

Carpeta de Campo

Semana 15/03 - 22/03

Escribimos una declaración jurada sobre la compra del NeuroSky Mindwave debido a que Matías Pierri tuvo la oportunidad de que un conocido se lo compre en USA Y abaratar costos

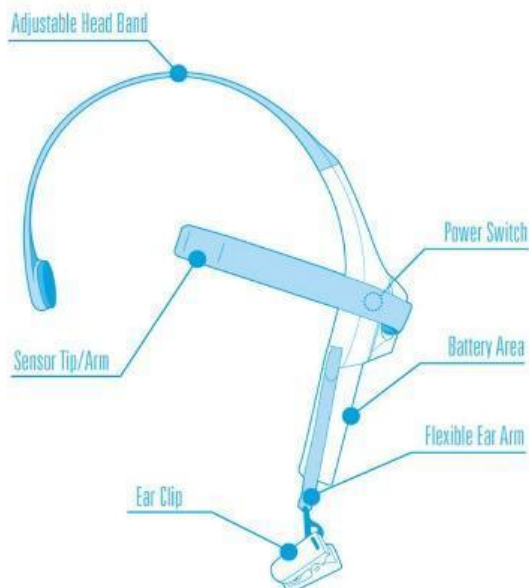
Clic [aquí](#) para ver la Declaración Jurada

NeuroSky Mindwave

Es una diadema con biosensores de electroencefalografía que permite hacer la medición de las ondas eléctricas del cerebro y vincularse a diferentes aplicaciones para investigación, entrenamiento cognitivo y bienestar.

El NeuroSky Mindwave Mobile 2 mide y emite de forma segura los espectros de potencia EEG (ondas alfa, ondas beta, etc.), los medidores NeuroSky eSense (atención y meditación) y los sensores para el parpadeo de los ojos. El dispositivo consta de un auricular, una orejera y un brazo sensor, los electrodos de referencia y de tierra del auricular están en la pinza para la oreja y el electrodo EEG está en el brazo del sensor, apoyado en la frente sobre el ojo.

En nuestro caso, lo vamos a usar para que cuando la persona parpadee se active el sistema.



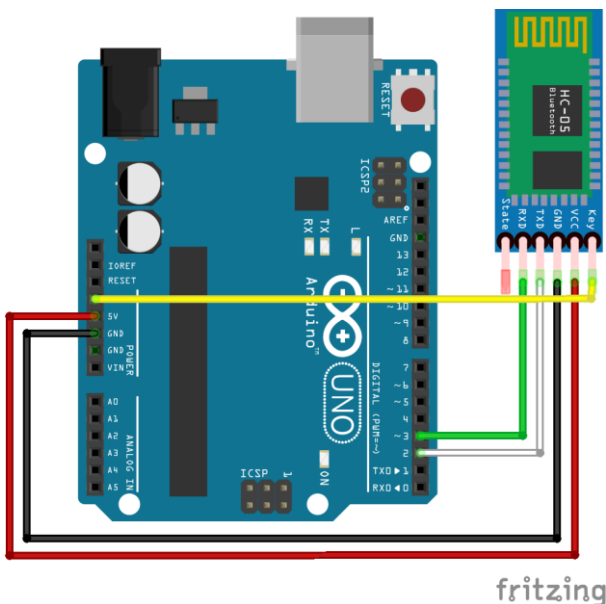
Motor vibrador

Investigamos sobre vibradores y llegamos a la conclusión de que lo más viable es reciclar vibradores de teléfonos viejos como los Nokia. Un motor vibrador es un motor común en el cual se le pone un peso en la punta del émbolo lo que lo hace vibrar, además se coloca dentro de un contenedor que aumenta la vibración.



Conexión entre BT y Arduino

Establecimos el entorno de trabajo BT con Arduino usando los HC-05. En base a las conclusiones confeccionamos una guía de cómo usar dicho entorno



[Guía de como Establecer conexión entre BT](#)

Usando Arduino logramos hacer que funcionen los vibradores extraídos del Nokia usando la función `Tone()` y `noTone()` que te permite poner la frecuencia de vibración la cual va a variar según lo que se necesite

[Guía de Sistema de vibración](#)

Como la Raspberry se maneja con JSON estuvimos aprendiendo a como procesar un string JSON en Arduino y poder hacer el intercambio de datos entre Raspberry y Arduino por BT. Nos dimos cuenta que para que se lean bien los datos hay que usar `Serial.readString()` ya que `Serial.read()` solo detecta que entra un char

JSON (acrónimo de JavaScript Object Notation, «notación de objeto de JavaScript») es un formato de texto sencillo para el intercambio de datos.

Empezamos a trabajar con el sensor cerebral conectado a un teléfono y logramos que cuando parpadee el usuario, el teléfono lo detecte, pero esto es una app y no nos permite tocarla mucho para usarla en nuestro Arduino, pero era a modo de primeros pasos. Usando un módulo bluetooth y comandándolo por AT tratamos de configurarlo insertando los siguientes datos

Baudrate = "576000"

BIND ADDR= "0081, f9,29eb31"

PSSW = "1234"

Con esta configuración no funciona ya que la password estaba mal puesta. Viendo la documentación nos dimos cuenta que la pass es 0000

PSSW CORRECTA ="0000"

Ya de esta forma pudimos conectar el Arduino con el HC-05 pero con el example que nos da el fabricante solo podemos medir nivel de atención y relajación. Tenemos que investigar como medir nivel de pestañeo. Sin embargo, hicimos una prueba donde se prenden leds de acuerdo a que tan concentrado está el usuario

Establecimos el entorno de trabajo en Google Visión IA que es usar la Nube de Google para, a través de un sistema de redes neuronales, obtener un análisis de lo que se ve en una foto. Para esto tuvimos que crear un proyecto en Google Cloud Services y habilitar la api de Google Visión AI. Una vez habilitado esto nos dan un archivo JSON que tiene nuestra API Key y la insertamos en la Raspberry usando

```
export GOOGLE_APPLICATION_CREDENTIALS=/home/pi/filename.json
```

Una vez seteada la Raspberry se usan los siguientes comandos para instalar las dependencias necesarias

```
python3 -m pip install --user pip
```

```
python3 -m pip install --user google-cloud-vision
```

```
python3 -m pip install --user Pillow python3 -m pip install --user picamera
```

