

Trabajo Práctico Integrador

Fecha de entrega:

- a. 03/07/2024

Forma de entrega:

- b. Se deberá generar un informe donde contenga los siguientes puntos
- **Carátula:** con los integrantes del grupo
 - **Link a un repositorio de github:** subir el código fuente, proyecto, etc, así como cualquier otro archivo que crea conveniente.
 - **Descripción:** Una breve descripción sobre la solución de software que implementaron; que finalidad tiene, lenguaje utilizado, frameworks, si tiene restricciones de hardware, versión de sistema operativo, de navegador, etc.
 - **Manual de usuario:** Una guía sobre como utilizar la solución software, pueden incluir capturas, paso a paso, etc. Puede ser un video también y directamente en el documento agregan el link a donde se encuentra el archivo de video.
 - **Conclusiones:** En esta sección se debe describir las dificultades que encontraron al realizar el trabajo práctico
- c. Entregar el informe por plataforma MleL. Este debe ser en formato .pdf, con nombre TP1_Integrador_NumerodelGrupo.pdf.

Enunciado:

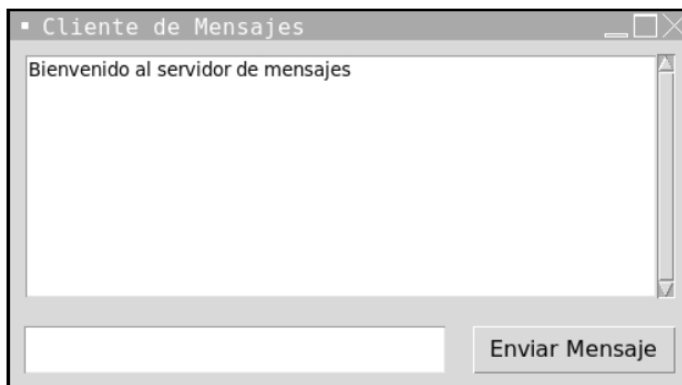
Implementar una solución software, a elección del grupo, que integre procesos/hilos que se comuniquen y sincronicen entre sí. Puede apegarse a lo visto en las partes 1,2 y 3 del TP1, utilizando los lenguajes vistos y utilizando colab o pueden proponer ustedes un lenguaje, framework, etc. A continuación se citan los trabajos de cursadas previas como para que tener una mejor idea, pero desde ya el límite/alcance y el tema lo define el grupo.

A continuación se listan a modo de ejemplo los trabajos de anteriores cursadas:

2023 – Segundo Cuatrimestre

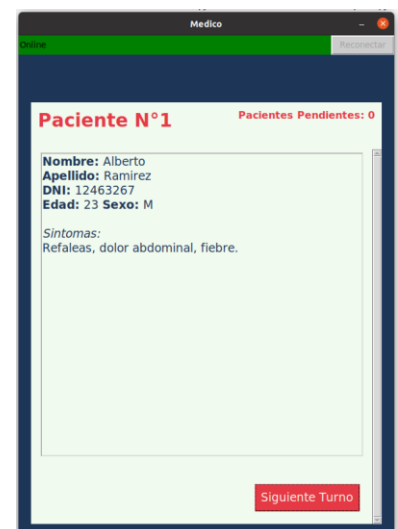
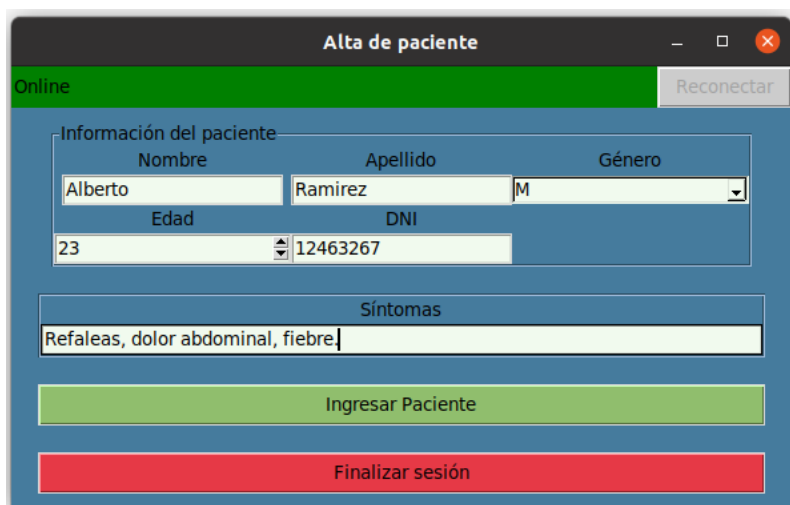
Sistema de Mensajería Segura con Tkinter y Encriptación

Este proyecto ofrece una solución para la comunicación entre clientes y servidores mediante una interfaz gráfica desarrollada con la biblioteca **tkinter** de Python. La implementación utiliza cifrado de extremo a extremo utilizando la biblioteca **cryptography** para garantizar la confidencialidad de los mensajes transmitidos.



