

Sistema a la medida para el control y monitoreo de una casa inteligente por medio de servidor web

Malcolm Davis

Computer Engineering

mdavis.cr@ieee.org

David Monestel

Computer Engineering

mdavis.cr@ieee.org

Fabian Solano

Computer Engineering

Email: mdavis.cr@ieee.org

Resumen—En este proyecto se demuestra la utilidad de los sistemas a la medida con el desarrollo de un prototipo de domótica que utiliza un sistema operativo GNU-Linux a la medida sobre un sistema embebido con procesador ARM. El prototipo puede simular el manejo del estado de las luces, puertas, una ducha y una cámara IP utilizando un raspberry pi como dispositivo de desarrollo principal.

Palabras clave: Domótica, Raspberry Pi, Yocto, IoT, Servidor Web, Computación Ubicua, Sistemas Embebidos, TEC.

I. INTRODUCCIÓN

El principal objetivo de la computación ubicua es la creación de productos inteligentes conectados que tengan una alta disponibilidad, y que hagan la comunicación más fácil[1]. Estos dispositivos son utilizados para integrar elementos del entorno del ser humano facilitando su quehacer diario.

La domótica es la combinación de los sistemas a la medida, las tecnologías de la información y la construcción; consiste en utilizar los mismos para crear sistemas de computación ubicua que se integran con las casas logrando optimizar el uso de los recursos y creando facilidades de control remoto de algunos componentes de ellas. Nace del auge de los sistemas a la medida y tendencias como el IoT. Este documento resume el proceso realizado para el diseño y desarrollo de una "casa inteligente". Esto con la finalidad de demostrar la utilidad de

estos sistemas mediante el desarrollo de un prototipo de una casa inteligente que, con el uso de un sistema a la medida implementado en un raspberry pi puede ser controlada desde un dispositivo móvil.

II. SISTEMA DESARROLLADO

Para proponer el prototipo del sistema desarrollado en proyecto se utilizó la metodología modular, se decidió de este modo para así facilitar el diseño individual de cada uno de los componentes o soluciones de subproblemas con menor complejidad que al final se unen para componer el sistema completo [2]. La subdivisión de estos problemas se hace en subsistemas individuales que cumplen con los criterios de desarrollo y pueden ser utilizados para crear el sistema completo, más adelante se explica cada uno de ellos. Para conocer más sobre la implementación de cada modulo, puede referirse al documento de diseño que se entregó junto con este proyecto.

II-A. Servidor Web

Para poder mantener la disponibilidad de las funciones del sistema de manera remota, se implementó un servidor web en la placa de desarrollo Raspberry Pi 2. Se implementó en el lenguaje de programación C con la biblioteca de sockets del sistema. Se definió un protocolo de mensajes que utiliza tanto

el servidor como los clientes que se conecten al sistema así como usuarios y contraseñas por defecto. El sistema utiliza un token aleatorio generado por cada sesión para así poder validar las ordenes que le son enviadas.

II-B. Biblioteca I/O

Para poder utilizar la funcionalidad de entrada y salida general (GPIO) del Raspberry Pi, se implementó una biblioteca en C que accede y modifica a los registros necesarios para interactuar con los pines de entrada y salida para así activar o desactivar dispositivos conectados. Esta modificación de registros en Linux funciona como la escritura o lectura de un archivo [?].

II-C. Control de Luces, puertas y Ducha

El sistema es capaz de encender y apagar las luces de las diferentes habitaciones de la casa que son representadas por leds. Así como abrir o cerrar las puertas que se representan mediante diferentes actuadores. Por último, se agregó al baño una bomba de agua que activa la ducha. El estado de cada dispositivo se puede ver en la aplicación móvil, y además se puede modificar.

II-D. Cámara IP

Para representar la cámara IP se utiliza el módulo de cámara de Raspberry Pi. Para poder tomar las fotografías se agregó a la imagen creada con Yocto el modulo "userland" que permite la utilización de comandos que controlan el módulo de la cámara. Esta foto posteriormente es enviada para su visualización al cliente para realizando una serialización en base 64[4].

II-E. Aplicación Móvil

La aplicación móvil es el punto de contacto que utiliza el usuario para poder interactuar con los dispositivos de la casa, desde la misma se pueden utilizar las funcionalidades mencionadas anteriormente. La aplicación se realizó desde 0 con android puro con el programa android studio. Se genera un instalador que puede ser instalado en cualquier dispositivo con android 5.0 en adelante. Para la comunicación con el servidor se utilizaron los sockets de java.



Figura 1. Diseño de Maqueta

II-F. Maqueta/Dispositivos

Para poder mostrar la funcionalidad del sistema en el mundo real, se elaboró una maqueta a escala de una casa que contiene las características descritas anteriormente. Para lograr esto se realizaron cortes sobre una lamina de MDF creando las paredes y divisiones de la casa, posteriormente se agregaron detalles para contextualizar los espacios y de último se agregan los cables necesarios para utilizar los dispositivos. En la figura 1 se puede observar el diseño de la maqueta.

