



# Arquitecturas específicas y empotradas: MagicMirror

MAGIC MIRROR CON CONTROL POR VOZ, USUARIOS Y EDICIÓN EXTERNA

ADRIANA ALONSO YUGUEROS

ANTONIO BARRAL GAGO

PABLO ÁNGEL ÁLVAREZ FERNÁNDEZ

MÓNICA ALONSO MATEOS

08 DE ENERO DE 2018, ARQUITECTURAS ESPECIFICAS Y EMPOTRADAS

## ÍNDICE

|  |    |
|--|----|
| 1. Índice de imágenes .....                                    | 2  |
| 2. Índice de plano.....  | 3  |
| 3. Resumen de la idea del proyecto y cómo surge. ....          | 4  |
| 4. Objetivos iniciales del proyecto .....                      | 5  |
| 5. Organización del grupo y las tareas .....                   | 6  |
| 6. Cronograma del proyecto.....                                | 7  |
| 7. Descripción de los materiales utilizados .....              | 9  |
| 8. Descripción de las herramientas de Software utilizadas..... | 10 |
| 8.1. Amazon Alexa .....  | 10 |
| 8.2. MySQL y PHPMyAdmin .....                                  | 10 |
| 8.3. Python .....  | 10 |
| 8.4. HTML, CSS, PHP y JavaScript .....                         | 11 |
| 8.5. C++ Arduino .....   | 11 |
| 8.6. Apache .....  | 12 |
| 8.7. Bash.....   | 12 |
| 8.8. Raspbian.....   | 12 |
| 9. Desarrollo y montaje .....                                  | 13 |
| 9.1. Montaje.....  | 13 |
| 9.2. Desarrollo.....   | 15 |
| 9.2.1. Base de datos .....                                     | 15 |
| 9.2.2. Instalación y configuración de Alexa .....              | 16 |
| 9.2.3. Back-end.....   | 17 |
| 9.2.4. Front-end .....   | 18 |
| 10. Resultados obtenidos .....                                 | 25 |
| 11. Presupuesto .....  | 26 |
| 12. Posibles mejoras o evoluciones .....                       | 27 |
| 13. Bibliografía.....  | 28 |

# 1. Índice de imágenes

|                               |    |
|-------------------------------|----|
| 1) Diagrama de Gant           | 8  |
| 2) inicio.html                | 20 |
| 3) registrar.php              | 21 |
| 4) sinregistrar.php           | 22 |
| 5) personalizar.php           | 23 |
| 6) Diagrama de funcionamiento | 24 |

## 2. Índice de plano

|   |    |
|---|----|
| 1) Esquema general del espejo                   | 13 |
| 2) Esquema módulo RFID con Raspberry Pi 2       | 14 |
| 3) Esquema sensor de movimiento con Arduino Uno | 14 |
| 4) Esquema entradas Raspberry Pi 2              | 15 |

### **3. Resumen de la idea del proyecto y cómo surge.**

La idea consiste en desarrollar un Magic Mirror, que se trata de un espejo que cuando detecta movimiento, muestra algunos datos de carácter genérico como son el tiempo, la fecha, la hora... Este Magic Mirror tiene de especial que contiene usuarios y cada uno de ellos puede personalizar lo que el espejo muestra cuando acceden a su espacio. Cada usuario tendrá a su disposición una tarjeta o un llavero de tipo RFID para poder acceder. En el caso de que el Magic Mirror lea una nueva tarjeta RFID, se registrará automáticamente a ese nuevo usuario con sus preferencias sobre qué elementos quiere y de qué tamaño. Siempre que el espejo detecte movimiento o un usuario acceda a su espacio se mostrará una frase motivadora. Además, tiene incorporado un reconocimiento de voz que le da más funcionalidad al espejo, como por ejemplo poder encender la radio para escucharla.

El cristal que se utiliza es un cristal como el que se usa en comisaría. También se necesita un marco para ocultar los componentes del Magic Mirror, como puede ser la raspberry.

Al principio de esta asignatura un miembro del grupo ya tenía en mente este proyecto, ya que era muy útil para interactuar con este espejo en casa. Este miembro propuso al resto de desarrollar el Magic Mirror, y se aceptó. A la hora de escoger proyecto se vio que fue desarrollado por otros alumnos el curso anterior, por lo que todo el proceso de construcción del espejo no sería necesario. Además, se habló con el profesor para que diese su consentimiento para hacer un Magic Mirror como proyecto de la asignatura. El profesor aceptó la propuesta con la condición de que se desarrollase alguna característica que el año pasado no implementaron los alumnos anteriores. El profesor proporcionó la idea de que el proyecto tuviese diferentes usuarios que vendrían dados por tarjetas RFID. El reconocimiento de voz por Alexa y la posibilidad de poder personalizar el Magic Mirror fueron propuestas por el grupo.

## 4. Objetivos iniciales del proyecto

A continuación, se explicarán los objetivos principales planteados al comienzo del proyecto:

- Control por voz: Una de las ideas principales de este proyecto, es integrar la capacidad de controlar mediante la voz, ciertas órdenes para realizar diferentes funciones con el espejo. Esta implementación se realizó con la herramienta software llamada Alexa, que se explicará más adelante. Este objetivo utilizaría un micrófono ya implementado en el espejo.
- Página base: Un objetivo predefinido para este proyecto es crear una página principal, que sea común para todos los usuarios. Esta situación se presentará en el caso de que el usuario que quiera acceder al espejo, no esté registrado o no se acceda con ninguna RFID. Esta página mostrará elementos base como puede ser la hora, la fecha y el tiempo; sin dar opción a modificarla.
- Tarjetas RFID: Este objetivo fue propuesto al principio del proyecto por el profesor. Consiste en un método de logueo, en el que el usuario poseerá una tarjeta con un número identificativo. Esta identificación hará posible el reconocimiento del usuario. Al pasar la tarjeta por delante del lector RFID, el espejo mostrará el espacio elegido por el usuario.
- Registro: Otro objetivo inicial que se planteó, consistía en otorgar a cualquier usuario registrarse obteniendo una tarjeta RFID. Este proceso se basa en que cualquier tarjeta no almacenada en la base de datos del espejo, podrá ser registrada, únicamente pasándosela al lector RFID. Se almacenará en la base de datos asignando a ese usuario poseedor de la tarjeta un identificador. Con ese identificador se podrá proceder a la personalización de los elementos.
- Personalización: Otro objetivo principal del proyecto es otorgar al usuario la posibilidad de editar la información que prefiere que aparezca en la página principal del Magic Mirror. Esta tarea se realizará a través del móvil del usuario. Tan solo debe de recordar su id e introducirla para entrar en la personalización de su página. En el dispositivo telefónico, le aparecerá un formulario para que elija los elementos que desea que se muestre. Al terminar esta operación se actualizarán esos elementos en la base de datos para mostrárselos cuando el usuario acceda a su espacio.

