

# INFORME FINAL

Azcona Marcos - Alvarez Cristian Gabriel

## Introducción

El software desarrollado y planteado como trabajo final de materia se basa en un juego, Sopa De Letras, el cual consiste en descubrir palabras distribuidas de forma "aleatoria" en una grilla. El objetivo del mismo es fortalecer la enseñanza de los tipos de palabras (sustantivos, adjetivos y verbos) abordados en la materia "Prácticas de Lenguaje", facilitando el entendimiento de los mismos ,y puesta en práctica en escuelas de nivel primario.

## Desarrollo

El videojuego desarrollado fue planteado con el siguiente esquema:

- Oficina
- *Ingreso de palabras.*
- *Reporte de palabras en conflicto*
- *Configuración por parte del docente.*
- *Solución de la sopa.*
- *Muestra de resultados*

**Oficina:** se pedirá al usuario seleccione la oficina en la que se encuentra desde una ventana que contendrá las oficinas disponibles. De esta oficina seleccionada se obtendrá su último registro de temperatura, con el cual se cambiará el "look and feel" del programa acorde a la temperatura obtenida.

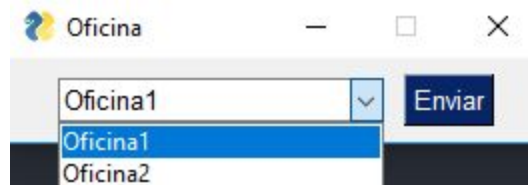


Figura 1: Lista De Oficinas

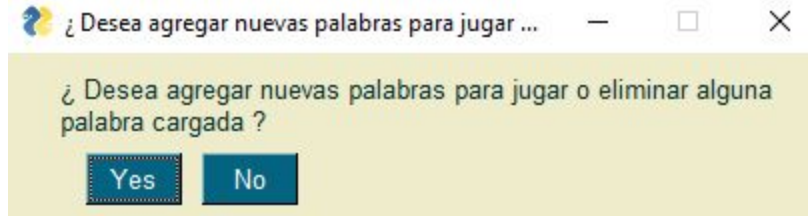


Figura 1.1: Ejemplo de la interfaz con el Look & Feel cambiado

***Ingreso/Modificación de palabras:*** se le dará al docente la opción de ingresar palabras, las cuales se distribuirán de forma aleatoria en la grilla. Al ingresar una palabra, se le mostrará al mismo una ventana la cual indicará el tipo de palabra, y en caso de que la misma haya provocado error entre Pattern Es(módulo externo para verificación de palabras) y Wikcionario(diccionario web), se le mostrará una ventana que indicará que la palabra se incluirá en el reporte. Si una palabra ingresada ya había sido ingresada con anterioridad, se dará la opción de eliminarla de la lista de palabras a distribuir en la grilla.

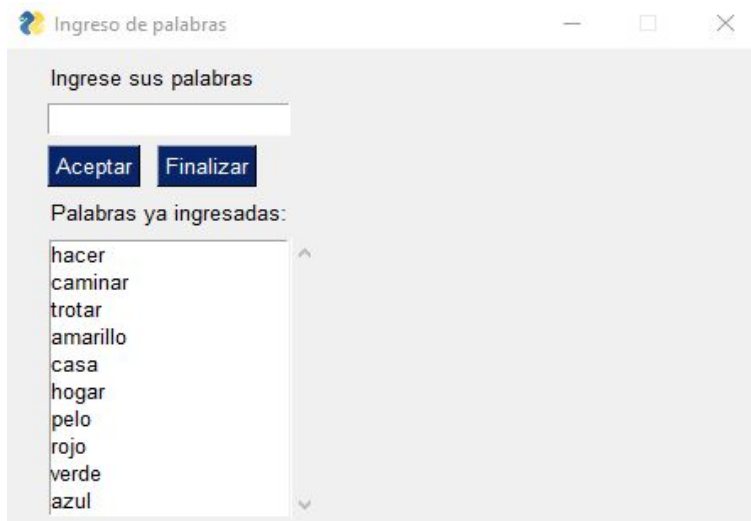


Figura 2: Ventana de Ingreso de Palabras

**Reporte de palabras en conflicto:** al docente se le mostrará una ventana con opciones de tipo de fuente que se usará a la hora de mostrar el reporte. En caso de que no haya palabras conflictivas, se le mostrará que no se generó conflicto.

