

EMILI

“más allá del espejo”

Autores:

Jesús Ángel Avilés Baena

Alfonso Bautista Núñez

Jesús Calderón Pavón

Salvador Guerrero Jiménez

Rafael López González

Jorge Marrón López

Zaida Martínez Lobato

Ismael Ontanilla González

Elías Rueda Alcázar

Tutor:

Rafael Salvador Jiménez

I.E.S. LA CAMPIÑA



ARAHAL (SEVILLA)

ÁREA: INFORMÁTICA DE SISTEMAS Y SOFTWARE

GROW LAB'19



CEYS Círculo de Economía
y Sociedad
By CESUR



ec2ce
easytosee

1. RESUMEN EJECUTIVO	3
2. OBJETIVOS	5
3. DESARROLLO	6
3.1. Materiales empleados	6
3.2. Presupuesto	8
3.3. Proceso de desarrollo	9
4. COLABORACIÓN: FORMA Y CRITERIO DE SELECCIÓN DE LA TEMÁTICA, SISTEMA DE TRABAJO	16
4.1. Selección de la temática	16
4.2. Sistema de trabajo	16
4.3. Planificación	17
4.4. Colaboraciones	18
5. RESULTADOS Y CONCLUSIONES	19

1. RESUMEN EJECUTIVO

Nuestro proyecto, **EMILI**, "más allá del espejo" ha consistido en el diseño, planificación, construcción, desarrollo e implementación de una interfaz electrónica a modo de espejo. Responde a las siglas:

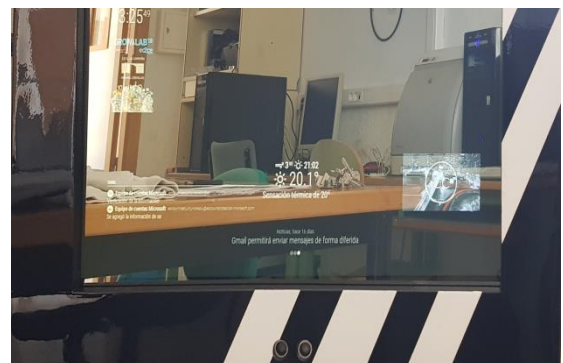
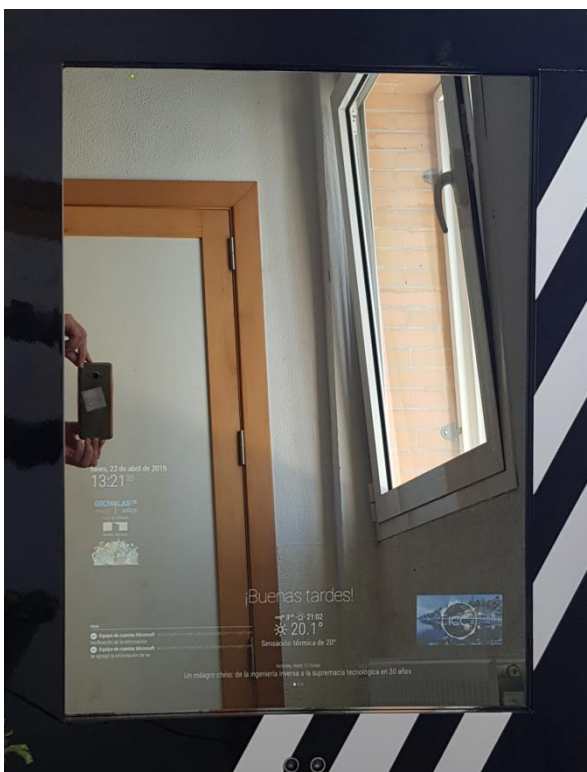
E → Espejo, es obvio, la interfaz completa es un espejo.

M → Multimedia, porque ofrece tanto información textual, como audio y video.

I → Interactivo, nos permite interactuar mediante reconocimiento de voz con comandos propios mediante su micrófono integrado, además de con cualquier dispositivo electrónico conectado: móvil, tablet, ... y por supuesto de forma clásica mediante teclado y ratón, ya sea por cable o directamente por bluetooth.