## 5. Organización del grupo y las tareas

Estas son las tareas que ha habido durante el proyecto y el reparto entre los diferentes componentes del grupo:

| TAREA  | ORGANIZACIÓN      |
|--|-------------------|
| Elaborar piezas para la sujeción de la Raspberry, RFID, Arduino y micrófono.     | Pablo y Antonio   |
| Diseño del cableado del Arduino, RFID y el sensor de movimiento                  | Adriana y Antonio |
| Organizar y colocar los componentes del espejo                                   | Pablo y Adriana   |
| Implementar el sensor de movimiento a través de Arduino                          | Antonio           |
| Implementar la funcionalidad del micrófono a través de la herramienta Alexa.     | Antonio           |
| Buscar más variedad de elementos para su presentación en la web                  | Adriana y Mónica  |
| Implementar la personalización desde el teléfono móvil                           | Adriana           |
| Implementar la página principal  | Pablo y Mónica    |
| Diseñar la base de datos   | Todo el grupo     |
| Implementar la página del usuario registrado                                     | Mónica            |
| Implementar la página del usuario no registrado                                  | Adriana           |
| Script para mostrar frases aleatorias al usuario                                 | Mónica            |
| Implementar el controlador de los elementos mencionados anteriormente, en Python | Antonio y Pablo   |
| Pruebas  | Pablo             |
| Documentación  | Todo el grupo     |

## 6. Cronograma del proyecto

Antes de mostrar un cronograma, primero se procederá a enunciar los diferentes hitos que tiene el proyecto, y con la fecha que se propuso:

| Hito  | Fecha inicial | Fecha final |
|---|---------------|-------------|
| Instalación de Raspbian SO  | 06/11/2017    | 06/11/2017  |
| Elección de la idea   | 13/11/2017    | 13/11/2017  |
| Elección de objetivos   | 20/11/2017    | 20/11/2017  |
| Planificación del proyecto  | 27/11/2017    | 27/11/2017  |
| Reparto de tareas   | 27/11/2017    | 27/11/2017  |
| Construcción inicial del espejo   | 27/11/2017    | 04/12/2017  |
| Impresión piezas 3D   | 04/12/2017    | 10/01/2018  |
| Desarrollo código Arduino   | 04/12/2017    | 11/12/2017  |
| Implementación de la base de datos                                      | 04/12/2017    | 27/12/2017  |
| Desarrollo código Python  | 04/12/2017    | 03/01/2018  |
| Implementación de la herramienta Alexa                                  | 04/12/2017    | 27/12/2017  |
| Desarrollo código HTML y PHP  | 18/12/2017    | 03/01/2018  |
| Ensamblamiento de piezas 3D y demás componentes (altavoz, micrófono...) | 18/12/2017    | 10/01/2018  |
| Pruebas   | 03/01/2018    | 15/01/2018  |
| Corrección de errores   | 03/01/2018    | 15/01/2018  |
| Desarrollo de la memoria  | 08/01/2018    | 17/01/2018  |



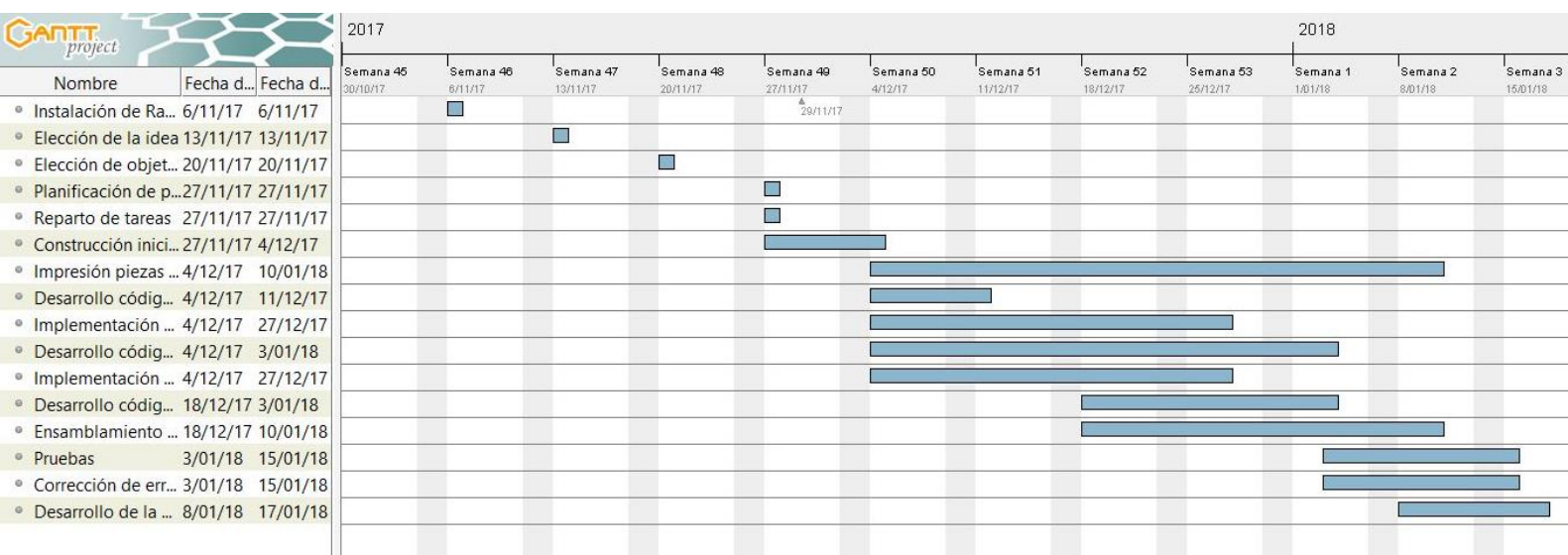


Foto 6.1. Diagrama de Gant

## 7. Descripción de los materiales utilizados

- Materiales reutilizados del curso anterior:
  - Estructura de madera del espejo: utilizado para el marco del Magic Mirror. Esta madera está pintada de negro.
  - LCD de una pantalla: En esta pantalla se mostrarán todos los datos que se ilustrarán en el espejo.
  - Micrófono: Situado en la parte superior del micrófono.
  - Cristal reflectante: Es el espejo en el que se podrá reflejar el usuario y también por el cuál a través se mostrarán los datos.
  - Cable VGA: Cable que se conecta desde la pantalla al adaptador de HDMI.
  - Adaptador de cable VGA a HDMI: Se conecta desde el cable VGA a la pantalla.
  - Adaptador USB para micrófono y altavoces
  - Raspberry: Situado en el interior del espejo.
  - Arduino: Situado al lado de la Raspberry, en el interior del espejo.
- Piezas 3D:
  - Unión entre Raspberry y Arduino dejando zonas de respiración por encima y por debajo.
  - Sujeción orientada para el micrófono.
  - Sujeción del RFID.
  - Sujeción interna del altavoz.
- Raspberry:
  - Cables: Conectan los componentes.
  - Placa lectura/escritura RFID: Placa situada en un lateral del Magic Mirror.
- Arduino:
  - Cables: Conectan los componentes
  - Sensor de movimiento: Situado en la parte inferior frontal del espejo.