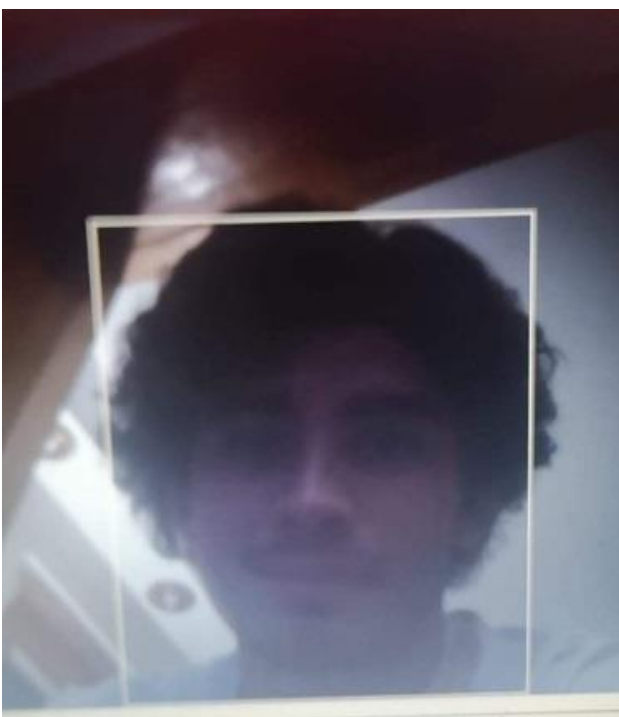
Usamos este comando para sacar una foto

```
raspistill -o cam.jpg
```

Y este ejemplo para poder probar el análisis de la foto

```
python3 camera_vision_face.py
```

Nos devuelve este análisis donde la Raspberry se encargó de detectar las personas que aparecen en la foto en este caso Theo. Lo borroso de la foto se debe a que Theo tiene el celu que no enfoca, la definición en sí de la Pi Camera es bastante buena



Ahora tenemos que ver cómo usar la Raspberry para conectarla al BT y además que todos estos programas que describimos antes estén funcionando en Loop y de forma automática cuando se prende la Raspberry.

Semana 22/03 - 29/03

Estuvimos viendo documentaciones y aparentemente para poder leer el valor de Blink Strength tenemos que escribir en el registro **0x16**

TABLA DE CÓDIGOS			
EXCODE	CODE	VLENGTH	VALUE
-	0x02	-	POOR_SIGNAL Quality (0 a 255)
-	0x04	-	ATTENTION eSense (0 a 100)
-	0x05	-	MEDITATION eSense (0 a 100)
-	0x16	-	Blink Strength (0 a 255)
-	0x80	2	RAW Wave Value (-32768 a 32767)
-	0x83	24	ASIC_EEG_POWER
-	0x55	-	[EXCODE]
-	0xAA	-	[SYNC]

Tabla 5.1 Tabla de códigos del chip ThinkGear

Se hicieron ejemplos con Arduino para empezar a leer los datos Json que se recibirán de la Raspberry. Se han hecho pruebas de como formar un Json y como obtener el valor de Json. Luego se probó mandar archivos String por BT, pero nos dimos cuenta que la función no permite mandar Strings por BT:

```
BTMaster.write()
```

Se ha visto varios videos o formas de intentar mandar el String por BT de otra forma, pero no se ha conseguido, entonces se empezó a investigar para pasar un String a un char Array. Se ha encontrado un ejemplo en el que muestra que la función:

```
str.toCharArray(charArray,buf)
```

Siendo str el String ingresado, charArray, la variable donde se guarda el charArray y buf la cantidad máxima a convertir de String en char Array.

Se ha tenido problemas con esta función ya que dependiendo del orden en el que se declaraba el String, funcionaba o no. Pero se ha logrado descubrir el error, creando 2 programas:

- Uno que simula el parpadeo, sacar foto y mandar el String Json por BT a través de un charArray
- El otro que recibe el char Array y lo convierte en un String, y obtiene el valor del Json de color y de acuerdo al color realiza una vibración

Sistema de vibraciones con Json y BT

Hemos realizado el logo para el proyecto:



Semana 29/03 al 05/04

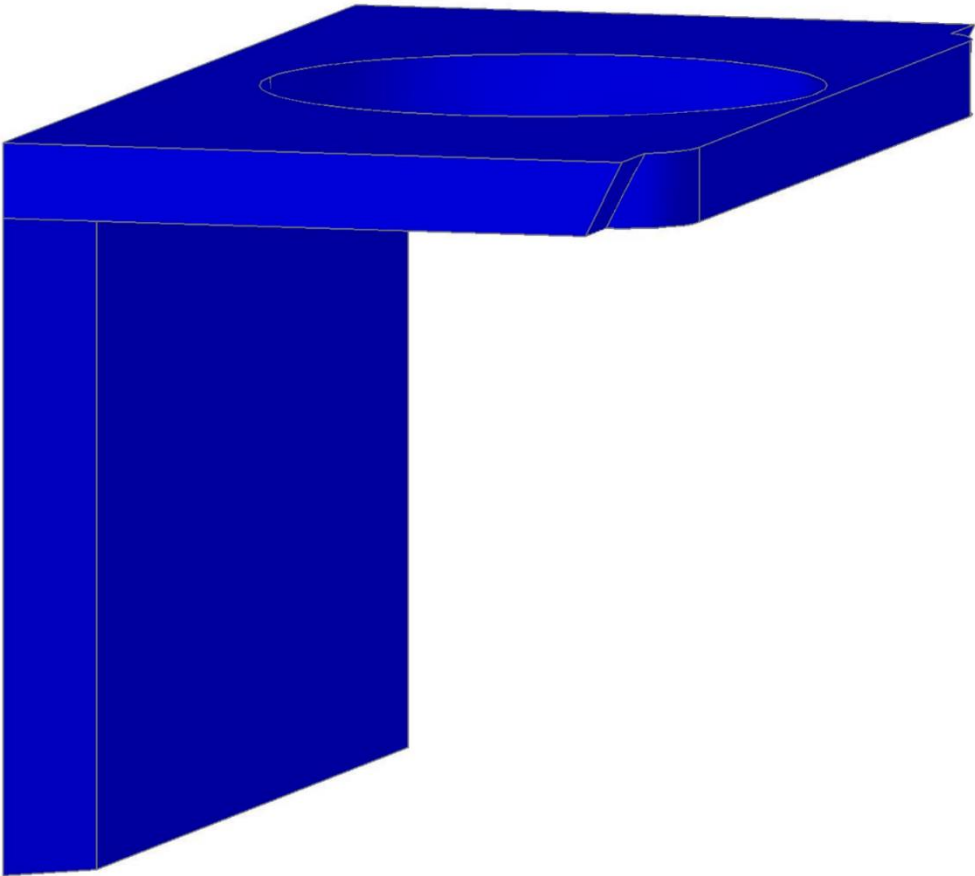
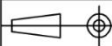
Se crearon las piezas en 3D del cargador de pilas. Las piezas están diseñadas teniendo en cuenta las plaquetas que van a ir colocadas y están los agujeros para los respectivos cables.