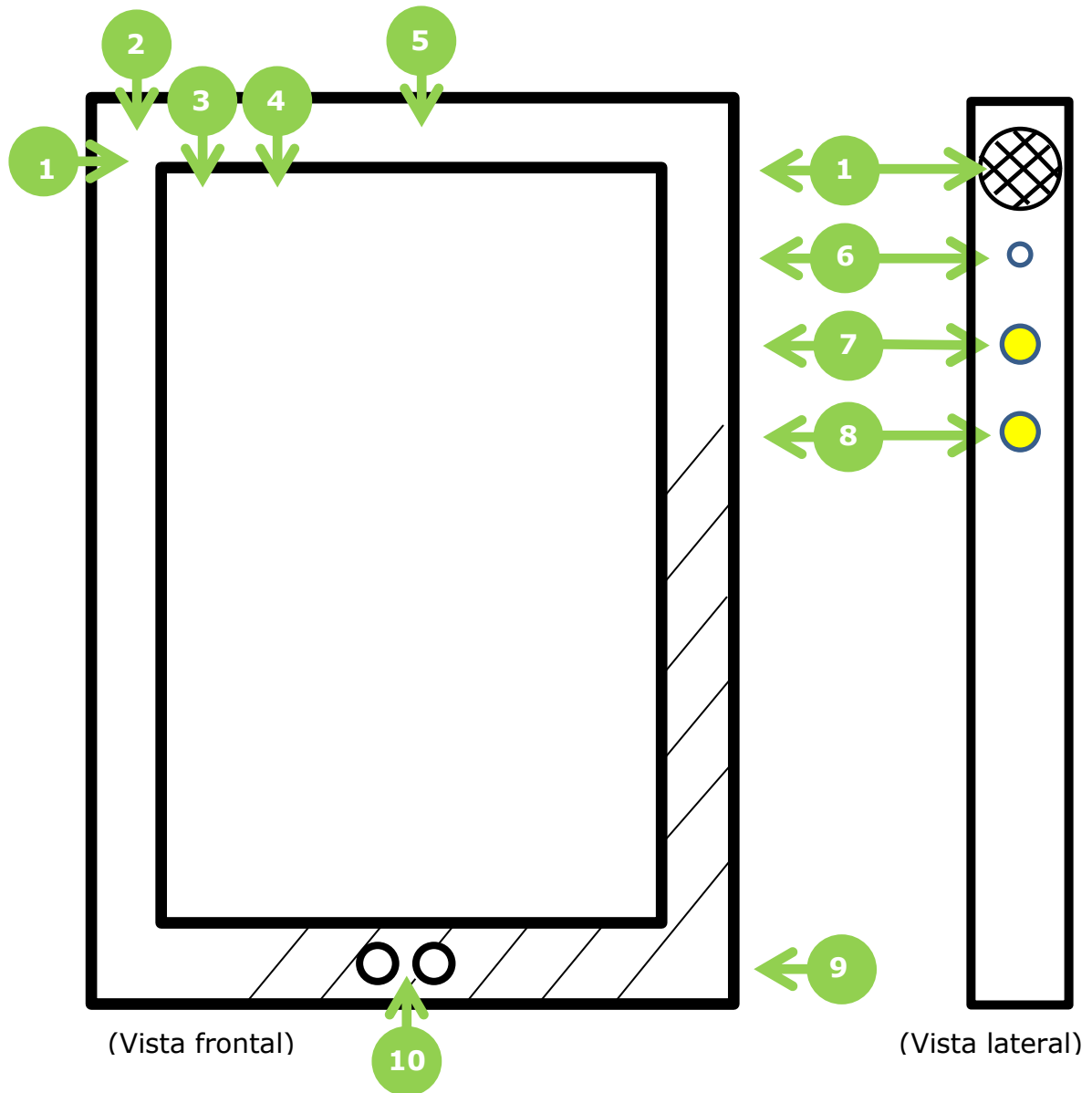
L → Lógico, a saber, es el núcleo básico del mismo, utilizamos en su programación el pensamiento computacional en varios lenguajes: Processing (Arduino), Python, Javascript.

I → Integrado. Bajo una misma estructura encontramos todos los conceptos anteriores que se comunican perfectamente entre ellos.



Detalles EMILI con información on-line actualizada

El siguiente gráfico con las principales funciones de EMILI



CONTROLES

- | | |
|--|--|
| ① Altavoces estéreo | ⑥ Sensor de luz |
| ② Control de Volumen | ⑦ Botón On/Off (pantalla ultrasonidos) |
| ③ Led rojo (OFF) | ⑧ Botón On/Off sonido |
| ④ Led verde ON (pantalla ultrasonidos) | ⑨ Micrófono (reconoc. voz) |
| ⑤ Led color RGB (ó tira LEDS) | ⑩ Sensor ultrasonidos (distancia) |

2. OBJETIVOS

Hemos clasificado los objetivos propuestos al inicio del proyecto en dos grandes grupos:

Objetivos generales

- Diseñar y construir una interfaz multimedia a modo de espejo inteligente y conectado, con información actualizada en tiempo real, personalizada e interactiva.
- Realizar un trabajo colaborativo en equipos de alumnos, con tareas y procesos diferenciados y conseguir integrarlos finalmente en un mismo proyecto.
- Integrar y colaborar entre varios departamentos en un proyecto multidisciplinar.
- Aplicar procedimientos de planificación y organización por parte del profesorado y del alumnado.