Descripción de piezas 3d y la referencia a su plano que irán en otro fichero y las referencias estarán en el índice de planos.

## 8. Descripción de las herramientas de Software utilizadas

En esta sección se van a exponer las diferentes herramientas de software con una pequeña descripción de por qué se utilizan y donde, además de su funcionalidad:

### 8.1. Amazon Alexa

Es el asistente virtual de Amazon. La razón de utilizar Alexa fue que el resto de asistentes, o había que pagar por su uso (Google) o estaba desactualizado y abandonado (Jasper).

Se intentó exhaustivamente con Jasper, debido a que estaba creado principalmente para Raspberry Pi, pero dio una gran cantidad de problemas y su documentación estaba obsoleta por años.

La funcionalidad de Alexa es poder obtener datos del tiempo, escuchar música o escuchar las noticias sin necesidad de identificarse en el espejo, entre otras. Además, Alexa tiene una gran cantidad de comandos que pueden ser útiles, aunque estén menos relacionados con la funcionalidad del espejo. (<https://www.cnet.com/how-to/amazon-echo-the-complete-list-of-alexa-commands/>) No todos están disponibles en esta versión.

El punto negativo de Alexa, es que su versión “*open source*” para Raspberry Pi todavía le queda añadir muchas funcionalidades y tiene fallos (e.g: cuando se apaga la radio Alexa tiene un error y deja de funcionar).

### 8.2. MySQL y PHPMyAdmin

Ambas herramientas forman la base de datos donde se almacena la información de cada usuario, siendo la primera el lenguaje, y la segunda, el entorno desde donde se ha trabajado con MySQL. La razón de usar estas tecnologías fue que son las más comunes usando una Raspberry Pi.

### 8.3. Python

Fue el lenguaje utilizado para prácticamente toda la funcionalidad back-end de la aplicación. El archivo escrito en Python es el encargado de comunicarse con la base de datos y la parte web de la aplicación.

Dentro de esta herramienta de software se han utilizado diferentes librerías para poder interactuar con las tecnologías del espejo. Las más importantes son:

- PyMySQL: Librería que soporta MySQL en Python. Era de las más sencillas para Python 3
- MFRC522: Librería utilizada para poder leer las tarjetas RFID.
- RPI.GPIO: Librería utilizada para poder leer al lector de tarjetas RFID en los pines de la Raspberry.
- Serial: Librería usada para conectar la Raspberry con Arduino.

Se utilizó Python debido a que es el lenguaje más común en una Raspberry, debido a su facilidad de uso y también a las librerías disponibles.

## 8.4. HTML, CSS, PHP y JavaScript

Estas tecnologías son usadas para la creación de las webs que se van cargando en el espejo (tanto la base como las de los usuarios). Además de componer con HTML, la página que se muestra al usuario, se realiza la funcionalidad gracias a PHP y JavaScript. Por último, se le dan los estilos a este espacio con el lenguaje de CSS. Por supuesto, estos lenguajes hacen llamadas a API's para obtener la información requerida, se conectan con la base de datos, etc...

Su uso es básicamente necesario para poder crear las aplicaciones web que contiene el Magic Mirror, es decir, para la creación de los espacios personales de los diferentes usuarios.

## 8.5. C++ Arduino

Uso del C++ específico de Arduino para poder trabajar en dicho sistema. Es necesario para crear un script que use el sensor de movimiento, y este detecte si hay presencia y así, pasándole dicha información a la Raspberry, esta se encargara de encender la pantalla del espejo.

La decisión de implementar el reconocimiento del sensor de movimiento en Arduino fue para distinguir que un dispositivo se encargase de detectar los datos externos, y el otro se encargase de mover la aplicación en sí.

La razón por la cual la detección de RFID no fue implementada por Arduino fue que con este no podríamos escribir información en las tarjetas, mientras que con la Raspberry si existe esa opción, aunque al final no haya sido utilizada.

## **8.6. Apache**

El uso de Apache es necesario para tener un servidor HTML para que funcione tanto la base de datos como las diferentes aplicaciones web que se utilizan.

## **8.7. Bash**

Bash se usa para la creación de dos scripts que se ejecutan al iniciarse el sistema. Uno se encarga de ejecutar Alexa y el otro está encargado de mantener funcionando el script correspondiente del funcionamiento del espejo.

## **8.8. Raspbian**

Es el sistema operativo que se ha utilizado para la Raspberry Pi. La razón de su uso es que se usó desde la práctica relacionada con el aprendizaje de Raspberry a principio de curso.

## 9. Desarrollo y montaje

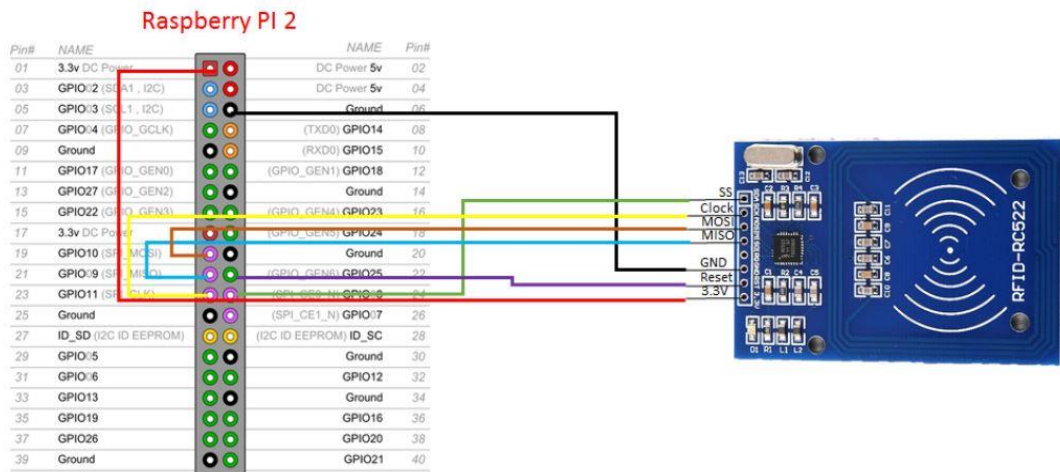
### 9.1. Montaje

Se empieza mostrando un esquema general del espejo, con los materiales reutilizados del año anterior. En dicho esquema se pueden ver las medidas del espejo en general, pero también de parte de sus componentes. El esquema está hecho sin la tapa trasera que protege todos los componentes además de proporcionar diseño.



