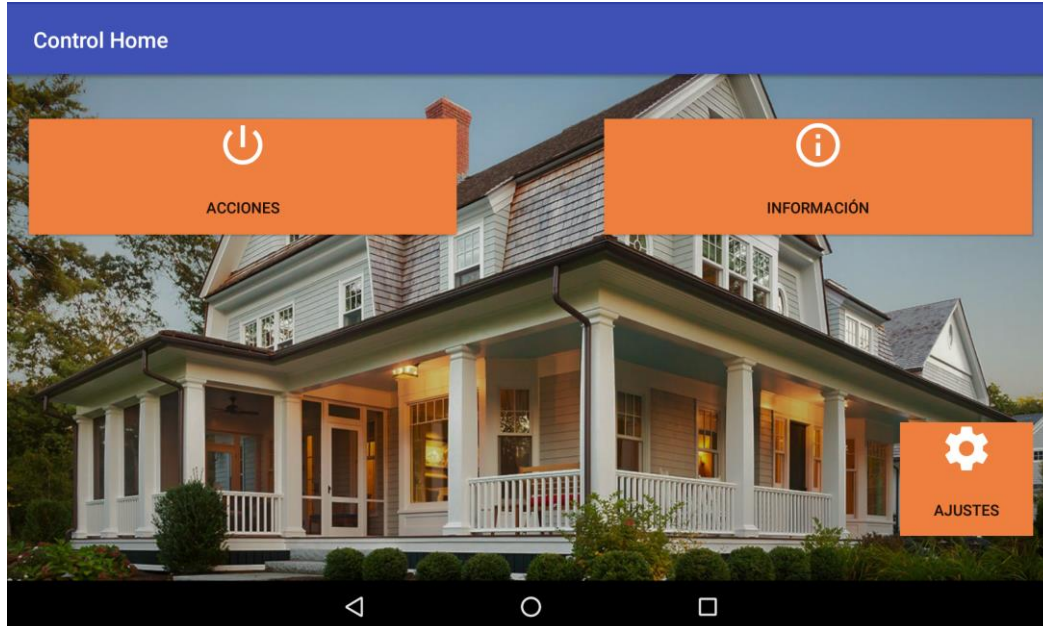
En este caso se seleccionará el botón de nueva estancia la cual nos mostrar la vista que se verá en la siguiente imagen y nos permitirá añadir el nombre de la nueva estancia



### AÑADIR ESTANCIA 7-5

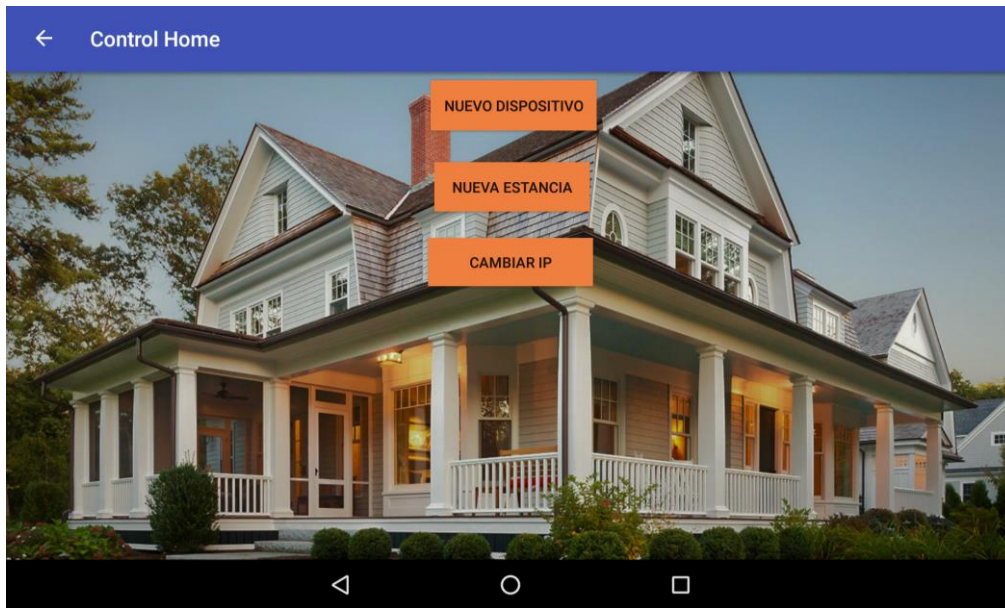
En este formulario se indicará el nombre de la nueva estancia cabe recordar que esta estancia solo será visible en el asistente de añadir nuevo nodo por lo tanto para añadir un nuevo nodo es necesario primero crear la estancia. En las opciones de acciones o de información esta estancia no existirá hasta que se le haya asignado algún sensor como se indicó anteriormente.