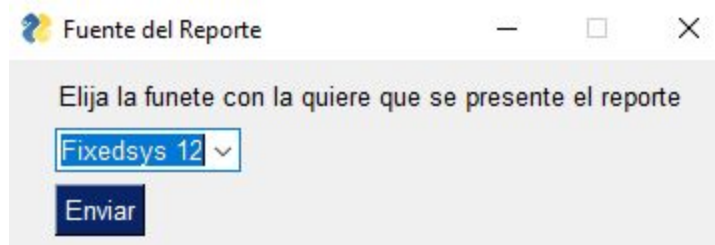


Figura 2.1: Ventana que permite la selección de la fuente.

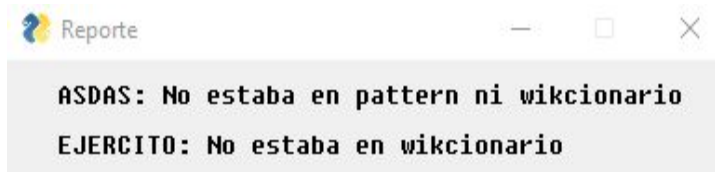


Figura 2.2: Reporte de palabras en conflicto.

**Configuración por parte del docente:** se le mostrará un menú de opciones donde podrá seleccionar:

- Cantidad de palabras por tipo (para cada tipo se dará un máximo de 6 palabras a distribuir)
- Color para representar cada tipo.
- Orientación de las palabras en la grilla (horizontal, vertical).
- Mayúsculas o minúsculas.

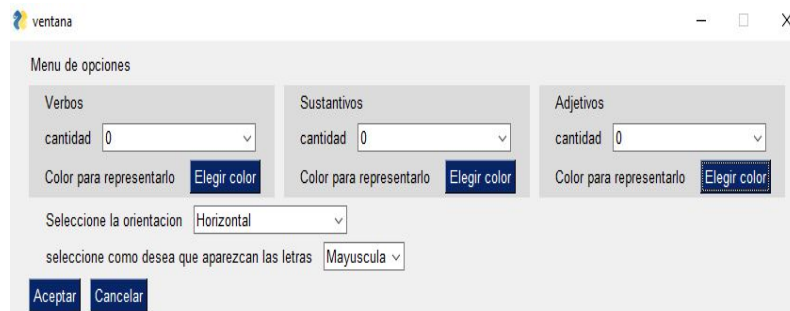


Figura 2.3: Ventana de configuración por parte del docente

**Solución de la Sopa:** el alumno debe elegir el tipo de palabra. Al elegirlo, cada una de las celdas seleccionadas quedará pintada con el color asociado a ese tipo de palabra. A su vez, durante la resolución del juego, el alumno podrá solicitar ayudas. Dependiendo del tipo de ayuda que el alumno pida, se le mostrará una ventana con información acerca de las palabras dispersas en la sopa.