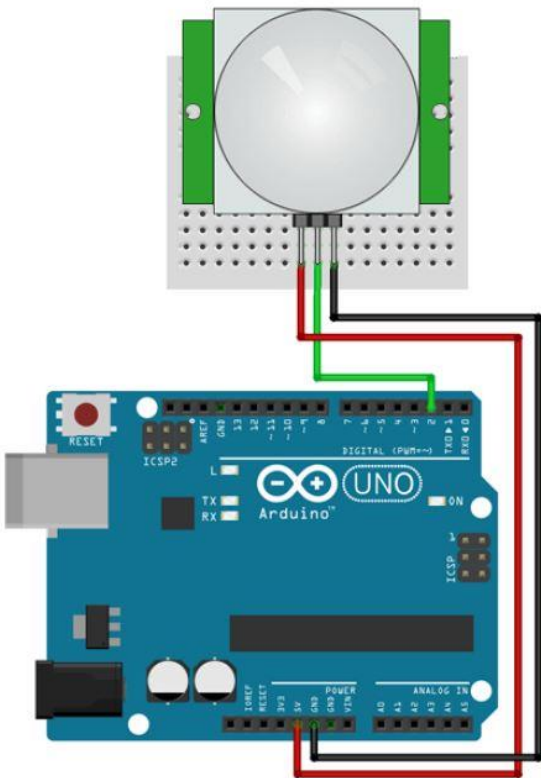
Esquema 9.1.1. Magic Mirror

Este esquema electrónico muestra la conexión entre el módulo RFID utilizado con la Raspberry Pi 2 utilizada. Este módulo es el encargado de leer las tarjetas RFID de los usuarios y pasarle el número de identificación a la Raspberry para su posterior uso.



Esquema 9.1.2. Módulo RFID con Raspberry Pi 2

El siguiente esquema muestra la conexión entre el sensor de movimiento utilizado con la placa Arduino Uno. Su función consistirá en detectar, mediante el movimiento, la presencia de algún usuario, evitando así que el Magic Mirror esté en uso constante.



Esquema 9.1.3. Sensor de movimiento con Arduino Uno

Este último esquema muestra todas las posibles entradas de la Raspberry Pi 2, a través de los diferentes puertos. En el propio esquema se puede observar la funcionalidad que recibirá cada puerto.



*Esquema 9.1.4. Entradas Raspberry Pi 2*

## 9.2. Desarrollo

En este apartado se comentará todo el proceso de desarrollo software del Magic Mirror desde el inicio, del cual partimos con la estructura del espejo montada, pasando por la implementación de una base de datos, una herramienta de reconocimiento de voz y el código necesario en diferentes lenguajes para la implementación del back-end y el front-end. Por último, se describirán los problemas que han surgido en los diferentes apartados.

### 9.2.1. Base de datos

Se decidió implementar una base de datos para abarcar parte de los objetivos propuestos del espejo. Como se ha explicado anteriormente, el Magic Mirror identifica los usuarios; por esa razón se ha creado una tabla llamada 'Usuarios' dentro de la base de datos 'MagicMirror'. Dicha tabla contiene información sobre el nombre del cliente, el código de la tarjeta RFID de dicho cliente, un código temporal usado para personalizar su espacio fijo personal y por último, un identificador único para cada usuario, que es auto-incremental, y por tanto, la clave primaria.

La tabla fue creada para almacenar los diferentes usuarios que estén registrados. Más tarde, se implementó que se añadiesen usuarios que no estuviesen registrados, realizando el front-end una tarea diferente a la que haría si se identifica un usuario registrado.

Una vez creada dicha tabla con su funcionalidad, se creó otra llamada 'Personalizaciones' que contiene las características del espacio fijo personal de



cada usuario. Es decir, si quiere mostrar los elementos disponibles con el tamaño y el tipo deseado (reloj, calendario, tiempo).

Ambas tablas están relacionadas por el ID del usuario en una relación 'One-to-One'.

El problema principal que se tuvo no fue relacionado con el diseño o implementación de la misma sino, más bien, con el acceso a ella de manera concurrente para el trabajo de front-end y back-end, es decir, fue un problema de comunicación en el equipo.

### 9.2.2. Instalación y configuración de Alexa

Cuando se decidió la instalación y uso de un asistente de voz para acompañar la funcionalidad del Magic Mirror, se optó principalmente por la instalación y configuración de la herramienta Jasper. La razón es que fue creado específicamente para usar en una Raspberry Pi.

Después de seguir la documentación oficial de su página, y haber concluido la instalación Jasper produjo muchos errores, lo que provocó que nunca llegase a funcionar. Se investigaron estos errores y se encontró que la documentación de Jasper estaba obsoleta desde hace 3 años. Se encontró un foro en el que continuaron con dicha documentación, pero los errores seguían persistiendo, por lo que se decidió cambiar de asistente de voz.

Se plantearon el asistente de voz de Google, y el asistente de voz de Amazon, Alexa. Se optó por Alexa ya que el de Google, después de un número de request era necesario pagar por este servicio.

Para la instalación de Alexa se siguió la documentación oficial en la plataforma de GitHub(añadir cita a la página). Aún después de realizar la instalación correctamente, aparecieron errores relacionados con la no identificación de micrófono y/o altavoz. La solución a estos problemas consistió en la desactivación de los puertos de entrada y salida de audios propios de la Raspberry Pi.

Una vez solucionados, se probaron e investigaron los diferentes comandos y funcionalidades que ofrecía Alexa, como ya fueron explicadas y mostradas en este documento con anterioridad.

La herramienta de Alexa provocó los siguientes problemas, algunos de ellos solucionados, y otros sin solución debido a que la versión "open-source" para Raspberry Pi está en fase Beta y está hecha para utilizarse en Estados Unidos (algunas aplicaciones no se pueden utilizar por este motivo). Los problemas serán enumerados a continuación:

- Errores relacionados con la entrada y salida de audio. **Solucionado.**
- Cuando se activa la radio y se apaga, Alexa en ocasiones produce un error irreparable.
- Incompatibilidad con algunas aplicaciones, como Spotify.

- La comunicación solo es posible en inglés y alemán.

### 9.2.3. Back-end

El desarrollo del back-end del Magic Mirror se dividió en 3 partes elementales. Un script para Arduino, un script en Python y un par de scripts en Bash para operaciones del OS de Raspberry.

#### Script Arduino

Este script programado en C++ de Arduino se encarga de recibir las señales de un sensor de movimiento, que detectará si un usuario se acerca o no al espejo, es decir, si detecta presencia o no.

Dicha salida es leída por el script en Python del que se hablará en el siguiente punto.