### Añadir nuevo dispositivo



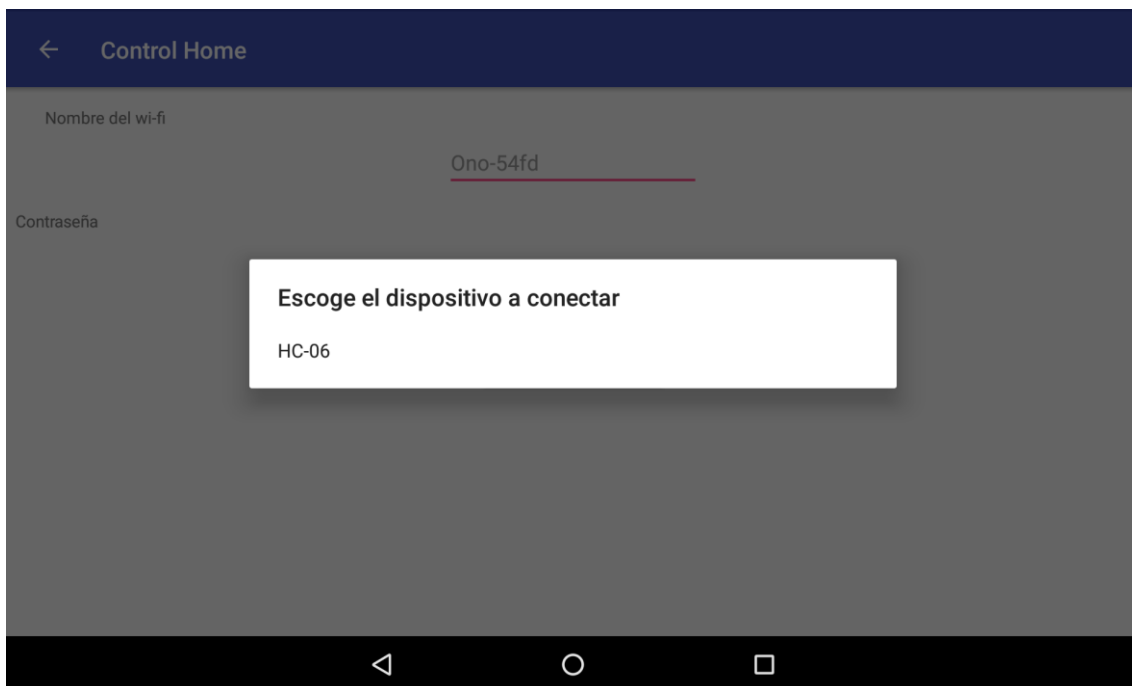
### Menú principal





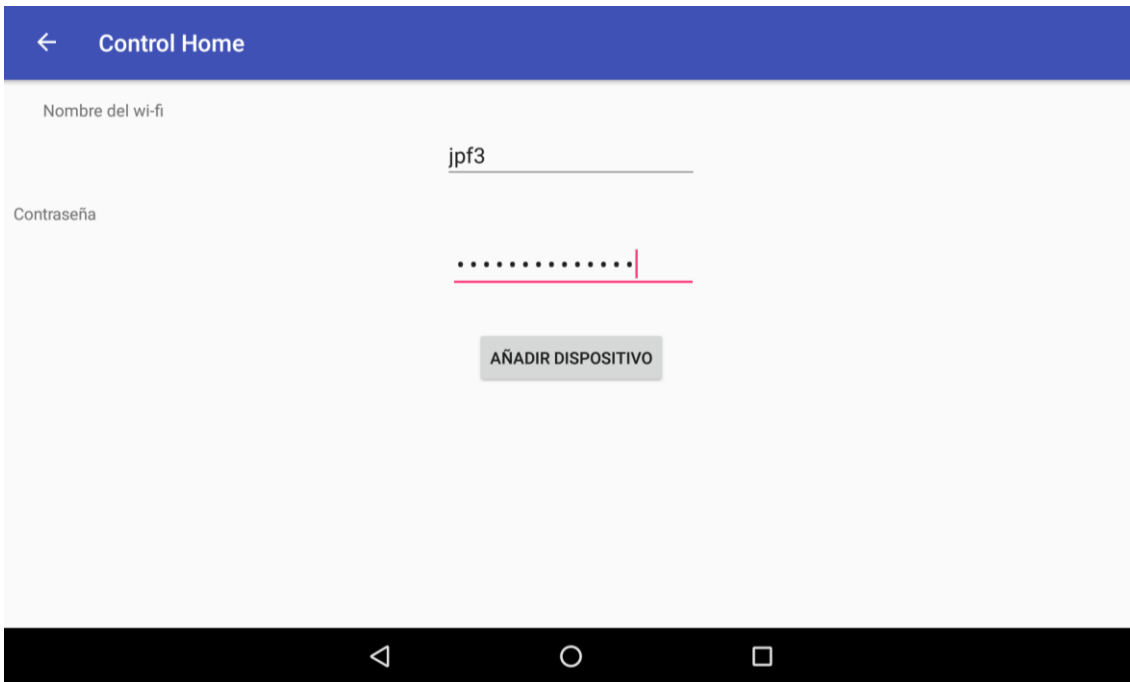
### Menú ajustes

En este caso seleccionaremos la opción de Nuevo dispositivo el cual comenzara con el asistente.



### SELECCIONAR BLUETOOTH 7-6

Antes de iniciar este paso es importante realizar un vinculación del nodo con la tablet ua que en esta vista se mostrar solo los dispositivos que esten vinculados