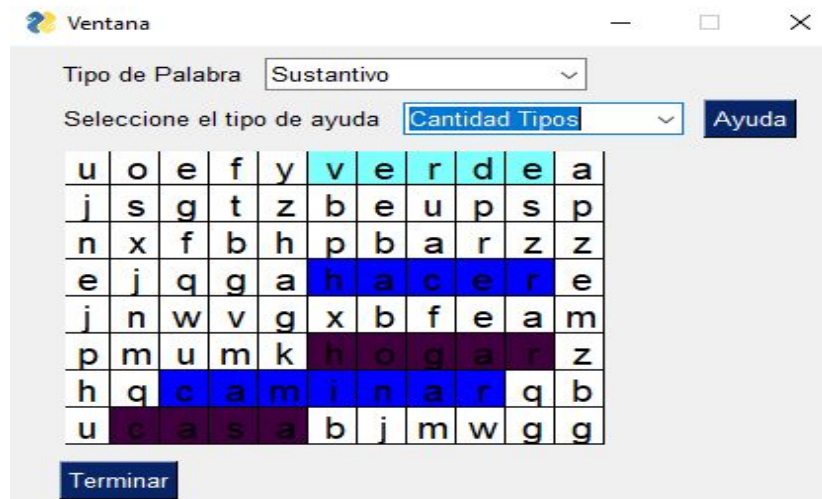


Figura 2.4: Sopa De Letras

**Muestra de Resultados:** una vez que el alumno desee finalizar la solución de la sopa, deberá apretar el botón terminar. Si éste completó exitosamente la misma, se mostrará una imagen de un premio como símbolo de éxito. En caso contrario, al alumno se le mostrará una ventana con una frase motivadora. En caso de haber logrado o no completar el juego, se mostrará un listado con las palabras que fueron acertadas y aquellas que no fueron acertadas.

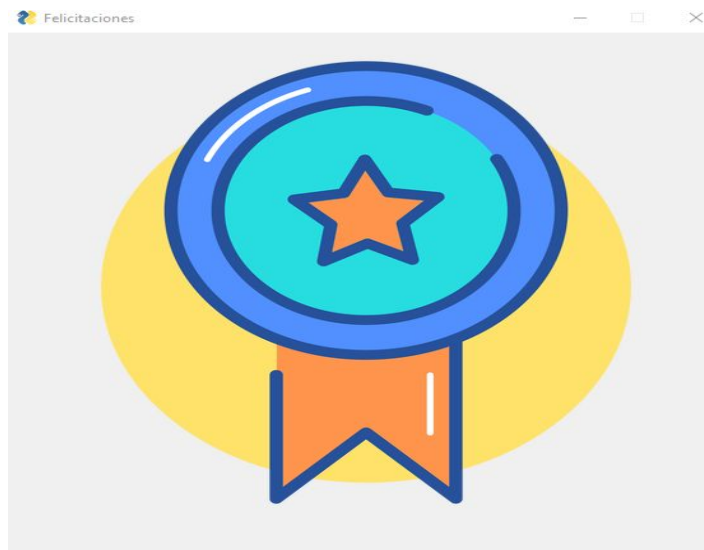


Figura 2.5: Ventana de éxito tras completar la sopa.

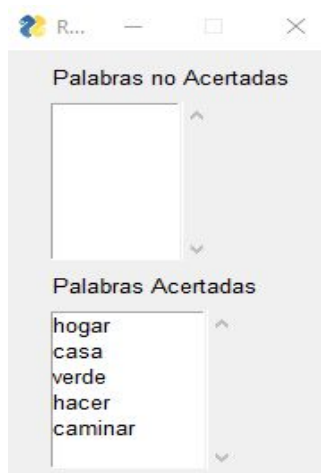


Figura 2.5a: Ventana con la lista de palabras que fueron Acertadas, y aquellas que no lo fueron

## *Trabajo con sensores*

Para el trabajo con sensores, se desarrollaron dos aplicaciones. Por un lado, una aplicación denominada "registro\_ambiental", la cual toma los datos de los sensores de temperatura y humedad de la raspberry, para guardarlos en un archivo denominado "datos-oficina", con una periodicidad de 1 minuto. Por otro lado, la segunda aplicación desarrollada, denominada "muestra\_datos" se encarga de mostrar los datos censados en matrices Led 8x8 sincronizadas cuando se detecte un sonido en el ambiente.

Ambos programas procesarán los datos correspondientes a una oficina proporcionada por el usuario.

## *Temas investigados*

### *Pygame*

Pygame es una librería open Source de Python diseñada a partir de la librería SDL (librería multiplataforma escrita en C++ para el manejo en bajo nivel de periféricos como teclado, mouse, mandos de consola, etc.), orientada al desarrollo de aplicaciones multimedia, como videojuegos, editores gráficos, etc.