Cargador parte 1

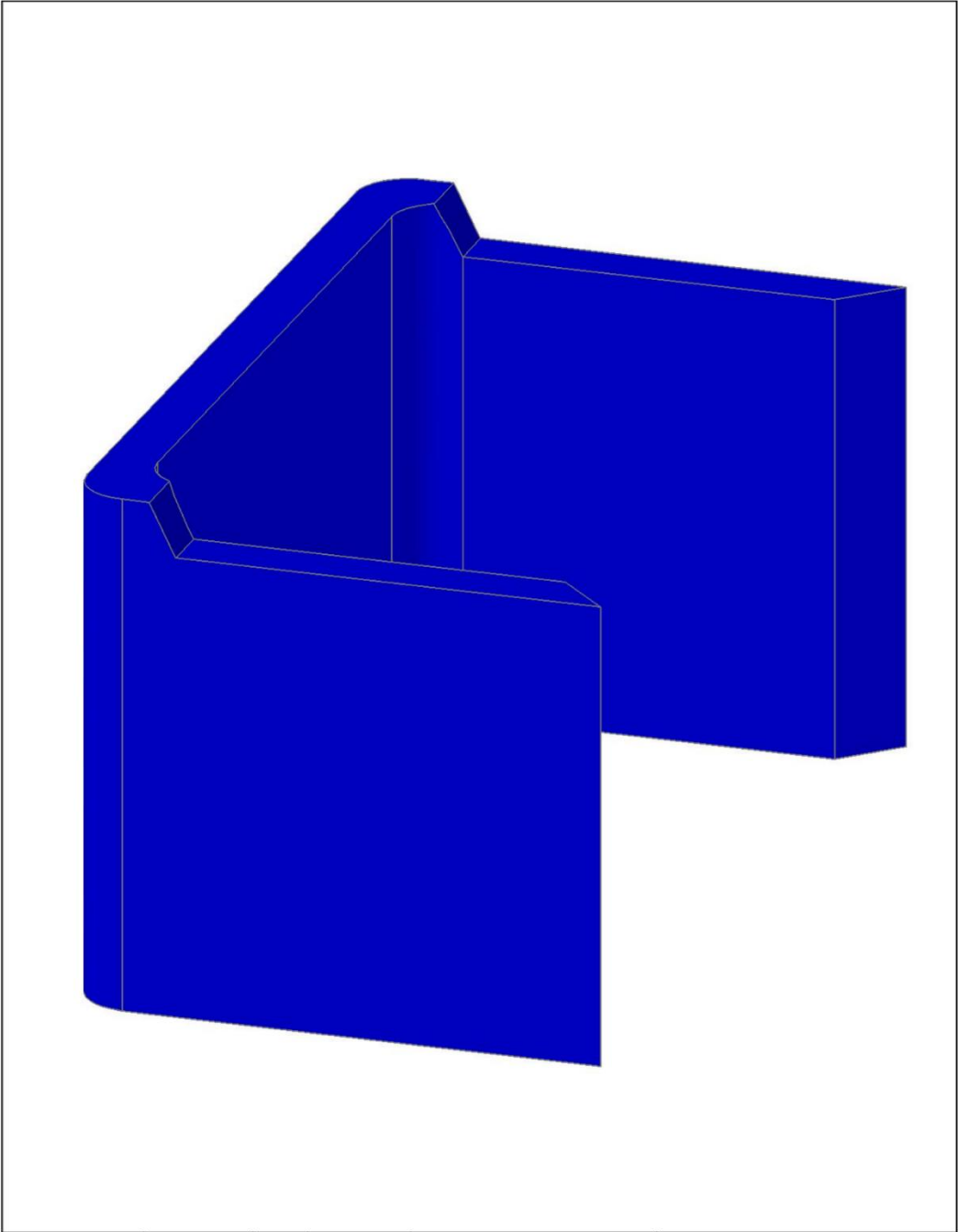


		Fecha	Nombre	EESTN° 7	TRQ
	Dib.	29/03	Fonte G.		
	Rev.	30/03	Pierri M.		
	Apr.		Medina S.		
	Esc. 2:1	Cargador de Pilas de Litio 186500			División: 7°2° Aviónica
					Plano n°: 1
	Toler. Rug.				Reemp:

Cargador parte 2

					
		Fecha	Nombre	EESTN° 7	TRQ
	Dib.	29/03	Fonte G.		
	Rev.	30/03	Pierri M.		
	Apr.		Medina S.		
	Esc.	Cargador de Pilas de Litio 186500			División:
	2:1				7°2° Aviónica
					Plano n°:
Toler.				1	
Rug.				Reemp:	

Cargador parte 3



		Fecha	Nombre	EESTN° 7	TRQ
	Dib.	29/03	Fonte G.		
	Rev.	30/03	Pierri M.		
	Apr.		Medina S.		
	Esc. 2:1	Cargador de Pilas de Litio 186500			División: 7°2° Aviónica
	Plano n°: 1				
Toler. Rug.	Reemp:				

Analizamos el problema que creamos en el repositorio de Ojo de Van Gogh ya que nos mezclamos con el manejo de branches y mil problemas entonces decidimos crear repositorios por cada área de trabajo del proyecto y los archivos de documentación y publicidad manejarnos por drive. Los repos quedan de la siguiente manera:

- Mindwave
- ArduinoVibrador
Raspberry
- Cargador de pilas

Ahora nos manejamos con ramas donde la versión final y estable del repo va a Máster y luego cada feature se desarrolla en una rama paralela. Luego de hacerle el testing a la misma y ver que no haya conflicto se hace un merge (se fusiona la rama con el máster y se cierra la rama)

Con respecto al mindwave decidimos poner en pausa el código del micro porque nos estábamos perdiendo con el tema del payload. Encontramos varios repos en GitHub con los códigos en Python donde al ejecutarlos te otorgan los datos. El plan es conectar por BT el mindwave a la computadora y correr el script. Una vez que el mismo esté listo

Nos estamos contactando con varias empresas preguntando sobre plaquetas flexibles para el sistema de vibración, aunque también se está evaluando la opción de utilizar los brazaletes porta celulares para correr. Se ha realizado el PCB del sistema ya que una empresa nos contactó y nos lo pidió para saber el tamaño y complejidad.