con la tablet tendremos que seleccionar al correspondiente que en nuestro caso es el HC-06.



Control Home

Nombre del wi-fi

jpf3

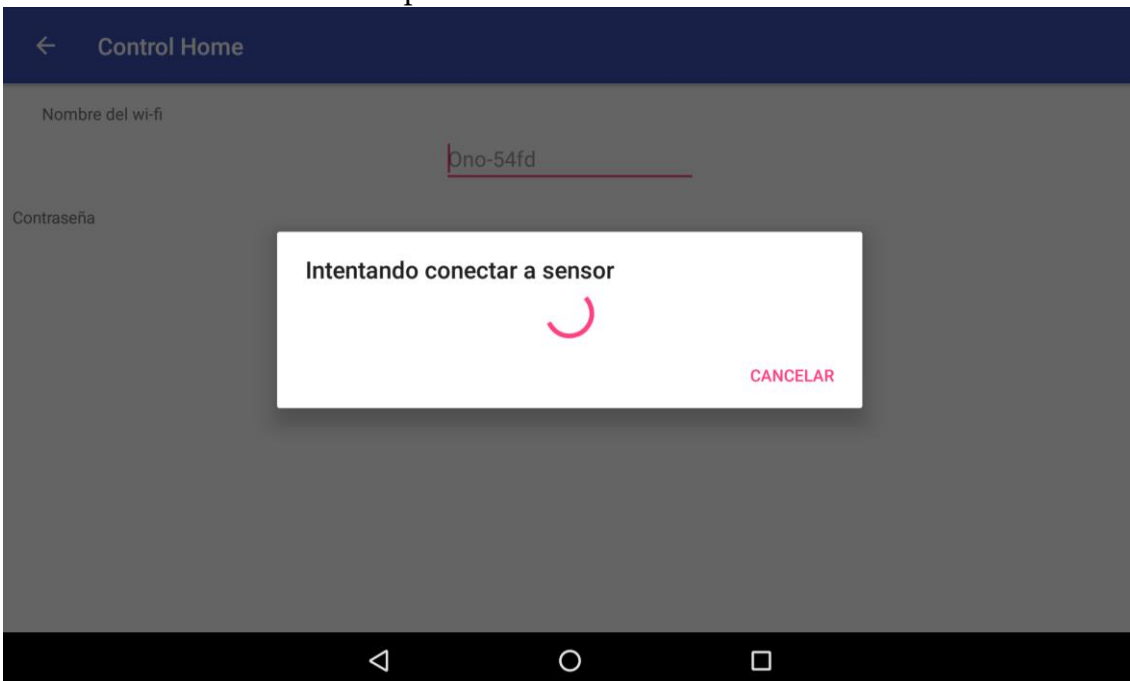
Contraseña

.....

AÑADIR DISPOSITIVO

#### FORMULARIO WIFI 7-7

En esta vista como podemos observar contienen un formulario ,por el cual es usuario indicara cual es el nombre del wifi (SSID) y la contraseña de este. A continuacion le daremos al boton de añadir dispositivo.