### *Pattern (Pattern.es y Pattern.Web)*

Pattern: "Pattern es un módulo "web mining" para el lenguaje de programación Python. Tiene herramientas para la minería de datos(Google, Twitter y la API de Wikipedia, "web crawler", HTML DOM parser)... machine learning(modelo de espacio vectorial, clustering, SVM), análisis de redes y 'canvas visualization' "

Pattern.web: "el módulo pattern.web tiene herramientas para minería de datos en línea,; petición asincrónica, una API uniforme para servicios web(Google, Bing, Twitter, Facebook, Wikipedia, Wiktionary, Flickr, RSS), HTML DOM parser... "

### *PySimpleGUI*

PySimpleGUI es un módulo open Source de Python destinado al desarrollo de interfaces gráficas. Como su nombre lo indica, el mismo ofrece recursos con poco nivel de complejidad, para facilitar el diseño e implementación de interfaces.

## *Git - GitHub*

Git es un controlador de versiones destinado al manejo y control de proyectos de software, el cual permite tener ramas en un mismo proyecto, permitiendo que distintos desarrolladores puedan trabajar de forma simultánea en distintas partes del proyecto de manera ordenada, para luego poder unir esas partes.

Definido por la página oficial: "Git es un sistema de control de versiones libre y open source, diseñado para el manejo de pequeños hasta grandes proyectos con rapidez y eficiencia.

"Github es una plataforma de desarrollo inspirada en la forma en que tú trabajas. Desde el open source hasta el negocio empresarial, puedes subir y analizar código, dirigir proyectos y producir software junto a 36 millones de desarrolladores."

## *DHT12 - Sensor de Humedad/Temperatura*

El sensor de Humedad - Temperatura DHT12 es un dispositivo de almacenamiento de salida digital calibrada de temperatura y humedad. El mismo, utiliza un protocolo de comunicación conocido como "I2C" (Inter Integrated Circuit), el cual permite la conexión entre varios "esclavos" a un mismo "maestro", pudiendo manipular a uno o más esclavos en forma simultánea, y Sincrónica, en que se espera a que se de la comunicación para poder continuar. Para la comunicación se utilizan 2 líneas: "SDA" y "SCL". SDA es una línea de transferencia de datos entre maestro - esclavo/s para lectura y/o escritura.

Ya que la transferencia de datos es bit a bit y el protocolo de comunicación es sincrónico, quiere decir que la transferencia por la línea SDA es controlada por la línea SCL.

El dispositivo DHT12, por tanto, cuenta con 4 líneas. Dos de las ya mencionadas SDA y SCL, VDD(power supply, el cual soporta un voltaje entre 2.2 Voltios y 5.5 Voltios) y GND (línea a tierra).

Permite detectar temperaturas de entre -20 °C hasta 60 °C, y Humedad de entre 20 y 95 RH(nivel de agua vaporizada en el aire).

## *ADAFRUIT - DHT*

Adafruit\_DHT es un módulo de python el cual permite el manejo de dispositivos DHT para sensor temperatura y/o humedad.

## *RPi - GPIO*

Rpi.GPIO es un módulo de python que permite controlar los pines GPIO de la raspberry. GPIO es una conjunto de pines de entrada/salida de propósito general, los cuales pueden ser tratados, en conjunto, como un único pin.

El micrófono conectado al pin de tipo GPIO se trata como un dispositivo genérico, de manera que para poder usarlo, se configura en forma directa la raspberry. Por lo tanto, para su configuración, se debe establecer el "canal" de comunicación entre el dispositivo y la raspberry, es decir, el número de pin físico en el que está conectado el dispositivo en el GPIO. Para esto, se cuenta con dos formas de configurar el canal. Por un lado encontramos el modo BCM, el número físico grabado en la raspberry, que se encuentra al lado de los enchufes, como "GP22"; y por otro lado el modo "BOARD", en que el número se determina por el número de enchufe.

En el modo "BCM", se debe trabajar con un diagrama de qué número de canal va conectado al pin en la placa RPi, siendo manejado desde el código, en cambio, en el modo "BOARD", no es necesario la revisión y manejo de la placa RPi desde nuestro código.

## *Matriz Led*

Para el trabajo realizado, se ha trabajado con la matriz Matriz7219.

Definido por el DataSheet de MAX7219: "Los MAX7219 / MAX7221 son controladores de pantalla de cátodo común de entrada / salida de serie compactos que interactúan con Microprocesadores ( $\mu$ Ps) para pantallas LED numéricas de 7 segmentos de hasta 8 dígitos, pantallas de gráficos de barras o 64 LED individuales".