Por último, se pensó en un principio que Arduino también leyese el lector RFID, pero debido a que en Python permitía más funcionalidad con la manipulación de datos generados al pasar una tarjeta por el lector, se optó por no implementarlo con Arduino.

#### Script Python

El script de Python canaliza prácticamente toda la funcionalidad back-end del Magic Mirror. Debido a esto, este script ha sido una fuente de problemas ya que se ha necesitado modificar muchas veces y, por ende, se ha picado mucho código innecesariamente.

La primera función que realiza el script es leer la salida producida por el script de Arduino, y apagar la pantalla si al cabo de un minuto no se ha detectado un usuario, o encenderla, mostrando una página base común para todos los usuarios. Independientemente de si se activa o no el sensor de presencia, se hizo parte del script concurrente para poder loguearte sin tener que pasar por delante (no es algo común, pero podría darse dicha situación).

Cuando el script lee la tarjeta RFID se conecta a la base de datos para comprobar si dicho usuario está registrado o no. Pueden suceder 2 cosas:

- El usuario está registrado. Se accede a dicho usuario y se carga en una web su entorno personal con los parámetros que ha definido el usuario.
- El usuario no está registrado. Se crea un nuevo registro en la base de datos y se carga una página de bienvenida en la que se muestran las instrucciones para el manejo del Magic Mirror.

El script también es el encargado de crear el código temporal (el cual se calcula repetidas veces hasta que sea único en la base de datos) y guardarlo

en el campo correspondiente al usuario que ha pasado la tarjeta RFID por el sensor para loguearse.

Este script está configurado para lanzarse en cuanto la Raspberry se encienda mediante el Crontab.

## Scripts del SO

Por último, existen un par de scripts escritos en Bash que realizan funciones relacionadas con el SO, el arranque y el mantenimiento de algunas de las funcionalidades comentadas. Son los siguientes:

- execute\_alex. Se encarga de ejecutar Alexa cada vez que se reinicie el SO para que esté disponible desde el principio,
- checking\_script. Su función es comprobar que el script de Python no se encuentre caído. Si este se cae, el script informará de la hora de la caída y ejecutará de nuevo el script.

Una vez tratados los diferentes aspectos del desarrollo del back-end, se expondrán los problemas que fueron surgiendo, siendo todos ellos resueltos con más o menos esfuerzo:

- La salida del script de Arduino producía unos caracteres extraños al principio del string de la salida generada por el script. Fue un problema de codificación que se solucionó cuando se encontró.
- Gran cantidad de cambios que se tuvieron que producir en el script de Python debido al cambio de ideas y a que las tareas estaban divididas y hubo conflictos de código entre miembros del equipo. Eso produjo una abundante pérdida de tiempo.
- Problemas con la librería de MySQL para Python 3. Se debió a que fue complicado encontrar una que tuviese una fácil implementación y que fuera soportada por Python 3.
- Problemas de lectura de las tarjetas RFID. Era necesario instalar varias librerías como SPI de la cual se prescindió al principio y, lógicamente, produjo errores.

### 9.2.4. Front-end

En esta parte se explicará el código que realiza la funcionalidad de lo que mostrará el espejo. Este código está realizado en lenguajes PHP, JavaScript y HTML, además de CSS para adecuar los estilos al espejo. A continuación, se describirá la funcionalidad del código dividido por partes.

El código encargado de mostrar al usuario el espacio cuando no se loguea, está implementado en el lenguaje HTML, donde se muestran los elementos de la página principal, como puede ser reloj, calendario y tiempo. Este archivo se

llama *inicio.html*. El espacio se muestra cuando el sensor de movimiento detecta a una persona.

Cuando un nuevo usuario que no está registrado pasa su tarjeta RFID, al no estar el código de dicha tarjeta en la base de datos, se mostrará una página dando la bienvenida a este nuevo usuario. Dicha página mostrará el código temporal del usuario y la dirección a la que tiene que acceder para poder personalizar su nuevo espacio. El archivo encargado es *sinregistrar.php*.

El código para el usuario que está registrado en la base de datos y accede al Magic Mirror por logueo de la tarjeta RFID, está desarrollado en lenguaje HTML y PHP, de tal forma que, en este último, se produce el acceso a la base de datos, según el ID de usuario, que se le proporciona desde el archivo llamado *formulario\_temporal.html*, explicada anteriormente. Al realizar este acceso, se obtienen los datos de los elementos deseados por el usuario, que se utilizarán para guiarse al mostrar los tipos y tamaños. El archivo encargado de realizar esta función se llama *registrar.php*.

En la página principal y en la de logueo, se mostrará una frase motivadora, que se ejecuta en JavaScript, de tal manera que se lee un archivo *file.txt* que contiene todas las frases, y de ahí, se escoge un número aleatorio, que servirá para escogerla frase a mostrar.

Por otro lado, para la personalización del espacio es necesario que el usuario introduzca su código temporal. Si el código temporal es correcto, se accederá al formulario donde el cliente escoge lo que quiere que se muestre y de qué forma en su espacio. El login es ejecutado por el fichero *index.php* y la página de personalizaciones por *personalizar.php*.

Los problemas que sucedieron en el desarrollo de esta parte son los siguientes:

- El principal problema que se tuvo en esta parte, se produjo por el desacuerdo según los gustos de cada miembro del grupo, quedando en un acuerdo común al final. Para mostrar la página acordada, no se realizaron múltiples cambios, solamente la posición de los elementos y el tamaño de los mismos.
- Otro problema fue los errores que producían algunos de los archivos, cuando se trató de hacer funcionar todo el conjunto en la Raspberry Pi. Hubo un fichero que debido a los errores léxicos el sistema era incapaz de ejecutarlo y de encontrarlo. Por otra parte, ocurrió por las mismas razones un fallo en el fichero encargado de los usuarios registrados.