Control Home

Nombre del wi-fi

lno-54fd

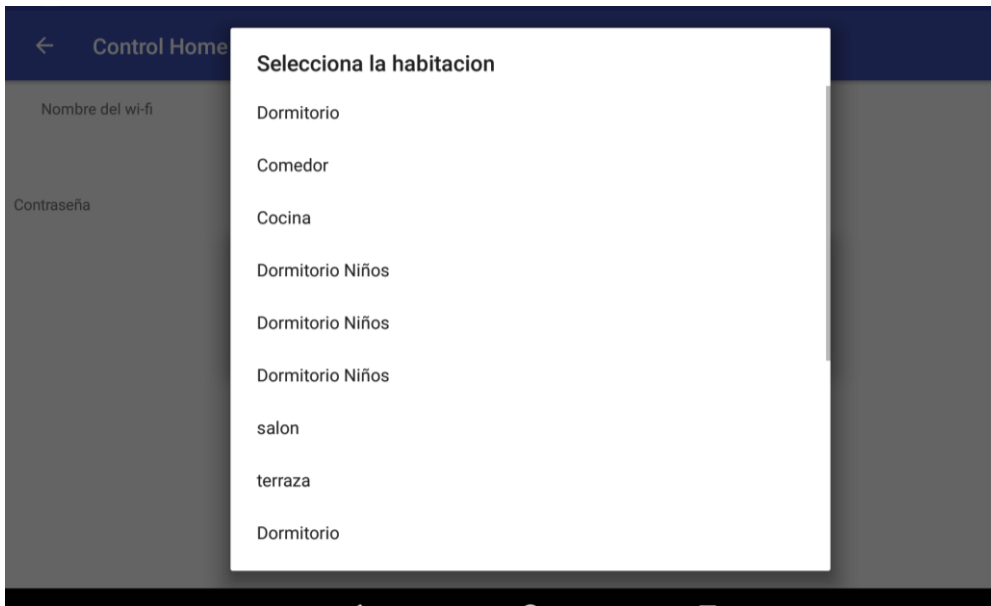
Contraseña

Intentando conectar a sensor

CANCELAR

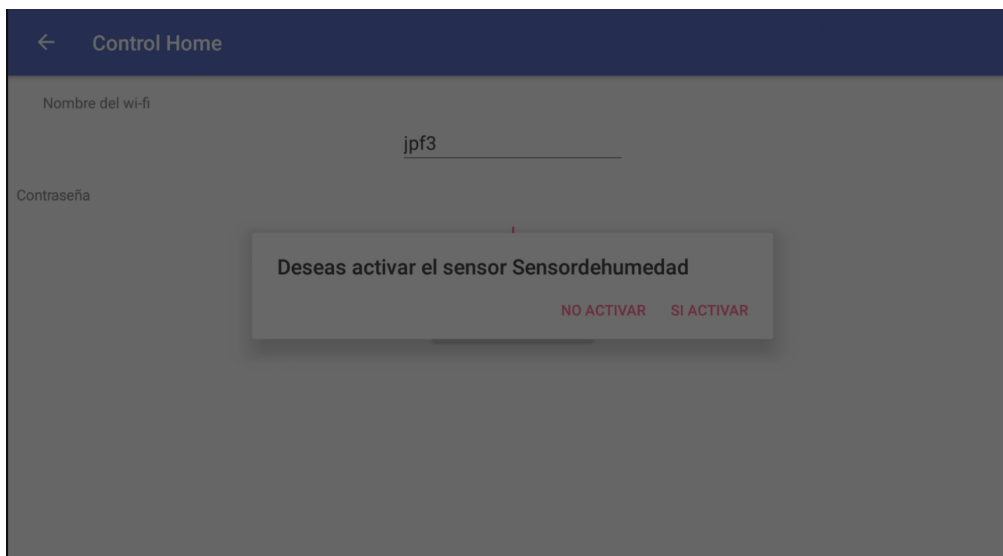
#### BANNER DE CARGA 7-8

La aplicacion se quedara a la espera de conexión hacia el nodo , esta conexión se realizara mediante internet a si que a partir de este punto el bluetooth no es necesario .Nada mas el programa tenga contacto con el nodo este le pedira mediante el servidor el directorio base que es el cual el nodo indica a la aplicación que funcionalidades tienes para evitar posibles bucles el usuario podra cancelar este proceso en el momento que el desee.