Semana 05/04 al 12/04

Hemos recibido las respuestas de las empresas por plaquetas flexibles y nos ha parecido muy elevados los costos por lo cual optamos por usar un brazalete para celulares de un tamaño de 18cm x 10 cm y se ha realizado el esquemático del circuito donde estará la Raspberry con la cámara, el Arduino Nano, el Bluetooth, dos pilas, un regulador y los vibradores.

Hemos calculado el consumo total del circuito para saber que tenemos que usar dos pilas de 3.7V ya que con una nos duraría poco tiempo, y por esto se ha investigado sobre Step Down, ya que tiene mejor rendimiento que un regulador, ya que se necesitan 5V de entrada en vez de los 7.4V que nos dan las dos pilas en serie.

También se ha realizado una librería para los colores en Arduino, y hemos encontrado el problema que si haces un archivo con la extensión: ".c " salta un error y se solucionó esto cambiando la extensión del archivo a ".cpp"

Se ha investigado sobre el mindwave y el funcionamiento de un programa y se ha conseguido que se detecte el parpadeo, pero también se detecta el parpadeo no forzado y para eso estamos viendo cómo solucionarlo.

Se ha conseguido que la Raspberry nos entregue el color en RGB con Json

Se ha comprado un dominio y estamos haciendo la página web a través de un localhost (XAMPP) y usando WordPress. Se ha elegido el tema de la página que es BlackootLite.

Semana 12/04 al 19/04

Se ha terminado el PCB de la placa

Montamos el host en 000webhost que nos ofrece 300 Mb de espacio y le instalamos WordPress. Tuvimos problemas con el dominio así que contactamos a soporte técnico de Hostinger y nos solucionaron el problema.

Clic [aquí](#) para entrar a nuestra pagina

Encontramos un módulo de Python que pasa de RGB a nombre de color y funciona, pero tienen sus limitaciones. El tema es que no da resultado a todos los RGB sino a los que son fáciles tales como 0,0,0 que da black, pero si le metemos algo como 65,23,19 ya se pierde.

Encontramos una alternativa basándonos en la página de un sensor de color que nos otorga Luis llamas

[Sensor para Detectar colores](#)

Acá muestra un sensor para detectar colores y en esta usan una librería que pasa los valores RGB a HSV que nos permite saber las distintas tonalidades de un color ya que distingue de rojo oscuro y rojo claro y nos permite tomar los dos como rojo. Esta librería la pasamos a Python y gracias a esto ya poseemos 12 colores identificables

Hemos realizado la librería para la vibración de los colores y para esto hemos creado una tabla para determinar la vibración de cada color:

Lista de colores para vibración

- El número 0 representa 0.25 segundos de duración de vibración
- El número 1 representa 1 segundo de duración de vibración
- Entre cada número hay 0.25 segundos de pausa

Código de Vibración	Colores
0000	Blanco
0001	Rojo
0010	Marrón
0011	Naranja
0100	Amarillo
0101	Verde
0110	Cian
0111	Azul
1000	Violeta
1001	Rosa
1010	Gris
1011	-
1100	-
1101	-
1110	-
1111	Negro

La página web se nos cayó. Nos suspendieron la página porque aparentemente superamos el límite de visitas al entrar a verla y nos dicen que si no pagamos en 7 días nos la borran.

Nos comunicamos con webhost a ver si hay alguna alternativa, pero no nos contestan así que no nos queda otra que pagar y mudarnos a Hostinger

Logo 3D

Se ha hecho el logo en formato 3D para después agregarlo como marca en la caja que tengamos.



Semana 19/04 al 25/04

Pagamos la migración de nuestra página a Hostinger que nos salió 10 dólares. Tenemos que esperar 48 hs a que nos digan si salió todo bien y no perdimos la página.

Recuperamos la página y esta todo en orden, aunque se nos desconfiguró el dominio y tenemos que cambiarlo así responde a ojodevangogh.tech . Eso sí, no podemos entrar al panel de WordPress por alguna razón.

Nos comunicamos con Hostinger y después de unos largos idas y vueltas de email mudamos completamente la página ya no tenemos nada en webhost está todo en Hostinger y funciona todo. Habilitamos la certificación SSL así que nuestra página es segura y ya tenemos acceso al panel de WordPress.

Tenemos un problema en el responsive desing que hace que desaparezcan los botones de la página cuando usamos una versión mobile y no sabemos que hacer

Semana 25/04 al 3/5

Vimos que no había manera de solucionar el responsive design más que pagar el upgrade del theme de WordPress a premium que son 40 dólares así que decidimos cambiar de tema de WordPress.

Empezamos la búsqueda y encontramos uno llamado Astrid que está muy bueno visualmente y tiene mucha opción de personalización. Lastimosamente no podíamos cambiarle el fondo blanco así que para que no queme tanto le colocamos letra Arial que es más gruesa y llama la atención.

Le instalamos un plugin a la página para que aparezcan botones que vinculen la página con Instagram, LinkedIn llamado UltimateShare.

Creamos una página de Instagram para empezar a subir cosas y ya grabamos unos videos de como la Raspberry reconoce colores y como a través de un color seleccionado funcionan los vibradores

Para un mejor orden creamos un repositorio nuevo donde metimos todas las cosas de los pequeños repos a través de git flow que permite ordenar todo en feature.

Para armar cosas lindas para redes sociales hicimos que la plaqueta tenga todos los micros y cosas con modelos 3D así verlos con el visor 3D de KiCAD

Semana 03/05 al 11/05

Se hizo una introducción para el video en After Fx, se le puso música de fondo y se subió a Instagram y a la página web el primer video mostrando como se reconoce el color con la cámara de la Raspberry

También se definió el modelo para hacer la tapa del cargador de pila y se empezó a hacerla.

Se agregó un plugin para que la pagina cargue más rápido y se corrigió el logo para que no sea de fondo blanco y pueda verse.