Objetivos académicos

- Instalación y configuración de sistema operativo Raspbian Stretch OS sobre tarjeta de memoria microSD en Raspberry Pi 3.
- Instalación y configuración de librería MagicMirror y los diferentes módulos utilizados mediante Python.
- Configuración y programación de interfaz mediante módulos web con Javascript y CSS.
- Instalación, configuración e implementación de reconocimiento de voz mediante Snowboy.

3. DESARROLLO

3.1. Materiales empleados

- Raspberry Pi 3 modelo B

Placa de ordenador de muy bajo coste y tamaño reducido. Se encargará de mostrar en pantalla la información seleccionada y del procesado de voz.



- Arduino uno (o compatible).

Placa electrónica de hardware libre de precio muy reducido que me permite interactuar con el mundo físico mediante sensores y actuadores.



- Placa de prototipado y cables de conexiones para Arduino.

Placa de conexiones para pruebas y conjunto de cables para conectar los diferentes componentes.



- Sensor ultrasonidos

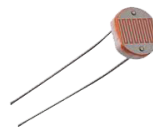
Tipo de sensor que me permite medir distancias. Lo utilizaremos para poder apagar/encender a EMILI dependiendo de si hay o no alguien mirándose frente al espejo.



- 2 Leds (rojo y verde)
Rojo para indicar apagado y verde para encendido. Van integrados directamente sobre el cristal.
- 1 Led RGB (o tira de LEDS)
Tipo de led que permite obtener un rango de colores utilizando diferentes intensidades para el rojo, verde y azul.
Tira de led para aumentar iluminación trasera ("efecto ambilight") con poca luz.



- Resistencia LDR
Tipo de resistencia que nos permite detectar la cantidad de intensidad lumínica que se percibe.



- Pulsadores
Accionarán el encendido de pantalla y altavoces.



- Resistencias distintos valores.



- Relés.

Nos permite conectar y controlar corriente alterna a 220 voltios mediante Arduino con 5 voltios



- Cristal tipo "Reflex"

Es un tipo de cristal (no espejo) que deja pasar menos cantidad de luz que un cristal normal y que dependiendo de la cantidad de luz que haya al otro lado reflejará más o menos y se comportará como un espejo

- Pantalla o monitor reutilizado
- Cable de conexiones USB, HDMI
- Cables de conexión eléctrica y regleta de enchufes

3.2. Presupuesto

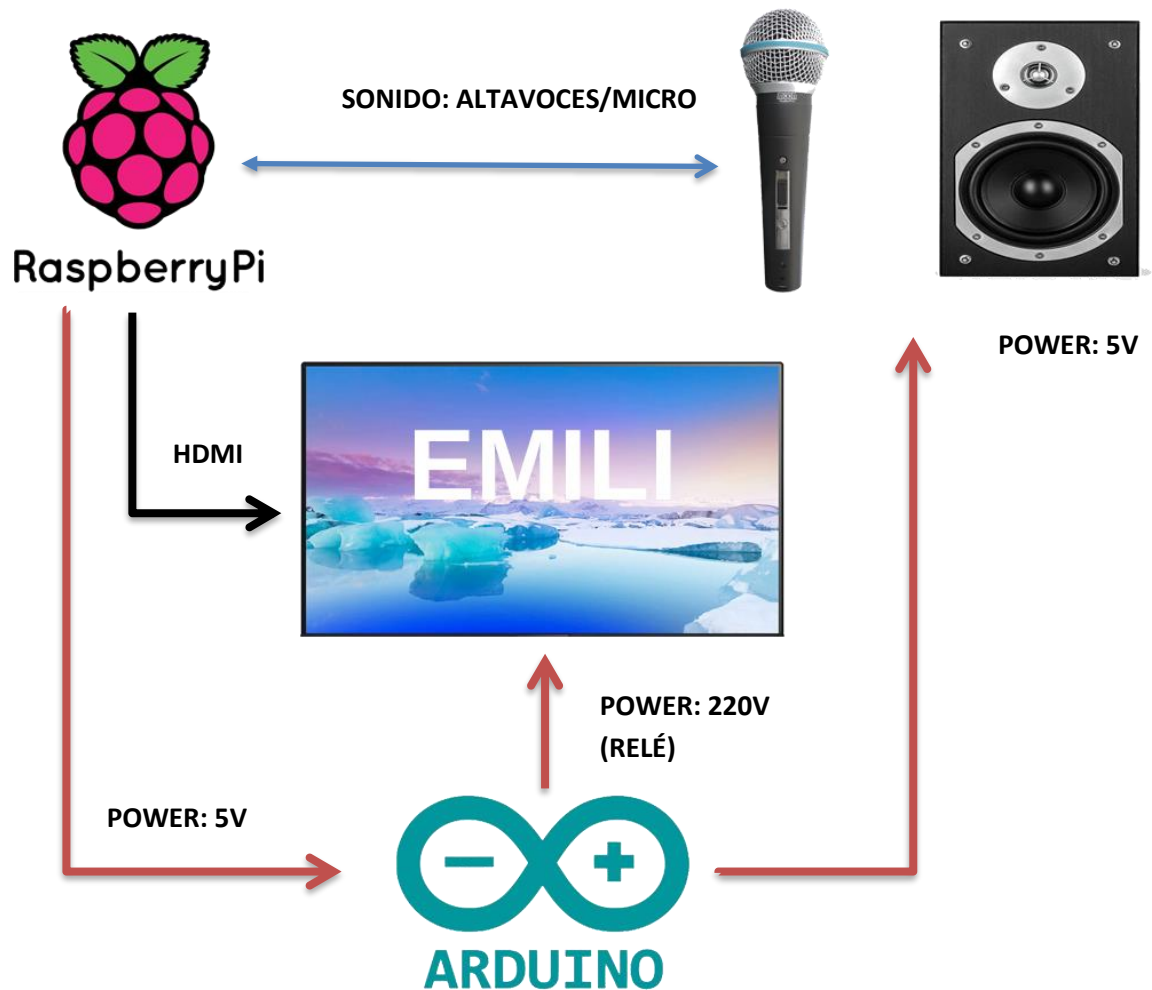
En la medida de lo posible hemos reutilizado elementos de los que ya disponíamos en el centro, tales como las placas Raspberry, Arduino, sensores y actuadores, monitor, etc, siendo el único gasto real de 25 euros, en el que hemos incurrido en el tipo de cristal empleado. Independientemente hemos querido hacer una estimación del gasto estimado si no disponemos de ninguno de los componentes:

- 1 Raspberry Pi 3: 19.47€
- 1 Placa Arduino UNO R3: 2.69€
- 2 Protoboard (400 contactos): 2.30€
- Cristal tipo "reflex" (25 euros)
- Cables (MM - MH): 1.75€ (40 cables)
- 1 Sensor de Ultrasonidos: 2.84€
- 2 Leds: 0.02€
- 1 Led RGB: 0.20€
- 2 Pulsadores: 0.49€
- 2 Relés: 2.30€
- 1 LDR: 0.13€
- 5 Resistencias (220 ohm): 0.14€
- 3 Resistencias (10k ohm): 0.09€
- 1 HDMI: 1€
- 1 Pantalla HDMI: 23€
- 2 Altavoces: 3.99€
- 1 Marco de Madera (Reutilizado)

3.3. Proceso de desarrollo

El trabajo se ha dividido en distintas fases con sus respectivos equipos responsables, de esta forma y atendiendo a los principales campos de actuación, hemos tenido las líneas de desarrollo siguientes:

El esquema general del proyecto obedece al siguiente gráfico:



Por la complejidad del proyecto y la gran cantidad de código desarrollado y ante la imposibilidad de recoger aquí todo el contenido, hemos creado un repositorio abierto en GitHub desde donde puede consultarse y descargarse todo el código utilizado.

REPOSITORIO PROYECTO EMILI: https://github.com/IESLACAMPINA/EMILI_GROWLAB19

A) RASPBERRY

1. Instalación de sistema operativo Raspbian OS

Lo primero a realizar es la instalación del sistema operativo Raspbian OS sobre nuestra Raspberry, para ello hemos utilizado las siguientes descargas, tanto de la imagen, como del software para su grabación

Descarga de sistemas operativo [Raspian Stretch OS](#)

Descarga de software [Win32diskimager](#) para grabar imagen en tarjeta micro SD.

Una vez grabada la tarjeta SD, la insertamos en nuestra placa Raspberry y ya la tenemos totalmente operativa.

2) Seguidamente pasamos a instalar el módulo que nos va a permitir realizar el nuestro proyecto, MagicMirror, una aplicación 100% software libre

```
bash -c "$(curl -L https://raw.githubusercontent.com/MichMich/MagicMirror/master/installers/raspberry.sh)"
```

Una vez instalado encontramos en la dirección /home/pi/Magicmirror/config el archivo config.js, que será el principal archivo de configuración desde dónde tendremos que implantar el resto de módulos.

Los módulos que son las partes visibles de nuestro espejo, son bloques de código javascript, que nos permiten realizar distintas funciones.

En concreto para este proyecto hemos seleccionado los siguientes:

Los módulos que hemos instalados (y su código) son los siguientes:

-Carousel → nos permite mostrar varios módulos en la misma pantalla en distintos intervalos de tiempo.

-Clock → Podemos visualizar la fecha y la hora actual, de manera digital.

-Simple-Logo → Nos muestra el logo oficial de GrowLab'2019 fusionado con el logo del instituto.

-Email → Vemos la bandeja de entrada de nuestro correo a tiempo real.

-Compliments → Genera diferentes frases dependiendo del momento del día en que nos encontremos.

-Newsfeed → Noticias actuales (tecnológicas) extraídas de la página ep00.epimg.net correspondientes a www.elpais.com.