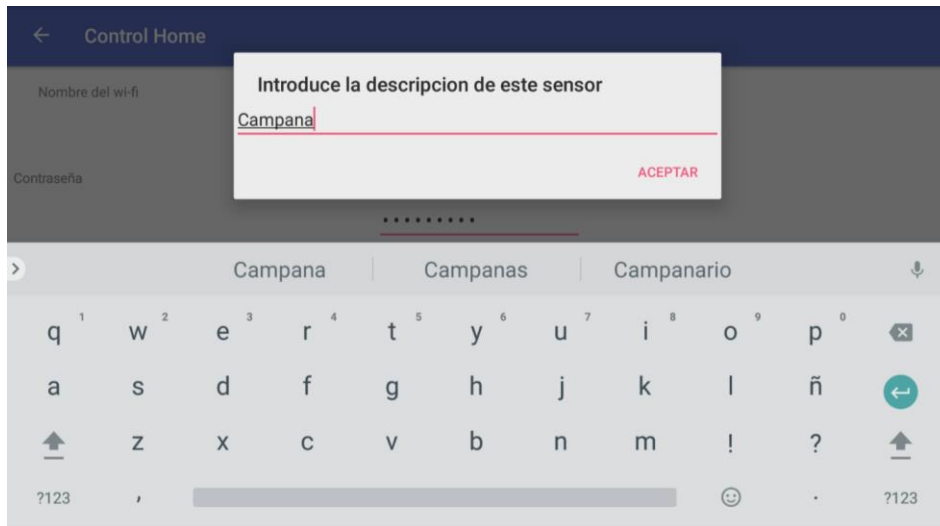
### SELECCIÓN DE ESTANCIA 7-9

Una vez la aplicación tiene la información base del nodo lo que hará será mostrar al usuario la lista de estancias que hay en la casa, es esta lista si que podemos encontrar estancias “vacías” de sensores. El usuario tendrá que seleccionar la estancia donde quiera instalar el sensor.



### CONFIRMACIÓN 7-10

Es esta vista lo que se le mostrará al usuario es si desea activar los diferentes sensores o actuadores que dispone el nodo, el usuario indicará uno a uno si lo desea activar o no. En caso negativo no habrá constancia de que este servicio lo proporcione el Arduino y la única manera de activarlo sería una reinstalación.

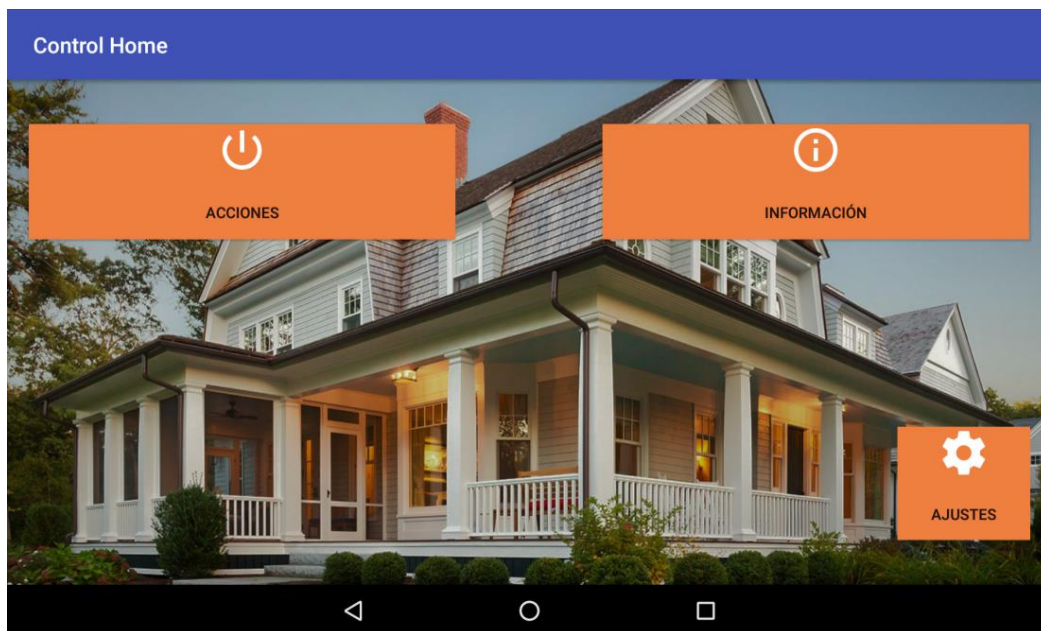


### ALIAS SENSOR 7-11

Es este formulario lo que se le pide al usuario es que le indique el nombre por el cual el lo identificara a la hora de ser mostrado.

