**PROYECTO ANUAL**

**Materia:**

**Sistemas Operativos**

**Integrantes:**

**Bruno Nahuel Typek**

**Juan Ignacio Lopez**

**Lucas Ariel Conde Cardó**

**Maia Vaccarilli**

**Martín Castillo**

**Docentes:**

**Leila Coronel**

**Rodrigo Daiqui**

**Grupo Nº3**

**6ºC**

**Ciclo lectivo 2022**

1. **Ideas:**

En esta primera entrega de la carpeta de campo sobre el proyecto vamos a mencionar las distintas ideas/propuestas que pensamos para luego pasar a describir y explicar las tres ideas que elegimos como factibles. Finalmente, vamos a describir la idea elegida para nuestro proyecto realizando una explicación y justificación de distintos puntos clave a la hora de realizar un proyecto, tales como el objetivo, el alcance, las metas, entre otros.

**Distintas ideas que tuvimos:**

* Red social imitación de Tinder.
* Red social imitación de Pinterest.
* Página en la que se denuncien robos.
* Editor de texto online (similar a Google Docs).
* Red social sin ningún tipo de censura (similar a la idea de Elon Musk con Twitter).
* Traductor de vocabulario “raro” (similar a Urban Dictionary).
* Servicio de mensajeria centrado en fotos y videos (similar a Snapchat).
* Página de compra y venta de artículos ilegales.
* Página para resolver ejercicios de programación (similar a Codechef).
* Buscador de comidas en base a ingredientes.
* Organizador y distribuidor de laboratorios (idea de la profesora).
* Traductor de lengua de señas (idea de la profesora).

Luego de plantear esas ideas seleccionamos tres que nos parecían factibles, estas fueron:

* Página en la que se denuncien robos: Esta idea consistía en llevar a cabo una página web es la que se pudieran denunciar robos para luego estos ser marcados en un mapa y poder ver cuales zonas son peligrosas y cuales no, estas se podrían clasificar por colores y por nombres (como “Zona segura”, “Zona insegura”, etc). Además, los robos se podrían describir así las personas que luego circulen por esa zona sean capaces de saber si tienen que tener cuidado al ver a alguien con características “sospechosas”.
* Organizador y distribuidor de laboratorios: Esta idea consistía en crear una página que utilizara un algoritmo de árbol genético para así poder generar la mejor solución ante el problema de la mala distribución de los laboratorios, además, esta idea también se podía extrapolar a otras áreas/problemáticas.
* Traductor de lengua de señas: Esta idea consiste (spoiler: es la idea elegida) en crear una página web/programa/aplicación para dispositivos Android (todavía no está definido esto) que permita traducir lengua de señas a texto y voz. A continuación se explicará con exactitud.

**Proyecto elegido:**

Como ya mencionamos anteriormente, este proyecto consiste en la realización de un “programa” que permita la traducción de lengua de señas a texto y voz.

El funcionamiento de este va a consistir en la captura de video con una cámara, la cual podría ser cualquier webcam, en el caso de que desarrollemos el proyecto para sistemas operativos de escritorio o en formato web, o la cámara frontal o trasera de un dispositivo móvil Android, en el caso de que desarrollemos el proyecto para esta plataforma, para luego poder realizar una clasificación de objetos de cada fotograma.

Esta clasificación de objetos se llevará a cabo con Mediapipe, un “framework” perteneciente a Google y de código abierto que perimite realizar esta tarea y entre una se sus tantas funciones tiene la de identificación y “trackeo” de manos. Otro punto positivo de esta herramienta es que permite ser utilizada en distintos sistemas operativos y lenguas de programación.

Al mismo tiempo de la clasificación de objetos que permitirá identificar las manos también se realizará una comparación que permitirá conocer que esta expresando el usuario. Esto se realizará con otra plataforma de Google llamada TensorFlow, esta ofrece herramientas de “machine learning” que nos van a servir para buscar la palabra correcta comparando las imágenes tomadas con las imágenes cargadas dentro de un “dataset” pre-entrenado con palabras y letras en lengua de señas.