-Calendar → Visualizamos eventos añadidos a nuestro calendario de nuestra cuenta de Google.

-Weather Forecast → Nos muestra el tiempo actual o el que hará en nuestra ciudad en los próximos 6 días.

-Wunderlist → Lista de la compra.

-Remote control → Permite administrar los módulos de EMILI desde cualquier dispositivo autorizado.

- Horario → Muestra un horario semanal por día previamente configurado por nosotros.

B) ARDUINO

Para la implantación de nuestra placa Arduino hemos ido realizando diferentes pruebas con los distintos sensores (ultrasonidos, resistencia lumínica LDR) y actuadores (botones, relés, leds encendido) para comprobar que tanto las conexiones como el software funcionaba correctamente y una vez testeado lo hemos ido integrando sobre la misma placa.

Para el desarrollo hemos utilizado una placa de prototipado, además de la propia placa de Arduino en la que hemos configurado un sensor ultrasonido (sensor de distancias) que enciende la pantalla cuando detecta un objeto a menos de un metro.

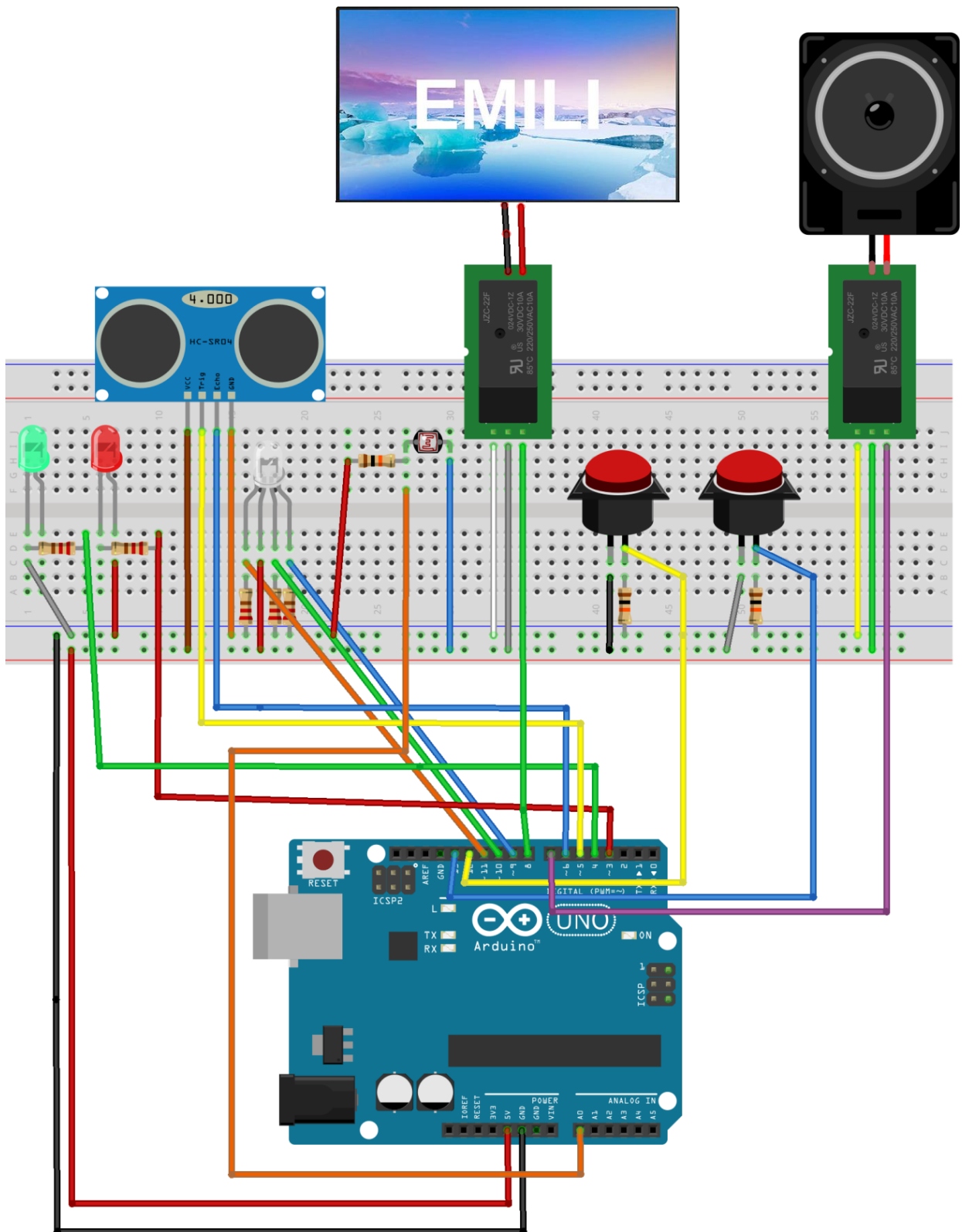
Así mismo hemos configurado un sistema de leds, en el que uno (rojo) está encendido, indicando que el ultrasonido está deshabilitado, por si queremos que no se encienda el sistema si hay alguien cerca. El otro led (amarillo) es su opuesto, se enciende cuando el ultrasonido está habilitado.