Semana 11/05 al 18/05

Se empezó a intentar conectar la Raspberry con el sensor cerebral y se configuro esta para que trabaje por Remote

Agregamos con gtts para que se pueda escuchar el color y lo probamos con un parlante y anduvo. Para traducir el texto color a voz se creó una nueva función que envía archivo txt que los recibe el gtts y los reproduce en un parlante

Para la PRIMER ENTREVISTA DE RADIO, se realizó una tarjeta publicitaria que se subió a Instagram y como se escucho bajo, se editó el audio grabado y se preparó para subirlo a IGTV y quedamos en que cuando la situación del país lo permita, nos volveremos a juntar para contar los nuevos avances.

Se dañó el disco de la computadora y perdimos lo que teníamos hecho de la tapa y no se había subido a git.

Semana 18/05 al 25/05

Se pudo conectar el sensor cerebral con la Raspberry, pero no lee el payload del Mindwave y estamos con ese problema que no sabemos cómo resolverlo

Se hizo un Backup de la página web porque se nos vencía el mes de Hostinger, pero debido a un error que tuvieron nos regalaron un mes y ya elegimos que para el mes próximo vamos a cambiar al host de Sitio Hispano que nos saldría \$860 el año ya que no se puede pagar por mes

Estuvimos pensando y se nos ocurrió que nuestra página tiene que tener para que los ciegos la puedan leer, entonces agregamos el Plugin Responsive Voice que permite a través de un botón reproducir el contenido de cada página.

Se editó y subió el IGTV de la nota de radio y también subimos el link de la página que hizo la radio FM106.5 sobre esta nota.

Se terminó la tapa y se empezó a realizar el encastre en la caja.

Se crearon archivos txt para el gtts para optimizar los tiempos de respuesta cuando se reproducen los colores, ya que lee los archivos creados en vez de crearlos y después leerlo.

Semana 25/05 al 1/06

Se logró la comunicación entre el Arduino y la Raspberry y creamos 2 librerías una para detectar colores y otra para crear un JSON. Esto se hizo para ordenar el código. A la librería de detectar colores también se le incorporó los audios para reproducir el color.

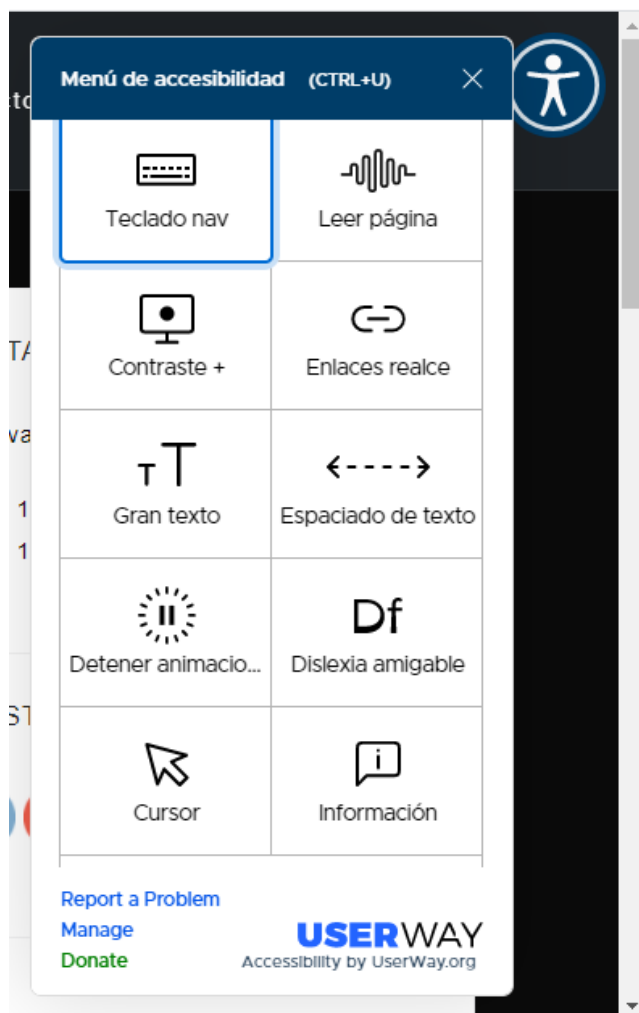
Se investigó como hacer que cada vez que la Raspberry se enciende empiece a correr el script automáticamente sin ponerle el comando utilizando Crontab.

[Script para automatizar inicio](#)

Se ha logrado pasar las fotos a formato base64 para reducir tiempos de espera y se logró una optimización. Antes tardaba 24 segs y ahora tarda entre 16 y 20 segs, siendo un factor clave la luz ambiente, ya que si hay poca luz tarda más.

Se ha investigado e implementado en la página web sobre plugins de accesibilidad, se han probado varios pero el que más herramientas nos proporciona es UserWay, Este plugin tiene varias herramientas

- Permite 3 niveles de lector de pantalla (medio, rápido y despacio).
- Navegar por teclado (accesibilidad para ciego y poca motriz)
- Cambiar los contrastes de fondo (gris, invertido, oscuro y claro)
- Permite resaltar enlaces
- Aumentar el tamaño y el espaciado del texto y también del cursor
- Detener animaciones
- Cambiar la letra para la gente que padece dislexia
- Aumentar el cursor y tener una guía de lectura
- Y por último ofrece resetear todas las opciones



Se ha avanzado con el modelado 3D de la caja que contiene el proyecto, modificando los espacios para las pilas ya que no iban a entrar con el que tenía y se avanzó con el encastre de la tapa con la caja agregando los agujeros en la misma.

Semana del 1/06 al 8/06

Se editó y musicalizó un segundo video para subir a las redes, el cual mostrara como al reconocer un color empezaría a reproducirse un sonido/vibración.

Se subió a ig historias de quienes somos y una breve descripción del proyecto, dejando estas mismas como destacadas y se aumentó 16 seguidores.

Se hicieron varias pruebas para empezar a reconocer texto y con el código que ofrece la api no lográbamos poder reproducir el texto, entonces hemos buscado otro código el cual nos permitió reproducir el texto que reconoció de la imagen que le cargamos a la Raspberry

Se han buscado diferentes librerías para hacer una lectura de OCR más rápida pero no se encontró, y viendo videos y características, hemos decidido pasar de la Raspberry Pi Zero w a la Raspberry Pi 3 y hemos tenido las siguientes mejoras:

- Reconocimiento de colores en 5 segs
- Reconocimiento de texto entre 7 y 11 segs
- Reconocimiento de objetos en 10 segs

Se hizo un código para reconocer objetos obteniendo el objeto con mayor seguridad. Se ha agregado para que el objeto sea traducido ya que la librería da el resultado en inglés. Hemos aprovechado esto para darle una herramienta más al proyecto y permitir que cuando una persona quiera leer texto que no sea español, sea traducido y se reproduzca por el auricular.