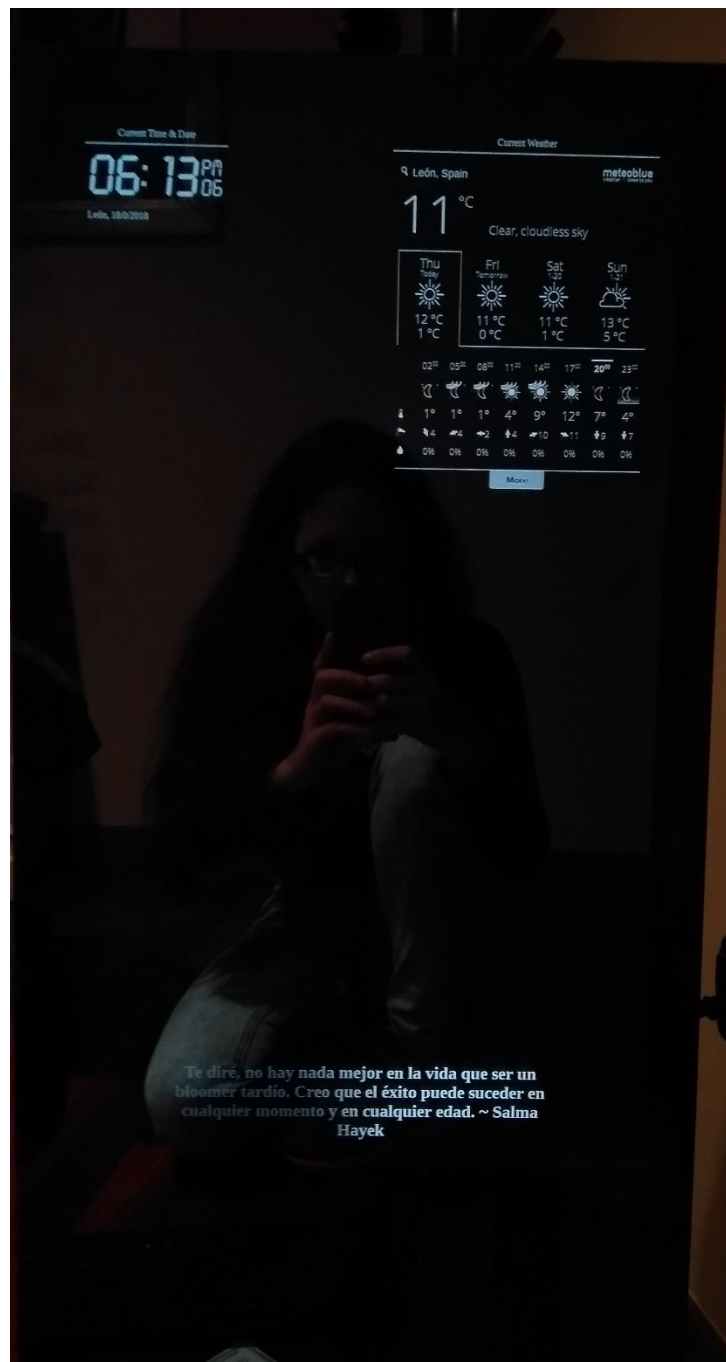


Foto 9.2.4.1. inicio.html

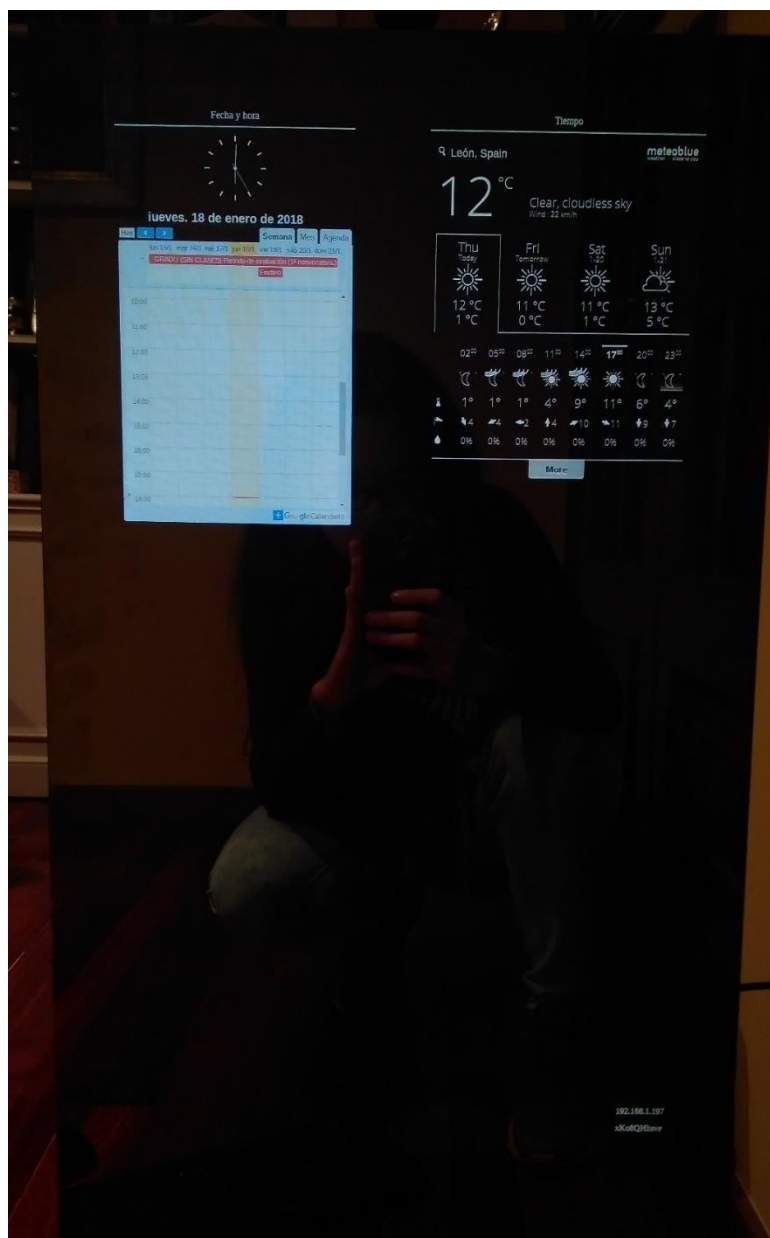


Foto 9.2.4.2. registrar.php

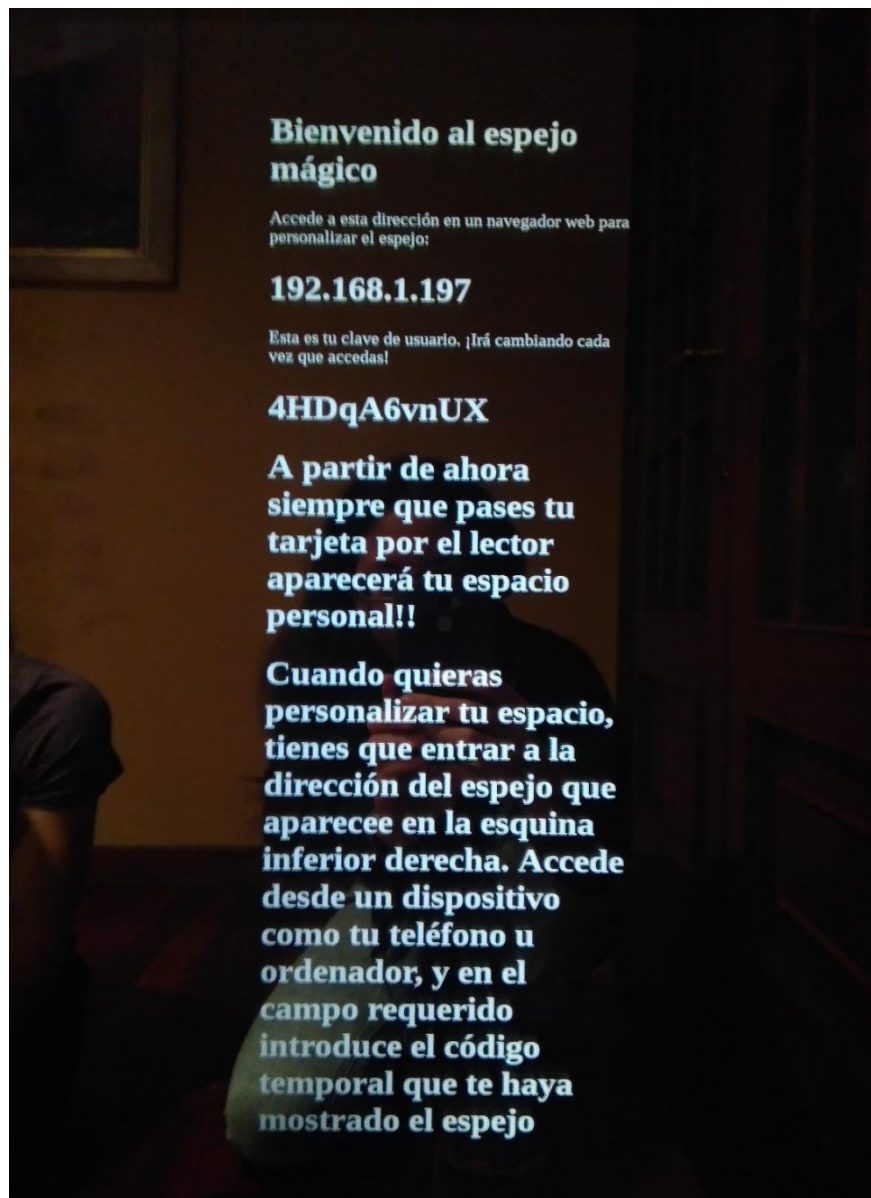


Foto 9.2.4.3. sinregistrar.php

Cambia tu nombre de usuario si lo deseas:

Pikachu

¿Quieres que aparezca un reloj en el espejo?

Sí

Selecciona un tamaño para el reloj

Mediano

Selecciona un tipo de reloj

Digital 3

¿Quieres que aparezca el tiempo en el espejo?

Sí

Selecciona un tamaño para el tiempo

Grande

Selecciona un tipo de tiempo

1

¿Quieres que aparezca un calendario en el espejo?