Hemos instalado además dos interruptores, uno de ellos hará que los altavoces se enciendan o apaguen, y el otro activa o desactiva el ultrasonido.

Después hemos utilizado un sensor de luz (resistencia LDR) para que cuando no detecte suficiente luminosidad se iluminen un sistema de leds y cuando haya la suficiente se apaguen.

El código Arduino completo, como el resto, podemos descargarlo desde el repositorio indicado [Aquí Código Arduino.](#)

El resultado final puede verse en el siguiente esquema realizado con el programa Fritzing (software libre que se puede descargar [aquí](#))



fritzing

C) RECONOCIMIENTO DE VOZ: SNOWBOY

Para el reconocimiento de voz hemos utilizado Snowboy, un conjunto de herramientas personalizado para la detección de palabras y reconocimiento sonoro que nos permite ejecutar comandos captando nuestra voz.

En primer lugar debemos de cambiar los parámetros del archivo `~/asoundrc` (contenido en [repositorio Github](#) del proyecto), en el cual se encuentran los dispositivos de sonido, el altavoz y el micrófono.

Para asegurarnos de que ambos dispositivos están funcionando tenemos los comandos `alsamixer` y también `aplay -l & arecord -l`

Una vez configurado nuestros dispositivos y de asegurarnos de que funcionan correctamente debemos grabar nuestra voz con los comandos que más tarde queramos que Snowboy reconozca.

En nuestro proyecto hemos utilizado 2 comandos de voz: APÁGATE y ENCIÉNDETE, que harán que la pantalla de nuestro espejo se apague o encienda respectivamente.

Utilizaremos el siguiente comando. Necesitaremos 3 archivos con el mismo contenido. En este caso la orden de voz es "APÁGATE".

```
arecord --format=S16_LE --duration=1 --rate=16000 --file-type=wav  
1.wav
```

```
arecord --format=S16_LE --duration=1 --rate=16000 --file-type=wav  
2.wav
```

```
arecord --format=S16_LE --duration=1 --rate=16000 --file-type=wav  
3.wav
```

Seguidamente utilizaremos el archivo de ejemplo `training_service.py`. Contenido del archivo [training_service.py](#) y ejecutamos los siguiente:

```
python training_service.py 1.wav 2.wav 3.wav apagate.pmdl
```

Donde `1.wav`, `2.wav`, `3.wav` son los audios que habíamos creado anteriormente y `apagate.pmdl` es el archivo que se va a crear automáticamente al ejecutar lo anterior.

Repetiremos el mismo proceso pero ahora para el comando "enciéndete", y así tendremos los dos archivos, `apagate.pmdl` y `enciendete.pmdl`. Una vez creado estos dos archivos utilizaremos el archivo que reconocerá nuestras órdenes que lo llamaremos `demo2.py`, cuyo contenido podemos ver en nuestro repositorio, [demo2.py](#) en GitHub.

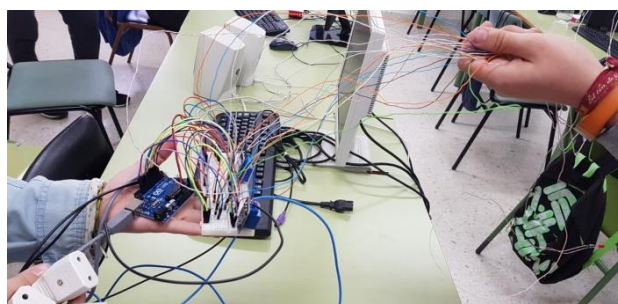
Finalmente, una vez creado este archivo usamos el siguiente comando que detectará nuestra voz y dependiendo de la orden apagará o encenderá la pantalla (en su caso el comando que hayamos predefinido).

```
python demo2.py apagate.pmdl enciendete.pmdl
```

D) ESTRUCTURA E INTEGRACIÓN COMPONENTES

En paralelo a la instalación y configuración de los distintos componentes de nuestro proyecto, hemos ido realizando la estructura física que va a servir de soporte a todo el proyecto. Para su realización hemos utilizado en su mayoría materiales reusados, en concreto, la estructura está formada por marcos de madera de puertas del propio instituto (renovadas o suprimidas por ampliación-reducción de aulas).

Incluimos aquí algunas imágenes del proceso de creación de la estructura e integración de los distintos componentes dentro del chasis.



E) TOQUES FINALES. MEJORAS INCLUIDAS.