Recepción de pacientes de un centro médico

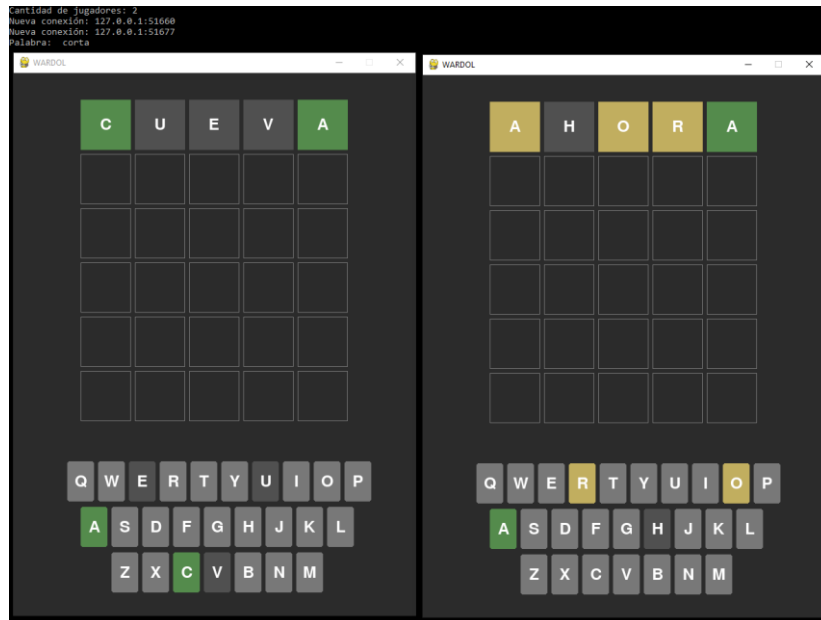
Esta solución de software se enfoca en la recepción de pacientes en un centro médico. Donde encontramos dos actores principales que son el recepcionista y el médico mismamente. En el cual, el primero se encargará de dar de alta a los nuevos pacientes que vayan llegando al establecimiento y el médico se encargará de llamarlos.



Utilizaron Python y Tkinter para las gráficas.

Wordle 2

El proyecto consiste en recrear el popular videojuego "Wordle". El objetivo de este juego es encontrar una palabra determinada en seis intentos. El juego puede ser de un jugador o multijugador, donde varios usuarios compiten para ser el primero en adivinar la palabra. Para desarrollar el mismo se utiliza una arquitectura de cliente-servidor.

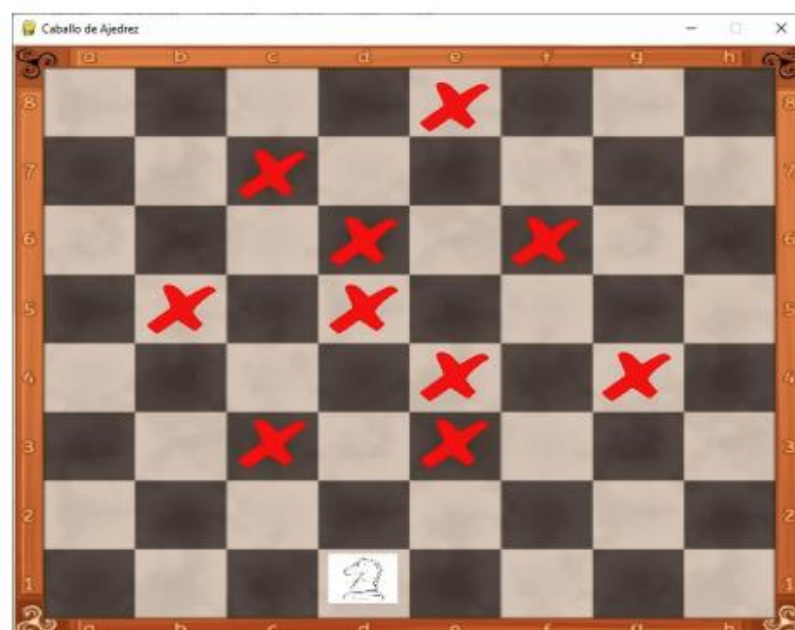


Utilizaron Go, Python, Pygame y tkinter.