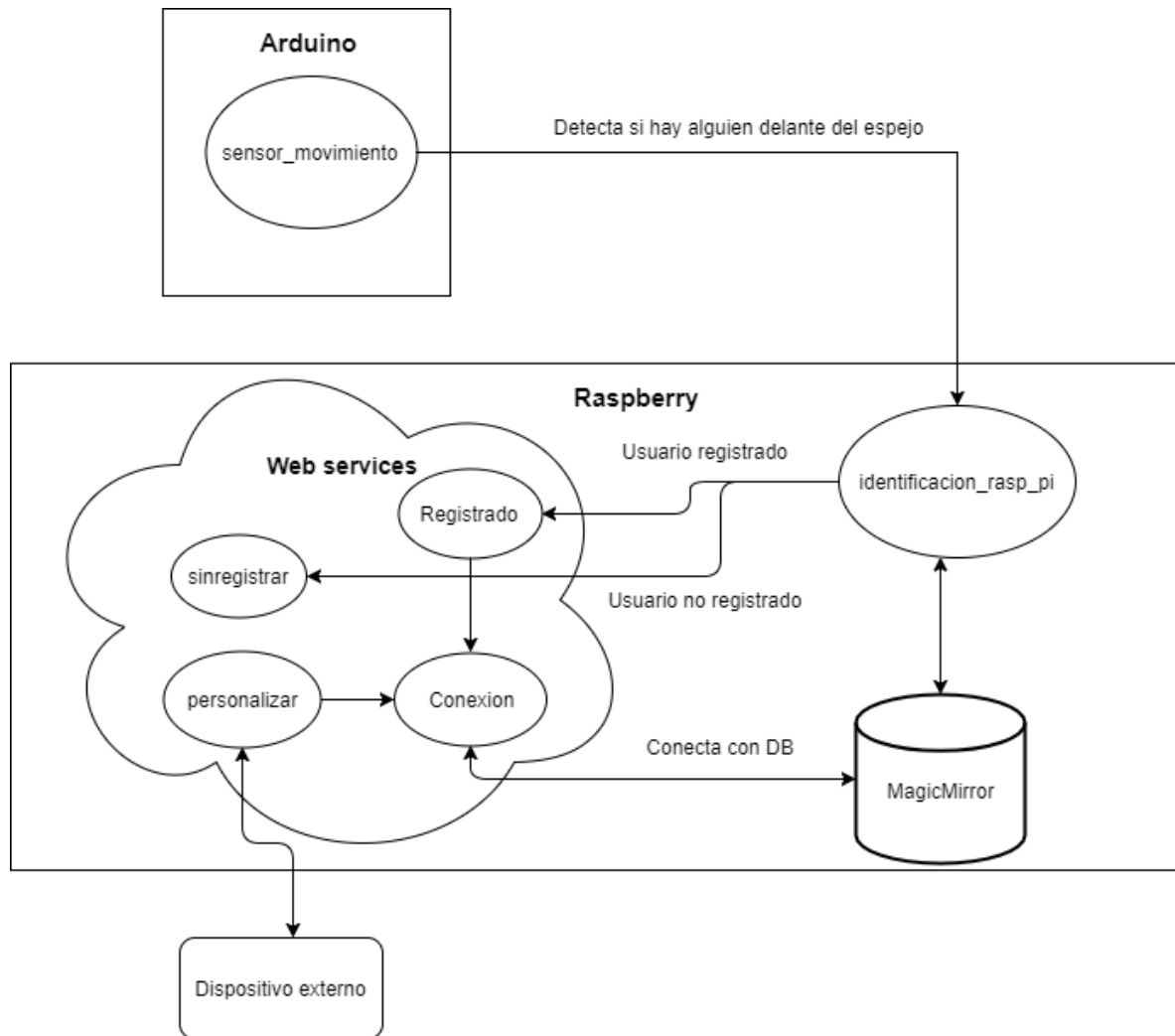
Sí

Cambiar apariencia

Foto 9.2.4.4. personalizar.php

Una vez explicado todo el desarrollo del Magic Mirror. Se ha creado un diagrama que explica a nivel general como están hechas las conexiones entre los diferentes componentes de software presentes en el proyecto:





*Foto 9.1. Diagrama de funcionamiento*

Por último, comentar que la mayor dificultad del proyecto residió en que fue muy complicado trabajar todo el grupo a la vez debido a que el Magic Mirror tiene un tamaño considerable, y su movilidad era bastante complicada a la par que delicada.