Para agilizar y automatizar todo el proceso, tanto de reconocimiento de voz, como de nuestro software para EMILI, y que funcione correctamente sin necesidad de intervención del usuario, hemos añadido varios scripts de inicio:

Para el reconocimiento de voz

- Script que sustituye el archivo de configuración `.asoundrc` por defecto, se dirige a la carpeta de snowboy y arranca el servicio de reconocimiento de voz.

```
#!/bin/sh
```

```
sudo rm /home/pi/.asoundrc
```

```
sudo cp /home/pi/.asoundrcseg /home/pi/.asoundrc
```

```
cd /home/pi/snowboy
```

```
python demo2.py apagate.pmdl enciendete.pmdl
```

- Iniciar reconocimiento de voz (Snowboy) al arrancar la máquina:

Editamos el archivo `/etc/xdg/lxsession/LXDE-pi/autostart` y añadimos

```
@point-rpi
```

```
x-terminal-emulator -e /etc/init.d/arrancarsb
```

Para EMILI

- Iniciar EMILI al arrancar la máquina automáticamente

```
crontab -e
```

Dentro del archivo añadimos la siguiente línea:

```
@reboot ( /etc/init.d/arrancarmm )
```

Donde, el contenido del script `arrancarmm` es:

```
cd /home/pi/MagicMirror
```


```
npm start
```

4.1. Selección de la temática

Al inicio, se proponen varias ideas de proyecto a realizar. En grupo se discuten las distintas alternativas a modo de Brainstorming. Finalmente por originalidad y viabilidad se elige por mayoría absoluta este proyecto.


4.2. Sistema de trabajo

Para realizar EMILI nos hemos dividido en los siguientes grupos,




[RASPBERRY TEAM]

Instalar y configurar Raspbian OS sobre una RaspBerry Pi Model 3, junto con la librería MagicMirror y una serie de módulos. Estos permiten a EMILI realizar unas funciones específicas




[ARDUINO TEAM]

Realizar tanto la parte de programación como de conexionado hardware de Arduino y los respectivos sensores y actuadores.



[SNOWBOY TEAM]

Configurar la aplicación de reconocimiento de voz "SnowBoy" que permite reconocer nuestra voz mediante un micrófono, analizarla y realizar aquella función que le hayamos indicado. Interactuar cómodamente y de manera sencilla con EMILI sin la necesidad de teclas o pantalla táctil.



[BRICO TEAM]

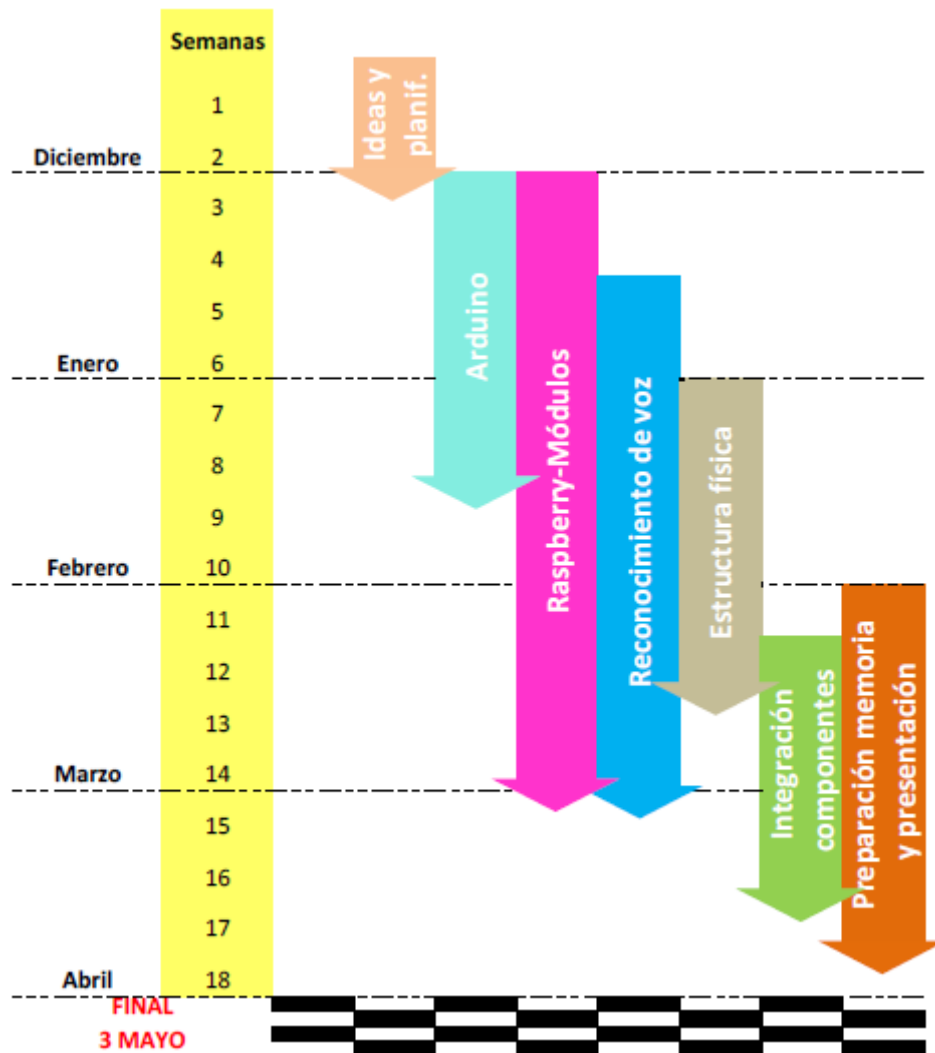
Creador de la estructura física del espejo y acabado final e integración de los distintos componentes.