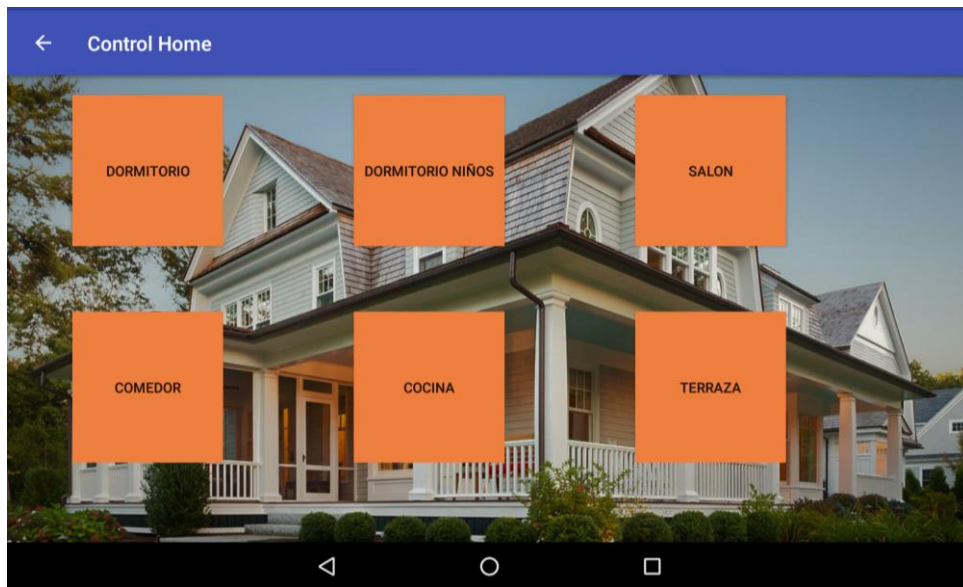
### Encender luz / Ejecutar acción.

Como se ve a continuación se mostrará los pasos para que el sistema reciba la orden de encender una luz y esta se encienda



### Menú Principal

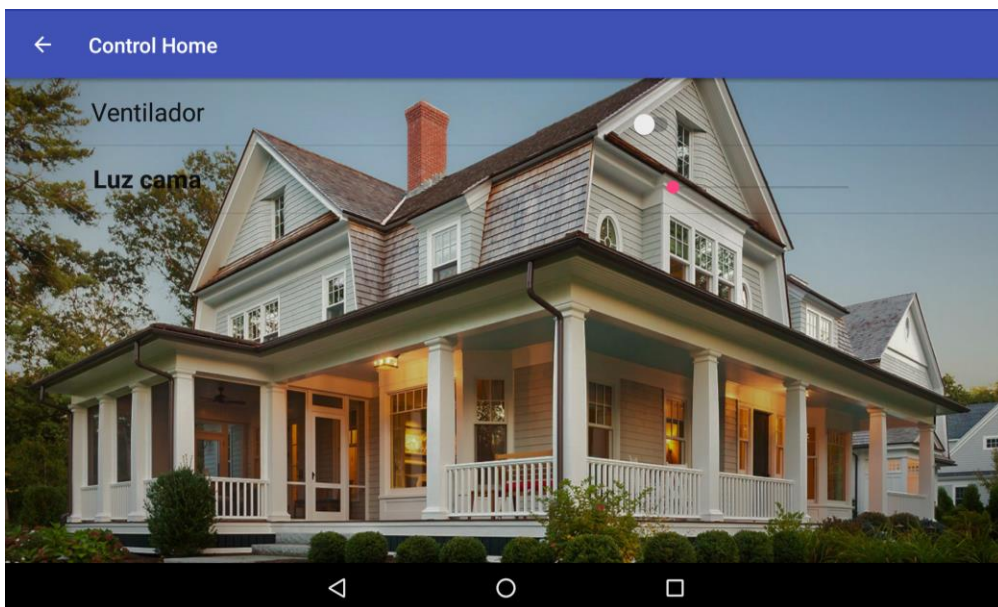
Es este caso iremos al botón de acciones que nos llevara a la vista de estancias orientadas a acciones



### Estancias acciones

**Imagen 13.55**

Como podemos observar en esta vista se muestra las estancias que cumplan los siguientes requisitos por un lado que tengan acciones asociadas e indirectamente que tenga actuadores y no sensores.