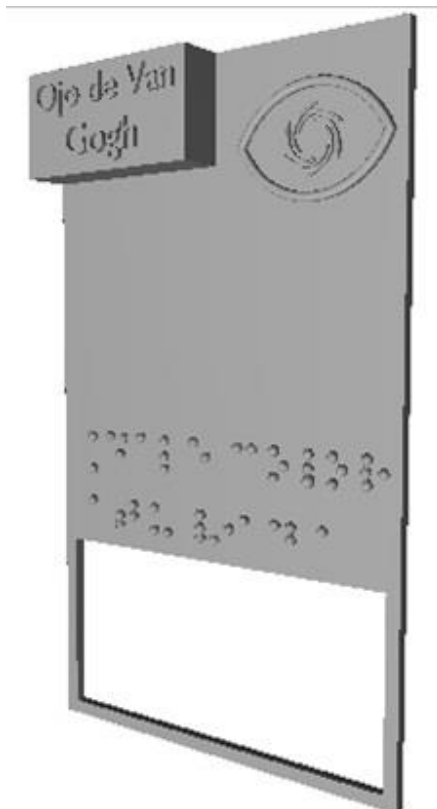
Se ha instalado el plugin Yoast que sirve para saber qué características faltan para mejorar el SEO de la página (optimización de los buscadores). Luego se ha procedido a cambiar la página de inicio agregando información y cambiando títulos de acuerdo a la palabra clave ingresada "ciego ven colores".

Debido a que la actividad cerebral de cada persona es distinta y no se puede tomar una medida precisa para usar cuando se parpadea, hemos empezado a investigar sobre desviación estándar. Para ver el grafico de la desviación estándar se investigó sobre la librería pandas, pero después vimos que no era necesario y teníamos problemas con el ajuste de escala.

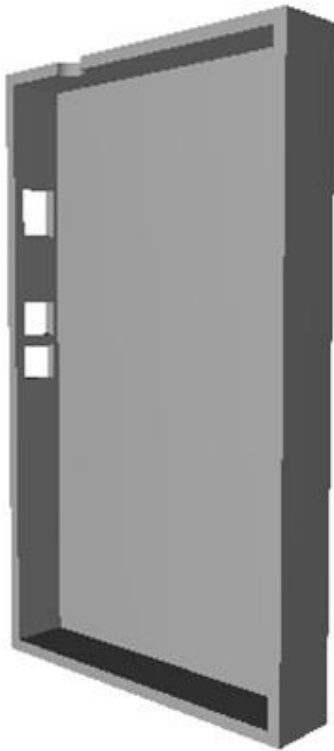
Se hizo un graficador de las ondas cerebrales a través de Arduino usando Matplotlib y hemos visto que la desviación estándar con 2 parpadeos no es tan estable ya que varía entre 50 y 60 pero cuando la persona parpadea 3 veces es mucho más estable dando unos valores de 80.

Se termino con el diseño 3D de la caja y tapa agregando un encastre para que entre a presión, quedando lista para su impresión.

Tapa de la caja



Caja por dentro



Semana del 8/06 al 15/06

Se crearon librerías para tener el código más prolijo.

- Librería de reconocimiento de texto tanto en español como en otro idioma, ya que tiene la opción de que cuando se reconozca que es otro idioma sea traducido
- Librería de reconocimiento de objetos, dando el contenido en español.

Se agrego para que se pueda reconocer la ubicación de dichos objetos detectados y limitando a que te diga solo los que la api detecta con una buena precisión (score).

Se ha mejorado la página "El proyecto", basándonos en las características de mejorar la legibilidad y algunas cuestiones de SEO.

Semana del 15/06 al 22/06

Se grabo un video mostrando el reconocimiento de colores, objetos y leyendo un texto. Luego se agregaron efectos especiales

Se hizo una publicación para IG por la entrevista que tuvimos el viernes, y se editó el video, musicalizo y se subió a IGTV. Nos hemos contactado con ASAC y estamos esperando la respuesta.

Mudamos la página a un nuevo host (SitioHispano) que nos salió 860 \$ el año

Se ha detectado cuando entra información del Mindwave para tomar el valor promedio de la persona en los primeros segundos.

Se adapto para que se reconozcan los 2 colores que más ocupan en la foto y luego ordenarlo para que primero este el que mejor se reconoció (score más alto).

Se actualizo la librería de objetos, agregándole la ubicación del mismo

Semana del 22/06 al 29/06

Se hizo una librería para el código de Mindwave para que quede más ordenado.

Se incluyo en la librería de RGB para que se pueda reconocer 2 colores si la diferencia de score es menor a 0.10

En la página web, agregamos las sliders en la paginas de noticias para que sea más cómodo y lindo.

Se hizo el anuncio para la radio 99.3 FM que sale el viernes y se ha publicado con la edición de musicalización el tercer video mostrando el reconocimiento de colores, texto y objetos

Se nos ocurrió para mejorar la calidad del reconocimiento de colores, crear una tabla de Excel y copiar los 140 colores que nos da la página de Wikipedia con sus RGB y sus nombres. Luego exportamos esa tabla a pandas y hemos agregado 2 condiciones para los colores negro y blanco.

Semana del 29/06 al 06/07

Se subió el IGTV de la entrevista de la radio y también se ha subido a YouTube y a la página web.

Se agrego dos condiciones de blanco y negro para descartar más rápido y si no se cumplen que recorran el código a través de búsqueda binaria en Rojo, haciendo que el código sea más rápido.

Hemos puesto que, si no se encuentra algún color en la tabla, se agregue ese valor a la misma. Luego será revisado por alguno de los integrantes, clasificándolo y quedando la tabla ya actualizada.

También hemos probado algunas fotos y clasificando los colores que no se han reconocido.

Semana del 06/07 al 14/07

Se incluyo en la página web un mensaje de bienvenida y también un nuevo símbolo cuando las paginas están cargándose.