4.3. Planificación

El desarrollo del proyecto se ha llevado a cabo en varias líneas convergentes:

1. Idea, selección del proyecto a realizar y planificación
2. Desarrollo Arduino
3. Desarrollo Raspberry
4. Reconocimiento de voz
5. Fabricación de la estructura física
6. Integración de componentes
7. Preparación de la memoria y presentación del proyecto

Como guía de seguimiento y planificación mostramos el siguiente esquema:



4.4. Colaboraciones

Además del trabajo realizado por los distintos equipos de alumnos, hemos contado con las colaboraciones y apoyos tanto personal como material de los siguientes dptos:

- Dpto. de informática al completo, así como materiales necesarios, espacios, horas lectivas, extraescolares, etc.
- Colaboración del dpto. de automoción (electromecánica y carrocería), para el pintado y esmaltado final de la estructura del espejo.
- Colaboración del dpto. de tecnología, que ha puesto a nuestra disposición las distintas herramientas y materiales para la fabricación y acabado de la estructura.
- Asesoramiento del dpto. de Lengua para la finalización de esta memoria y orientaciones para una correcta exposición oral final.

5. RESULTADOS Y CONCLUSIONES

Atendiendo a los resultados obtenidos tras el proceso de desarrollo del proyecto, hemos de extraer las siguientes conclusiones:

1. Desde el principio teníamos claro que queríamos realizar todo el proceso, idea, planificación, construcción, prueba y perfeccionamiento del proyecto. De hecho el resultado final ha superado incluso nuestras expectativas.
2. Originalidad. Hemos desarrollado una idea a partir de una implementación original y la hemos mejorado e integrado con otras soluciones, como el control por voz, integración con Arduino y múltiples posibilidades de ampliación.
3. Hemos partido de un objeto común, un espejo, presente en todos los hogares, comercios, etc. lo hemos transformado en un objeto del internet de las cosas (IOT), que es a su vez un centro multimedia, con información actualizada y personalizada que es capaz de interactuar con el usuario.
4. Reutilización de materiales. Reciclaje y reutilización de materiales presentes en el centro.
 - La estructura general de madera está formada por marcos de puertas del centro reutilizados.
 - Monitor principal reutilizado
 - Arduino y Raspberry ya presentes en el centro
 - Altavoces reutilizados de otros que no estaban en uso.
5. Utilización de hardware libre. Para el proyecto hemos utilizado hardware libre, una placa Arduino y una Raspberry pi, ambas de precio muy reducido y con múltiples aplicaciones disponibles y reutilizables para otros proyectos
6. Utilización de software libre Todo el software utilizado se encuentra bajo la licencia GPL y es de dominio público. Además, todo el código que hemos generado, tanto Raspberry como de Arduino, se suministra bajo la misma licencia, de forma abierta y gratuita.
7. Todo el código que hemos generado, mejoras, configuraciones, etc. está documentado en un repositorio de GitHub.com en internet y accesible de forma gratuita y libre en la dirección https://github.com/IESLACAMPINA/EMILI_GROWLAB19
8. Múltiples posibilidades. Un presupuesto muy ajustado que ofrece un abanico de posibilidades inmenso, tanto en el ámbito personal, empresarial, institucional, "maker". Consideramos el tamaño elegido para el prototipo como ideal, pero lo verdaderamente importante es que la ampliación del mismo no implica un aumento del presupuesto

puesto que lo único que aumentaría de tamaño es el propio cristal, quedando el resto de componentes inalterados.

Desde el IES "La Campiña", queremos agradecer a EC2CE y CESUR por la organización de esta edición de GrowLab y por haber contado con nosotros para la misma.

Para nosotros ha supuesto un esfuerzo importante, de horas tanto lectivas como extraescolares, que ha conllevado integrar muchos de los contenidos de materias estudiadas además de otras ampliaciones adicionales inicialmente no previstas, pero podemos concluir que visto el proyecto final construido y el trabajo e implicación de los equipos, dicho esfuerzo ha merecido la pena.