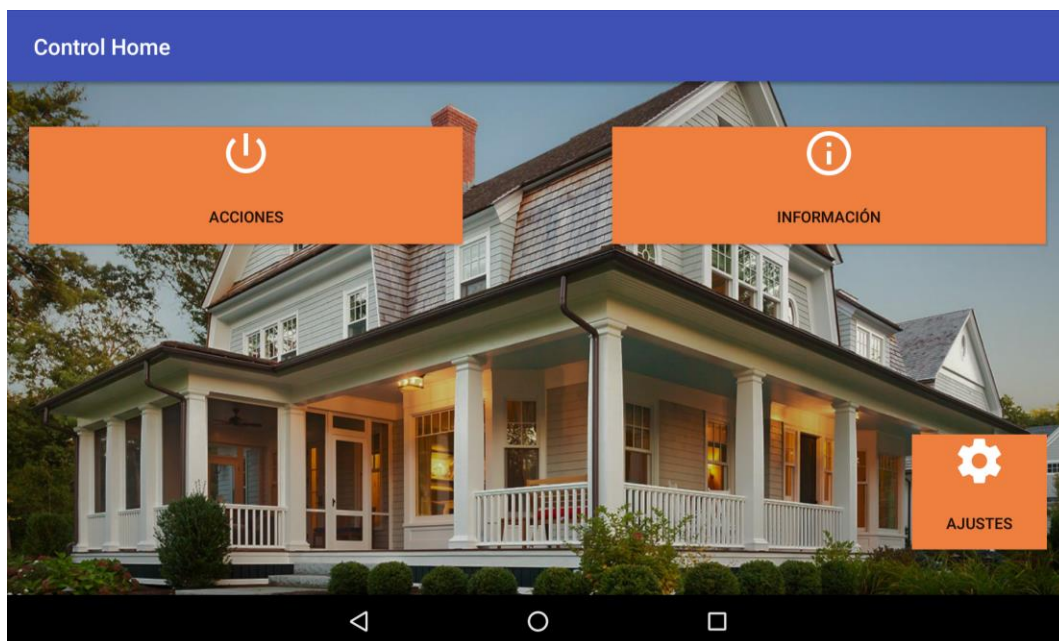


### ACCIONES 7-12

Para finalizar nos mostrar las acciones disponibles para esta estancia que como se puede observar en el ejemplo aquí tendríamos las dos posibles opciones que proporciona el sistema que es un elemento binario que sería el botón (Ventilador) o en el caso de un abanico más grande de valores que sería el slide (Luz cama).