El ya mencionado dataset será una compilación de distintos datasets**\*** disponibles en páginas web como Kaggle y RoboFlow, entre otras. Con respecto a este surgió un problema que será detallado en el informe de avance.

Una vez que TensorFlow encuentre una coincidencia entre los gestos realizados por el usuario y una palabra en el dataset, esta pasará a ser mostrada por pantalla y, al mismo tiempo, reproducida de manera auditiva para que las demás personas presentes, que no hablen lengua de señas, puedan comprender lo que se esta diciendo.

En resumidas cuentas, este proyecto se resume en los siguientes puntos:

* Objetivo: Crear un traductor de lengua de señas a texto y voz para poder ser comprendido por gente que no hable dicho lengua.
* Alcance: Toda la comunidad de personas mudas que no pueden lograr una comunicación fluida con el entorno que los rodea puesto que este, en la mayoría de los casos, no sabe comunicarse en lengua de señas.
* Metas: Definir en que plataforma se realizará el proyecto (en el caso de no realizarlo en formato web deberemos crear una página web informativa sobre este), recolectar la suficiente información para poder armar un buen dataset, desarrollar el programa integrando todas las herramientas ya mencionadas, entre otras.
* Hitos: Todavía no están bien definidos. Uno era la aprobación de una idea y ya se logró, otro también será la presentación final del proyecto.

**\***Este puede estar conformado por distintos datasets que encontramos en internet:

* ASL (American Sign Language):

https://www.kaggle.com/datasets/pramod722445/sign-language-dataset/code

https://public.roboflow.com/object-detection/american-sign-language-letters/1/images/32acbe34c48eef8becffe27ff157f8c9

https://www.kaggle.com/datasets/datamunge/sign-language-mnist

https://github.com/dxli94/WLASL

https://www.microsoft.com/en-us/research/project/ms-asl/

https://www.kaggle.com/datasets/jordiviader/american-sign-language-alphabet-static

https://www.kaggle.com/datasets/grassknoted/asl-alphabet

https://www.kaggle.com/datasets/danrasband/asl-alphabet-test

https://www.kaggle.com/code/namanmanchanda/asl-detection-99-accuracy/data

https://www.kaggle.com/datasets/mrgeislinger/asl-rgb-depth-fingerspelling-spelling-it-out

* LSCh (Lengua de Señas Chilena):

[https://www.kaggle.com/datasets/bpablo27/abecedario-lengua-de-seas-chilenoespaol](https://www.kaggle.com/datasets/bpablo27/abecedario-lenguaje-de-seas-chilenoespaol)

* PSL (lengua de Señas Pakistaní):

https://data.mendeley.com/datasets/y9svrbh27n/1

* JOS (lengua de Señas del Árabe Levantino):

https://data.mendeley.com/datasets/y7pckrw6z2/1

* LSU (Lengua de Señas Uruguaya):

https://github.com/ariel-stassi/TRELSU-HS

* Hindi:

https://dl.acm.org/doi/10.1145/3394171.3413528

* LSA (Lengua de Señas Argentina):

http://facundoq.github.io/datasets/lsa64/

http://facundoq.github.io/datasets/lsa16/

1º entrega, realizada el día 11/06/2022.

**2º Entrega (11/06 a 07/08):**

Hasta el día de hoy se realizaron diversas tareas para avanzar con el proyecto las cuales serán detalladas posteriormente, además, también surgieron muchos problemas que serán presentados en el “Informe de avance”.

Las tarea desarrolladas hasta el día de la fecha fueron:

* Se definió la plataforma sobre la que se realizará el proyecto, el mismo es programado en Java y será para Android.
* Se definieron las principales librerias a utilizar, las mismas son OpenCV, en su versión 3.4.13, para la cámara y TensorFlow Lite para la detección de objetos.
* Se creó el proyecto en Android Studio, este es compatible con Android 5.0 (SDK 21) o superior. No se pudo usar una versión inferior por problemas de compatibilidad con las diferentes librerías y herramientas a utilizar, aún así, menos del 5% de todos los dispositivos Android tienen una versión inferior a 5.0.
* Se implementó OpenCV y se creó una clase para la cámara, que se abre cuando el usuario pulsa sobre el botón que colocamos en la clase principal.
* Una vez ya implementado OpenCV, se configuró que la cámara ocupe toda la pantalla, puesto que esta, dependiendo del dispositivo, aparecía “achicada” y “descentralizada”.
* Después se implementó TensorFlow Lite, la librería para la detección de objetos, y se cargó un modelo desarrollado por Google para la detección de algunos objetos este será utilizado como prueba hasta terminar de desarrollar por completo la detección de objetos.
* El siguiente y último paso hasta ahora consistió en programar el interpretador de la librería para que pueda detectar objetos y guardarlos para luego ser mostrados por pantalla.

En el Informe de Avance están detallados otros puntos, como las tareas en proceso y pendientes, además de los problemas presentados durante el desarrollo.

2º entrega, realizada el día 07/08/2022.

**Entrega Final (07/08 a 25/10): Mudeitor**

Durante este lapso de tiempo comenzamos con la realización de la página web, extrayendo una plantilla de un usuario de GitHub. A partir de esta continuamos trabajando tanto Back-End como Front-End sobre este sitio. Fuimos modificando los textos preestablecidos, utilizando los mismos para añadir información sobre la aplicación y de los roles durante la realización de la misma; agregamos y cambiamos las imágenes por fotos de los integrantes y capturas de pantalla de nuestra aplicación; pensamos un nombre para el proyecto (“Mudeitor”) y diseñamos su boceto para el título utilizando una fuente animada y llamativa para el usuario, como también la implementación de un logo creado y editado con Photoshop. Añadimos un API de Google Maps que muestra la ubicación exacta del colegio y al lado su información (Teléfono, Mail, etc…). Por último, creamos una sección de comentarios programada en PHP y con una base de datos “armada” en MySQL.

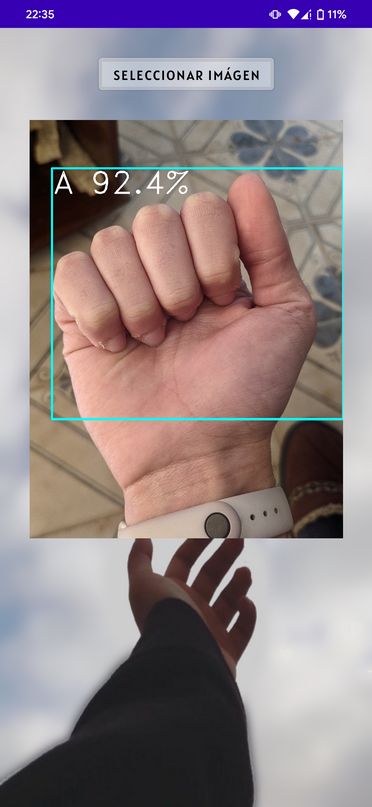
Al final de todo esto, hosteamos la página en “000WebHost” ([www.000webhost.com](http://www.000webhost.com/)), desde ahí podemos administrar todo lo que respecta a la base de datos y los archivos de la página.

Con respecto a la aplicación, luego de la prueba con objetos, comenzamos con la programación para que detecte las manos de la persona. Al finalizar, continuamos con la detección del lenguaje de señas separando el trabajo en dos partes, una que logre traducir en palabras a través de una imagen de la galería del usuario y por otro lado un modelo de traducción en tiempo real realizando el armado de frases. Conseguimos el pasaje de las frases a voz/audio, el cambio entre la cámara frontal y trasera, y finalizamos el proyecto con un diseño practico y atractivo para el cliente.

\*Ver puntos específicos del desarrollo y los problemas encontrados en el “Informe de Avance”.