La mejor solución encontrada fue colocar el espejo en casa de uno de los integrantes del grupo y abrir una conexión remota para que se pudiese a trabajar en él a distancia. Aunque esta no fue una solución completa ya que sobrecargaba la Raspberry, solo permitía un usuario y la colapsaba en ciertos momentos por lo que ya casi al final se decidió quedar para unificar todo el proyecto y que todos conociésemos todos los detalles.

## 10. Resultados obtenidos

Una vez finalizado el proyecto, se ha conseguido crear un Magic Mirror que funciona a nivel general y con unas aplicaciones divididas en interacción por voz mediante Alexa, e interacción con un espacio personal de los diferentes usuarios que el Magic Mirror tenga almacenados, y los nuevos usuarios que se vayan registrando. Esto se traduce en una web personalizada para cada usuario.

La conclusión extraída es que, con un poco de optimización y *'refactoring'* del código para un mejor funcionamiento, el Magic Mirror construido es un producto que perfectamente se podría poner a la venta, y que se podría ir actualizando para mejorar su funcionamiento, sobre todo, en lo que se refiere a aspectos que dependen de terceros (Alexa).

# 11. Presupuesto

Como se ha explicado anteriormente, nuestro proyecto ha heredado parte de los materiales que se necesitaban, se ha elaborado un presupuesto en base a realizar este proyecto desde cero.

Por un lado, el presupuesto de los materiales:

| Material                          | Cantidad | Precio      |
|-----------------------------------|----------|-------------|
| Cristal especial                  | 1        | 30€         |
| LCD(Pantalla)                     | 1        | 100€        |
| Madera                            | -        | 5€          |
| Altavoz                           | 1        | 20€         |
| Micrófono                         | 1        | 5€          |
| Adaptador USB Micrófono y Altavoz | 1        | 3€          |
| Cable VGA                         | 1        | 2€          |
| Adaptador VGA a HDMI              | 1        | 6€          |
| Sensor de movimiento              | 1        | 1€          |
| Piezas 3D                         |          |             |
| Cables                            | 15       | 1€          |
| Raspberry                         | 1        | 30€         |
| Arduino                           | 1        | 25€         |
| <b>TOTAL</b>                      |          | <b>228€</b> |

Por otro lado, el presupuesto de la mano de obra:

| Trabajador            | Horas trabajadas | Precio/Hora | Precio        |
|-----------------------|------------------|-------------|---------------|
| Ingeniero Informático | 25               | 35€/hora    | 875€          |
| Ingeniero Informático | 25               | 35€/hora    | 875€          |
| Ingeniero Informático | 25               | 35€/hora    | 875€          |
| Ingeniero Informático | 25               | 35€/hora    | 875€          |
| <b>TOTAL</b>          |                  |             | <b>3.500€</b> |

Como el software solo se va a desarrollar una única vez, en el caso de vender el producto no significa que se venda por la suma de los presupuestos de los materiales y de la mano de obra. El presupuesto de los materiales se respetaría, pero el de mano de obra se reduciría mucho por lo explicado. Por lo tanto, en el caso de querer vender un Magic Mirror a un minorista, cada unidad saldría por un precio estimado de: **300,00€**

## 12. Posibles mejoras o evoluciones

Empezar diciendo que esta idea ya está bastante explotada en este proyecto. Al tratarse de un espejo, no puede cargarse con mucha información, ya que los usuarios deberían poder mirarse y contemplarse en el espejo. La opción de añadir usuario, y con ella, que cada uno escoja la información que desea mostrar ya está desarrollada en este proyecto. Es por eso que algunas de las posibles mejoras que se ven a corto plazo serían las siguientes:

- Mejorar el aspecto gráfico de la edición: Cuando un usuario accede a la página para personalizar su espacio es una página totalmente plana, sin ningún tipo de estilo.
- Añadir funcionalidades para Alexa: Incluir funciones como el reproductor de Spotify sería una buena opción. También existe la posibilidad de conectar vía wifi con regletas que permitirían encender y apagar objetos por voz. Esto no se pudo realizar debido a que la versión “*open source*” de Alexa todavía no permite tales funciones.
- Añadir variedad de elementos mostrados en la pantalla: Aunque ya hay diferentes relojes o diferentes formas de mostrar el tiempo, se podría añadir más opciones además de alguna característica más.
- Hacer un mejor diseño estéticamente en las diferentes páginas de cada usuario.

Una mejora a largo plazo consistiría en incluir en el Magic Mirror una cámara Kinect, por ejemplo, para poder detectar los movimientos de los usuarios. Así, se podría hacer que el Magic Mirror ganase funcionalidad. Para personalizar el espacio personal se podría hacer desde el propio espejo, mediante gestos. Además, por esta cámara, mediante reconocimiento facial, se podría detectar a los usuarios sin necesidad de utilizar las tarjetas RFID. Y al tratarse de una cámara, si el usuario lo desea, poder sacarse fotos y que estas se envíen directamente a un dispositivo como un teléfono o un ordenador, o incluso publicarlas en redes sociales, si estas se conectasen previamente al espejo.

## 13. Bibliografía

- 1) Alexa installation and documentation (18 de Octubre de 2017).  
Disponible en: <https://github.com/alexa/alexa-avs-sample-app/wiki/Raspberry-Pi> [consulta: 4 de diciembre de 2017]
- 2) PyMySQL installation Python 3 (2016) Disponible en:  
<https://pymysql.readthedocs.io/en/latest/> [consulta: 10 de Diciembre de 2017]
- 3) How to Build Your Own Amazon Echo with a Raspberry Pi (13 de octubre de 2016). Disponible en: <https://lifehacker.com/how-to-build-your-own-amazon-echo-with-a-raspberry-pi-1787726931> [27 de diciembre de 2017]
- 4) Chromium Command Line Switches (23 de diciembre de 2017)  
Disponible en: <https://peter.sh/experiments/chromium-command-line-switches/> [23 de diciembre de 2017]
- 5) MFRC522-python (19 julio 2014) Disponible en:  
<https://github.com/mxgxw/MFRC522-python> [1 de diciembre de 2017]
- 6) MichMich/MagicMirror (18 de Mayo de 2017) Disponible en:  
<https://github.com/MichMich/MagicMirror/tree/master/modules/default/compliments> [29 de noviembre de 2017]
- 7) Tutorial de lector RFID (19 junio 2016) Disponible en:  
[http://www.naylampmechatronics.com/blog/22\\_Tutorial-Lector-RFID-RC522.html](http://www.naylampmechatronics.com/blog/22_Tutorial-Lector-RFID-RC522.html) [20 de noviembre de 2017]