La misma se conecta a través de pin tipo SPI, la cual cuentan con 5 líneas. Estas son:

1. VCC: pin de alimentación. Requiere de más de 5V.
2. GND: línea a tierra.
3. DIN: línea de entrada/salida de datos desde el Maestro al Esclavo. En nuestro caso, se conecta al pin SPI MOSI, actuando como entrada de datos desde la Raspberry hacia la matriz.
4. CS: línea de selector de chip. (Selecciona el esclavo con el que se comunicará el maestro)
5. CLK: encargado de la sincronización de la comunicación entre Maestro - Esclavo

Para el manejo de la matriz, se utiliza la librería luma.core, la cual provee compatibilidad con pillow (librería para el manejo y renderización de imágenes), permitiendo funcionalidades como soporte en la renderización de texto para pantallas pequeñas, entre otras.



# Problemas surgidos

## Desarrollo con Pygame

Inicialmente, el proyecto fue desarrollado con la librería Pygame. Sin embargo, el programa tuvo que ser cambiado "parcialmente" para ser desarrollado con PySimpleGUI, surgiendo problemas tales como definir los tamaños de la ventana, inconvenientes con los toques de la pantalla "graph\_bottom\_left" y "graph\_top\_right" (debido a que no comprendíamos correctamente el funcionamiento de los toques de la ventana), y distribución de los objetos "Celda" en la grilla (tamaño y posición en x,y).

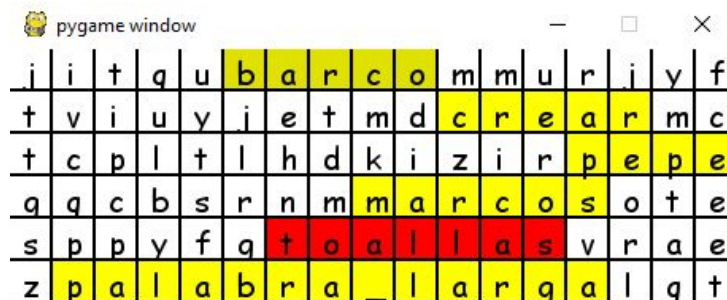


Figura 3: Vista de la Sopa De Letras de palabras elegidas con sus colores respectivos

La versión mostrada en la Figura 2 fue la última desarrollada con pygame, en la cual se distribuían las palabras correctamente en la grilla, y los colores con los que se rellenaban las celdas se elegían con las flechas del teclado.

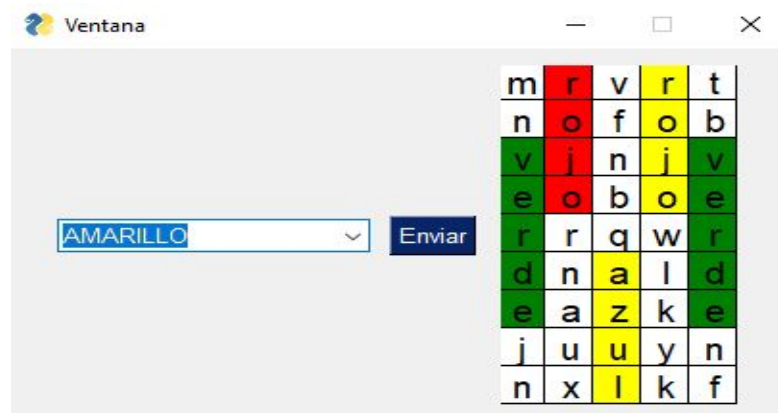


Figura 3.1: Primera Versión desarrollada con PySimpleGUI

## Validación de una palabra

El problema surge a partir de la verificación de si las celdas elegidas se corresponden con una de las palabras distribuidas en la grilla y, por otro lado, determinar si el tipo con el que fue seleccionada es el correspondiente para la palabra.

Una de las primeras ideas planteadas consistió en almacenar las letras de la celdas seleccionadas en una estructura de datos, para luego verificar si el conjunto de las mismas era una palabra válida. En rasgos generales, esto consistía en un diccionario, cuyas claves "primarias" eran los colores elegidos para los tipos; y los valores asociados a cada clave, un diccionario con clave fila/columna (dependiendo de la orientación elegida), y valor un String en el que se concatenaban las letras de cada celda seleccionada. Sin embargo, esta idea fue descartada debido a que no se podía verificar si las celdas seleccionadas eran consecutivas entre ellas.