**Capturas de la aplicación:**

****

****

****

**Capturas de la página web:**

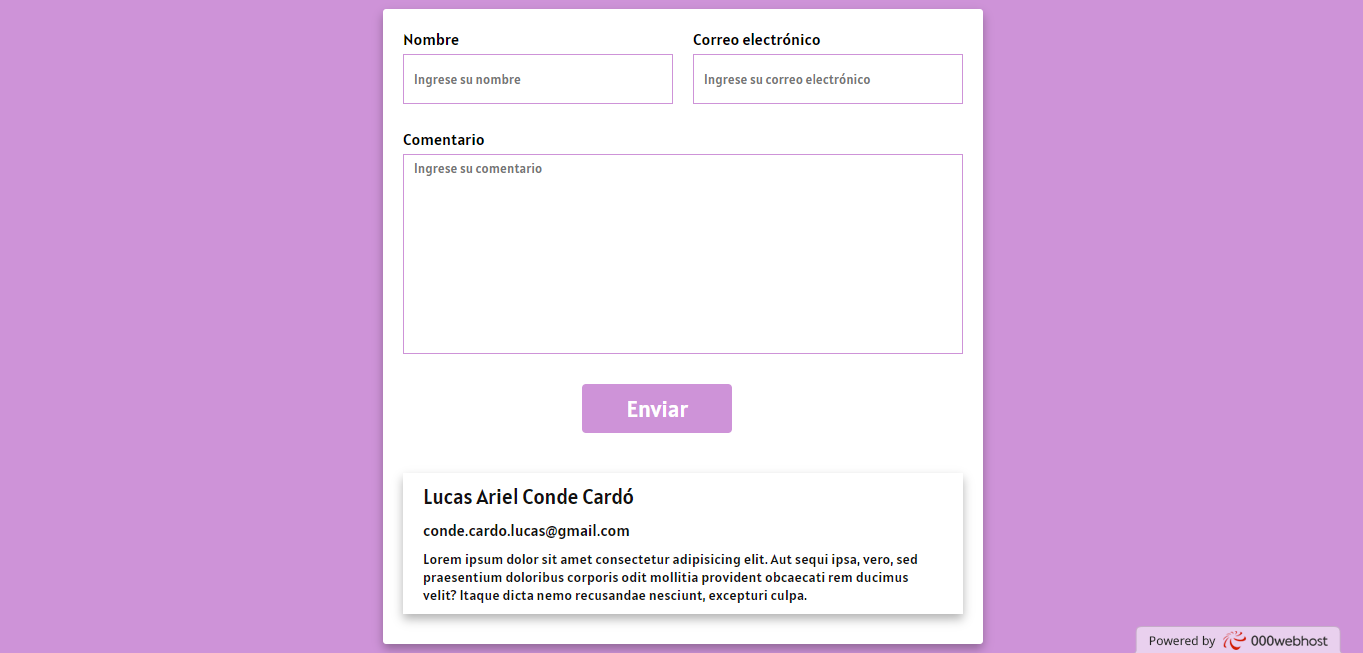












**Integrantes:**

El proyecto está integrado por 5 integrantes. Lucas Conde, es el encargado del desarrollo de la aplicación, administrar la base de datos y distribuir las diferentes tareas a los miembros del grupo. Por otro lado, también tenemos a Juan Lopez y Bruno Typek, los cuales se encargan del diseño de la página web y la toma de imágenes para la misma. También está Martín Castillo, quien también se encarga de la página web y de la redacción de la documentación (es posible leerla en el apartado de "Documentos") y de los textos visibles en la página. Por último, tenemos a Maia Vaccarili que se encargó de tomar imágenes para el dataset y colaborar con distintas tareas a lo largo del desarrollo. Como mención especial, queremos destacar la colaboración de Facundo Uthurralt, que se encargó de diseñar todos los logos y fondos, y de los profesores, Leila Coronel y Rodrigo Daiqui, que ayudaron y ayudan a lo largo del proyecto con ideas y apoyo.

**Links del proyecto:**

* Archivo instalable de la aplicación (Android):

https://drive.google.com/file/d/1cJLuYD6JhbA-pUIzwYXGucmHT0mFDkf-/view

* Código de la aplicación:

https://drive.google.com/file/d/1HqEEKg9CHFwVUN3qCfTn3Fy5F-xGLcUn/view?usp=sharing

* Repositorio de GitHub:

https://github.com/LucasConde22/Mudeitor

* Página web:

https://mud3itor.000webhostapp.com/