III. RESULTADOS

El primer resultado que se obtuvo está relacionado con las pruebas de la biblioteca de entrada y salida, este fue el primer módulo que se implementó y luego de la compilación cruzada se logró probar que todos los elementos estaban funcionando según lo esperado. El diagrama esquemático de la distribución de los componentes puede ser observado en la figura se desarrolló una placa para poder conectar los dispositivos con mayor facilidad, un resumen de estas conexiones puede observarse en la figura 3. Posteriormente se implementó la maqueta obteniendo el resultado mostrado en la figura 2. Posteriormente se probó el funcionamiento del servidor en la placa de desarrollo Raspberry Pi junto con la integración con la biblioteca implementada y las funciones externas de captura de fotografía. Tanto el servidor como la biblioteca funcionaron según lo esperado, así como la captura de imágenes con el modulo externo. Y la comunicación con un cliente en PC. Por último, para la implementación y pruebas de la aplicación de pruebas se utilizó el modo de debugueo de Android Studio



Figura 2. Implementación de Maqueta

Grupo Pines B1

Ground	Servos 1 - 4	In Luces 1 - 5	Out Botones 1 - 4	3.3V Raspberry	3.3V Botones
1	2 3 4 5	6 7 8 9 10	11 12 13 14	15	16

Grupo Pines B4

Ground	5V	Out Servos 5 - 1
1	2	3
4	5	6
7	8	9
10	11	12
13	14	15

Grupo Pines B2

In Botones 1 - 4
1 2 3 4

Grupo Pines B3
1
2
3
4
5
6
Ground

Figura 3. Conexiones de los Dispositivos

que permite una instalación rápida así como una terminal de debugueo para la aplicación, la conexión con el servidor y la invocación de las funcionalidades planeadas para el sistema se dió correctamente, en la figura 4 se pueden observar las principales pantallas de la aplicación. Por lo tanto, y según el alcance del proyecto, se dá por concluido el desarrollo del prototipo de domótica y se toma como un éxito.

IV. CONCLUSIONES

Es importante para este tipo de sistemas tener claras las limitaciones a la hora de diseñar las funcionalidades, para así poder aprovechar al máximo las capacidades del sistema. El utilizar un sistema operativo a la medida ayuda a evitar el desperdicio de recursos en procesos que no se van a utilizar y espacio que puede ser utilizado por otra información.

El éxito de estos sistemas se da en parte por la facilidad que aporta al usuario, así que en la etapa de diseño se debe de planear tomando en cuenta al usuario cual es la mejor

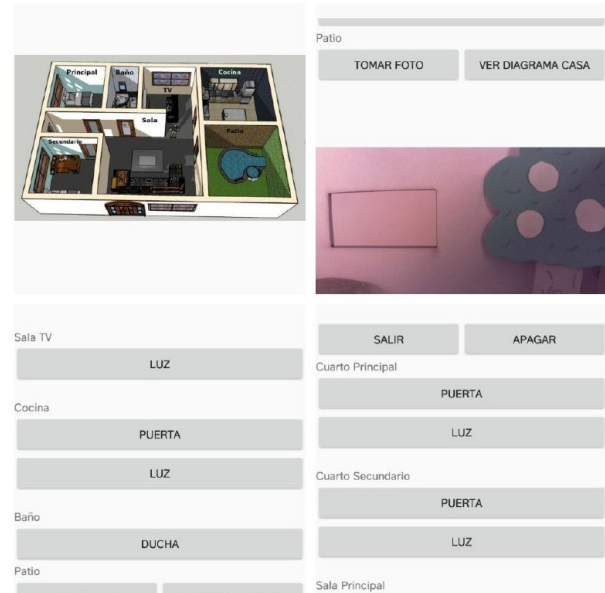


Figura 4. Aplicación Móvil

estrategia para interfazear al mismo con el sistema.

REFERENCIAS

- 1 "What is Ubiquitous Computing? - Definition from Techopedia", Techopedia.com, 2018. [Online]. Available: <https://www.techopedia.com/definition/22702/ubiquitous-computing>. [Accessed: 21 - Sep- 2018].
- 2 J. Leiva, "Diseño de Algoritmos", Lcc.uma.es. [Online]. Available: <http://www.lcc.uma.es/jlleivao/algoritmos/t2.pdf>. [Accessed: 21- Sep- 2018].
- 3 Aguilar, M, 2009. "Tutorial Linux". [Online] Available: TEC Digital
- 4 raspberrypi/userland", GitHub. [Online]. Available: <https://github.com/raspberrypi/userland>. [Accessed: 21- Sep- 2018]. 2008.