Como solución a lo planteado con anterioridad, se modificó la carga de la estructura de datos. Para esto, en lugar de almacenar las letras de las celdas seleccionadas, se almacena una lista que contiene número de fila/columna. Por tanto, verificar si las celdas eran o no consecutivas dejó de presentar inconvenientes.

## Conclusiones

En conclusión, el proyecto presentado tiene como finalidad, por un lado, el manejo de la Raspberry (conocer sus componentes, y la creación de una aplicación para el manejo de la misma), y por otro lado, el desarrollo de un videojuego, *Sopa de Letras*, que abarca contenidos dados en el nivel primario, es decir, el aprendizaje de los tipos de palabras.

## Referencias

1. <https://www.pygame.org/docs/tut/PygameIntro.html>
2. <https://www.pygame.org/docs/ref/display.html>
3. <https://www.pygame.org/docs/ref/draw.html>
4. <https://pysimplegui.readthedocs.io/en/latest>
5. <https://www.clips.uantwerpen.be/pages/pattern-web>
6. <https://github.com/PySimpleGUI/PySimpleGUI/tree/master/DemoPrograms>
7. <https://www.raspberrypi.org/>
8. <https://www.instructables.com/id/DHT12-i2c-Cheap-Humidity-and-Temperature-Sensor-Fa/>
9. [https://github.com/adafruit/Adafruit\\_Python\\_DHT](https://github.com/adafruit/Adafruit_Python_DHT)

10. <http://pdf1.alldatasheet.com/datasheet-pdf/view/73745/MAXIM/MAX7219.html>
11. <http://www.robototehnika.ru/file/DHT12.pdf>
12. <http://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>
13. <http://www.pibits.net/code/dht12-temperature-sensor-and-raspberry-pi-example.php>
14. <https://www.raspberrypi.org/documentation/usage/gpio/>
15. <https://sourceforge.net/p/raspberry-gpio-python/wiki/BasicUsage/>
16. <https://raspberrypi.stackexchange.com/questions/12966/what-is-the-difference-between-board-and-bcm-for-gpio-pin-numbering>
17. <https://luma-led-matrix.readthedocs.io/en/latest/install.html#gpio-pin-outs>
18. <https://www.drouiz.com/blog/2018/06/25/uart-vs-spi-vs-i2c-diferencias-entre-protolos/>

## *Anexo: Guía de Usuario*

### Funcionamiento del programa

El programa deberá ejecutarse desde el archivo main.py

Al principio de la ejecución se le pide al profesor ingresar palabras para jugar. En caso de que no haya ingresado palabras, se le pedirá que vuelva a ingresar palabras.

Cada palabra ingresada pasará por un proceso de validación. Una vez que haya ingresado palabras, se mostrará un reporte de palabras que generaron conflicto entre Pattern y Wikcionario (si es que hubo palabras en conflicto. En caso contrario se mostrará que no habrá reporte).

A la hora de mostrar el menú, se le dará la posibilidad de elegir:

- Cantidad de palabras que se mostrarán para cada tipo
- Colores para cada tipo
- Orientación del juego: Vertical u Horizontal
- Mayúsculas o Minúsculas.

## Ventana de juego

Al iniciar el juego, al alumno se le mostrará una barra donde podrá seleccionar el tipo de palabra que desea marcar, de manera que una vez que un celda haya sido seleccionada, se mostrará la misma coloreada con el color que haya elegido para ese tipo en el menú de configuración de juego. Debajo, se mostrará la grilla con las palabras distribuidas y la posibilidad de seleccionar, mientras juega, el tipo de ayuda que el alumno pida.

## Forma de juego

El alumno podrá marcar las celdas que se correspondan con las palabras distribuidas de los tipos.

Una vez que el alumno haya terminado de elegir las palabras y considere que ha completado el juego, deberá presionar el botón "Terminar", el cual mostrará una ventana con una lista de palabras acertadas y no acertadas.