Proyecto: Caballo de Ajedrez

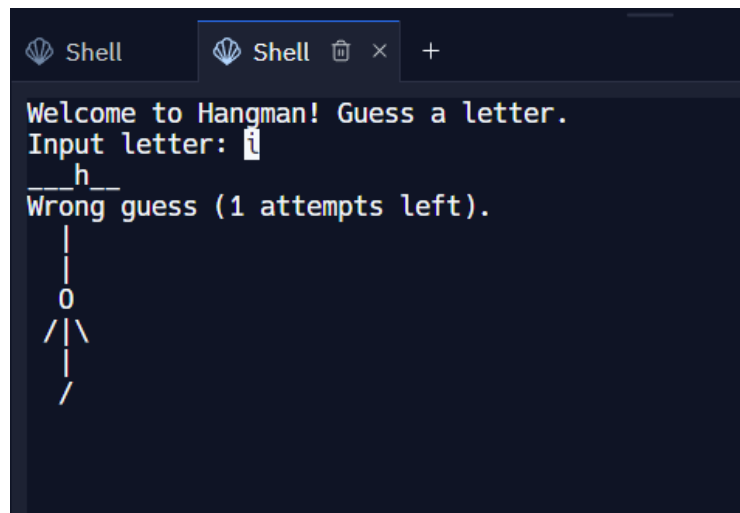
Se trata de un juego basado en el Ajedrez y, una de sus piezas, el caballo. El juego consiste en intentar recorrer todas las casillas de un tablero de Ajedrez con un caballo, que respeta el movimiento del juego en el cual se basa, y sin repetir casillas.

Se utilizó el lenguaje de programación **Python**, y el framework **Pygame** para renderizar la gráfica del juego y detectar los eventos que acciona el jugador.



Hangman

Este software implementa un juego de "Ahorcado" (Hangman) en un entorno cliente-servidor. Permite a múltiples jugadores conectarse a un servidor y jugar el juego individualmente. El servidor selecciona una palabra al azar que los jugadores deben adivinar y rastrea el progreso del juego para cada cliente. El juego se ejecuta en una interfaz de texto a través de la biblioteca "curses" en el cliente.



Utilizaron Python y Curses para el dibujado en pantalla.

2023 – Primer Cuatrimestre

MightyPass

Herramienta para evaluar la fortaleza de una contraseña y ofrecer recomendaciones para mejorar su seguridad. **Implementado en Python, utiliza hilos, semáforos y señales. Se ejecuta desde consola pero se puede integrar a una interfaz.**

API para realizar diversas tareas

Proporciona una interfaz de usuario con un combobox donde se elige una tarea a realizar, finalizada la misma se retorna un ID para poder obtener los resultados, como tareas de ejemplo se implementaron; traducir un texto a braille, generar un CUIL y encriptar una contraseña. **Implementado en Java 17, con el framework Springboot, base de datos H2, aplica los patrones de diseño Singleton, Strategy y Builder. También se utilizó Maven para construir y gestionar el proyecto.**

Ta-Te-Ti

Se implementa el típico juego del ta-te-ti donde se puede jugar de una a cinco partidas contra la CPU. Se utilizan **dos procesos pesados emparentados** (El padre controla la lógica y el hijo genera las estadísticas, victorias, empates, etc). Utilizan **threads** para mostrar los archivos de estadísticas **sincronizados con un Lock**. La **comunicación entre padre e hijo es con un pipe**. **Implementado en python y probado en colab**.

Encriptación de una carpeta mediante una clave

Esta solución de software encripta/desencripta carpetas utilizando el **cifrado AES**. Implementado en Python, comunican dos procesos no emparentados (Servidor y Cliente) que se comunican mediante dos FIFOs. También se generan threads, uno por cada subcarpeta de la carpeta a cifrar, los mismos se sincronizan con un mutex ya que acceden a un archivo común.

BINGO

Esta solución implementa un bingo que se juega en modo consola entre varios jugadores. **Implementado en Python y probado en colab**, utiliza dos programas (Servidor y Cliente) el servidor se levanta y queda a la espera de que se conecten 5 clientes, una vez conectados los clientes, el juego comienza y el servidor va generando los números. Se utilizaron sockets, fifos, memoria compartida y semáforos.

Carrito de compras

Utiliza principalmente C# y un mínimo de powershell y python. Utiliza Azure Service Fabric, visual studio y .NET 6.0 (System.Threading)