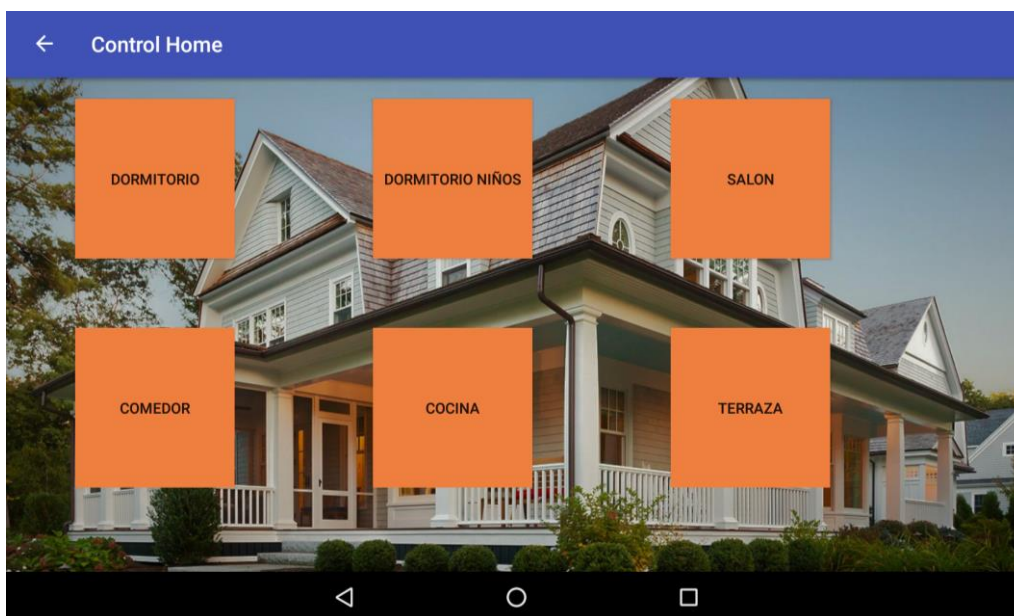
### Información de la casa





### Menú principal

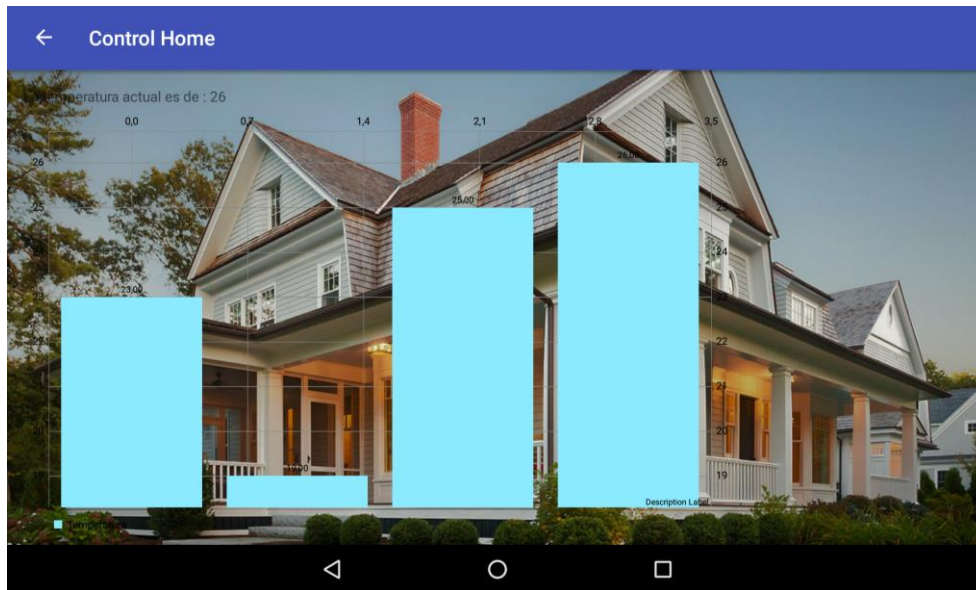
Para esta prueba nos iremos a la opción de información.



### Vista de estancias

Como se puede observar en la imagen anterior las estancias mostradas son distintas a las que se mostraron en el ejemplo de acciones.

A continuación, se selecciona la estancia de la cual queremos obtener dicha información.



### INFORMACIÓN 7-13

En esta vista podemos observar mediante un gráfico el historial de los últimos siete días sin embargo si no tuviéramos todos estos datos solo mostraría los datos de los días que tiene como se puede apreciar en el ejemplo.

En la parte superior de la vista se puede apreciar la temperatura actual de la estancia.

## 8 Conclusiones

Como se ha podido observar en el desarrollo de esta memoria y las pruebas realizadas, es una tecnología que da la posibilidad de ser ampliada fácilmente ya que haciendo algunos cambios se podrán crear más funcionalidades como por ejemplo control por voz y automatización referente a estados del medio.

Con este proyecto he podido unir diferentes aspectos que he ido adquiriendo durante toda mi formación en la escuela de Ingeniería informática impartida por diferentes profesores que cada uno ha dado su granito de arena para que al final de todo salir con unos conocimientos amplios en el mundo de la informática.

Referente a la tecnología usadas en este proyecto so podrían haber usado tecnologías diferentes, pero se ha abogado por el uso de esta ya que se tenía mayor dominio en algunas de ellas y en el resto ha sido posible su aprendizaje.

También ha servido para saber de lo que eres capaz una vez que acabas la carrera ya que es una de las primeras veces que está más solo ante el resto de desarrollar algo y tienes que tomas las decisiones referentes al desarrollo en todas las fases, eso si contando con la ayudada del tutor que te guían para que puedas conseguir tus objetivos.

## 9 Bibliografía

---

Malaysian arduino myardu- ESP8266 Reading and writing to esp8266  
EPROM: <http://www.slumberjer.com/myardu/author/slumberjer/>

Android: El Gran Libro de Android 5ª edición  
Autor: Jesús Tomás Gironés  
<https://www.amazon.es/El-Gran-Libro-Android-5%C2%AA/dp/8426722563>

Arduino: Manual de Programación Arduino  
Asignatura Diseño e Integración de Periféricos  
[https://poliformat.upv.es/access/content/group/GRA\\_11635\\_2016/material%20de%20apoyo/TUTORIAL/Manual%2BProgramacion%2BArduino.pdf](https://poliformat.upv.es/access/content/group/GRA_11635_2016/material%20de%20apoyo/TUTORIAL/Manual%2BProgramacion%2BArduino.pdf)

\*Curso Android Studio- conexiones bluetooth Android con Arduino  
Autor: Ramiro Madriaga  
<http://cursoandroidstudio.blogspot.com.es/2015/10/conexion-bluetooth-android-con-arduino.html>

MULE – Asignatura Integración de aplicaciones  
Material de la asignatura