Se ha contactado nuevamente con ASAC y MasScience y nos prometieron publicar pronto un artículo nuestro en su página. Se ha mandado mails tanto a medios nacionales como radios.

Se normalizaron todos los códigos haciendo que todos queden con la misma estructura de pedir el dato a obtener: color, texto u objeto.

Se ha creado una clase de búsqueda binaria y ordenado el main. Se creo una nueva clase llamada Google Vision Engine. En esta clase se han creados los métodos de:

- Pedir color
- Elegir ese color(query)
- Leer texto y reproducirlo
- Reconocer objetos y su ubicación

Nos hemos dado cuenta de que muchas variables y partes de código que no se usaban y las hemos borrado. Hemos agregado para que cada valor nuevo en la tabla, se reorganice de menor a mayor y se actualicen los id

Se preparo la encuesta en Google forms para saber que piensan las personas sobre el proyecto.

Semana del 14/07 al 21/07

Se ha realizado nuevamente el PCB, ya que hemos agregado el switch de 4 posiciones para que la persona elija una función del proyecto. También se ha reemplazado las pilas de litio y regulador de tensión y pusimos una entrada para un cargador portátil común.

Se paso el código de vibración, realizado en Arduino a Python para controlar el buzzer con la Raspberry. Esto se realizó ya que, si quisiéramos que el Arduino realice las vibraciones, se pierde tiempo para captar la señal que proviene del Mindwave. También se creó una función para que se pase de texto a audio

Semana del 21/07 al 28/07

Semana del 28/07 al 04/08

Hemos encontrado una tabla con 865 colores con sus nombres, valores hexadecimales y valores R, G, B e hicimos un código con dicha tabla buscando el color en la tabla que tenga menos diferencia con el color buscado.

Nos ha contactado la página de MasScience para decirnos que publicaron en su página un artículo sobre nuestro proyecto.

Clic [aquí](#) para ver el artículo del proyecto

Semana del 04/08 al 11/08

Se ha creado el método de audio y se agregó a la clase principal llamada Google Vision Engine, que es la clase que contiene todos los métodos. Una vez agregado la clase de audio se ha usado en los métodos de texto, objetos y color, para que se reproduzca por un parlante, el parámetro que se pida.

También nos hemos contactado con algunas personas que padecen ceguera para poder hacerles una entrevista y verificar cuáles son los problemas principales de estas personas en el día a día, y con esto nos han confirmado lo siguiente:

- Ellos se frustran no tener un sentido asociado al color y se preguntan que es, y al vivir en un mundo que se ve tienen que saber cómo vestirse, pero para ellos es lo mismo la combinación de colores. También buscan relacionar los colores con objetos, por ejemplo, que el rojo es del fuego
- Les molesta cuando las situaciones no tienen audios descriptivos. Y para cuando tienen que ir a comprar necesitan alguien que les diga cuanto sale y cuando es la fecha de vencimiento
- El reconocimiento de objetos para ellos es muy importante, ya que no pueden saber que hay cuando les mandan una foto, y si quieren ir a comprar necesitan de alguna persona para saber dónde está lo que ellos buscan.

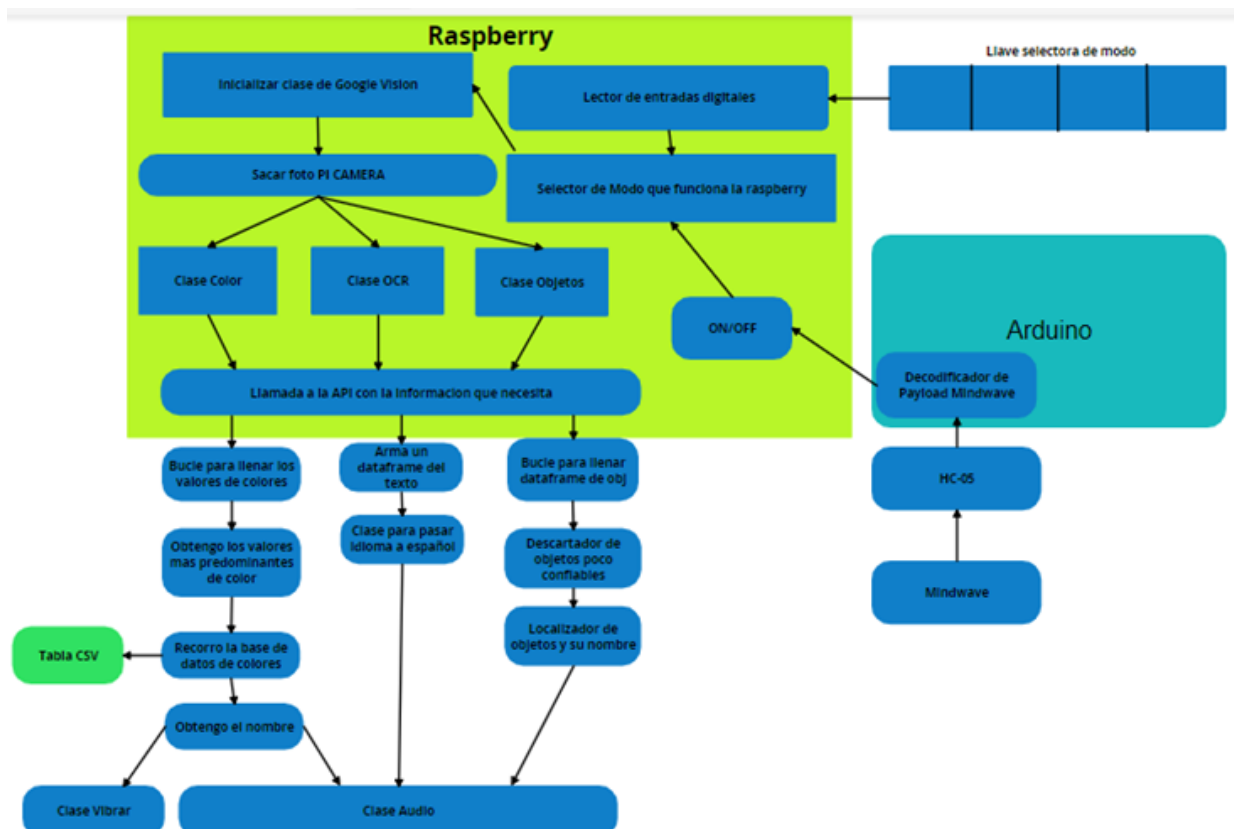
Semana del 11/08 al 01/09

Semanas para rendir exámenes y entrega de trabajos prácticos. Hemos hablado de los pasos a seguir del proyecto:

- Armar guion y video para presentar el proyecto
- Hacer un diagrama de arquitecturas para organizar el código.

Semana del 1/09 al 8/09

Luego de haber hablado con el profesor durante varias semanas de como seguir y que mejoras tendríamos que hacer, hemos empezado por reorganizar el código. Para eso hemos armado un diagrama de arquitecturas definiendo todas las clases padres e hijas



Hemos traducido el dataset de 860 colores para evitar traducirlo y hacer que el código sea más rápido

Semana del 8/09 al 15/09

Hemos realizado el refactoring de la clase de colores y un container de Docker para poder hacer un despliegue a producción.

Hemos empezado a investigar sobre cómo se arma un video para poder presentar nuestro proyecto y hemos planteado los puntos principales que tiene que tener este.

Semana del 15/09 al 22/09

Hemos realizado el refactoring de la clase de objetos y de la clase de texto, que heredan de la clase Google Vision Engine, que tiene como parámetros el cliente y la imagen que se saca con la cámara de la Raspberry. También se empezó a armar el guion.

Semana del 22/09 al 29/09

Hemos realizado el refactoring de la clase de audio que es la que se encarga de pasar de texto a audio y reproducirlo. A esta clase se le pasa un parámetro que puede ser: los colores, los objetos o el texto y lo reproduce por audio.

También se hizo el refactoring de la clase que se encarga de realizar una vibración para el color. A esta clase se le pasa como parámetro el nombre del color que se reconoció y de acuerdo a la tabla de vibración se hará la vibración.

Se ha terminado de armar el guion para el video, en el cual se detallan que tiene que durar entre 2 y 3 min, y hay 2 voces en off que irán explicando la problemática que tienen las personas con ceguera y cuál es la solución que ofrecemos nosotros, mostrando los videos que hemos realizado donde se puede observar el reconocimiento de objetos, lectura de texto y reconocimiento de color con otro video que muestra cómo será la vibración de un determinado color.

Semana del 29/09 al 06/10

Hemos realizado el main.py usando todas las clases que ya armamos las semanas pasadas, y hemos puesto para que si se detecta un parpadeo dependiendo de lo que se pida se obtenga ese parámetro.

Para probar el funcionamiento de las clases y el main.py hemos cargado las siguientes imágenes:

- Para reconocer objetos hemos usado como ejemplo, la siguiente imagen:



El resultado fue el siguiente:



```
main.py > ...
38     time.sleep(2)
39
40 elif(orden == "texto"):
41
42     text_manager = TextManager(IMAGE_PATH, CLIENT)
43
44     respuesta_texto_api = text_manager.get_response_api
45
46     texto = text_manager.get_text(respuesta_texto_api)
47
48     traducido = reproductor_audio.translate(texto, IDIOMA)
49     reproductor_audio.speak_audio(str(traducido))
50     time.sleep(2)
51
52 elif(orden == "objetos"):
53
54     object_manager = ObjectManager(IMAGE_PATH, CLIENT)
55
56     respuestas_objetos = object_manager.get_response_api()
57
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 2 1: Python

C:\Users\pc1\Google Drive\Cursos\ejemplos_python\proyecto>C:\Users\pc1\AppData\Local\Programs\Python\Python37\python.exe "c:/Users/pc1/Google Drive/Cursos/ejemplos_python/proyecto/main.py"

Ingrese la opción: objetos

Plátano está en el centro derecha

- Para reconocer el color hemos usado como ejemplo, la siguiente imagen:



El resultado fue el siguiente:

```
25 reproductor_audio = TextoToAudio()
26
27 orden = input("Ingrese la opcion: ")
28
29 if(orden == "color"):
30
31     colors_manager = ColorsManager(IMAGE_PATH, CLIENT, DATAFRAME_COLOR)
32     datos_rgb_api = colors_manager.get_response_api()
33
34     for value_color in colors_manager.get_color(datos_rgb_api):
35
36         traducido = reproductor_audio.translate(value_color, IDIOMA)
37         print(traducido)
38         reproductor_audio.speak_audio(str(traducido))
39         time.sleep(2)
40
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 2 1: Python

C:\Users\pc1\Google Drive\Cursos\ejemplos_python\proyecto>C:\Users\pc1\AppData\Local\Programs\Python\Python37\python.exe "c:/Users/pc1/Google Drive/Cursos/ejemplos_python/proyecto/main.py"
Ingrese la opcion: color
rojo

- Para reconocer el texto hemos usado como ejemplo, la siguiente imagen:



El resultado fue el siguiente:

```
main.py > ...
37     print(traducido)
38     reproductor_audio.speak_audio(str(traducido))
39     time.sleep(2)
40
41 elif(orden == "texto"):
42
43     text_manager = TextManager(IMAGE_PATH, CLIENT)
44
45     respuesta_texto_api = text_manager.get_response_api()
46
47     texto = text_manager.get_text(respuesta_texto_api)
48
49     traducido = reproductor_audio.translate(texto, IDIOMA)
50     print(traducido)
51
52     reproductor_audio.speak_audio(str(traducido))
53     time.sleep(2)

```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 2 1: Python

C:\Users\pc1\Google Drive\Cursos\ejemplos_python\proyecto>C:\Users\pc1\AppData\Local\Programs\Python\Python37\python.exe "c:/Users/pc1/Google Drive/Cursos/ejemplos_python/proyecto/main.py"
Ingrese la opcion: texto
1000
BANCO CENTRAL DE LA
REPÚBLICA ARGENTINA
M
1000
MIL PESOS
HORNERO
AVE NACIONAL

Hemos ordenado las carpetas de GitHub y realizado los distintos Readme para saber de qué se trata cada carpeta y se empezó a probar una plataforma, Animaker, para realizar la animación que irá acompañada del guion armado.

Prueba de la animación para el video



Semana del 06/10 al 13/10

Hemos empezado a crear columnas con los nombres de los colores principales para pasar a la función de vibrar, mientras que por audio saldrá el nombre completo, por ejemplo: rojo será para la vibración mientras que por audio saldrá rojo morado.

Se ha grabado el video para presentar el proyecto y se empezó a realizar la animación para mostrar la problemática del ciego y la solución que ofrecemos con el proyecto. También se ha realizado el informe descriptivo del proyecto.