**Fundamentos de Informática**

TRABAJO PRÁCTICO

INTEGRANTES:

* Lozano Marianela - 1221723
* Moneta Martin - 1222398
* Mottura Moreno Diana Natalia - 1219228
* Onetto Hector Martin - 1222380

# PROFESOR:

* MAQUIEIRA GUILLERMO RAMIRO

Cuatrimestre: 1 – Año: 2025

Logotipo, nombre de la empresa

Descripción generada automáticamente

# *Objetivo del Trabajo*

El objetivo principal fue desarrollar un juego interactivo en Python que permita al usuario adivinar un número secreto de 4 cifras. El programa debía:

* Generar un número aleatorio de 4 cifras.
* Informar si el número ingresado es mayor o menor que el secreto.
* Permitir salir del juego con -1.
* Registrar los intentos al acertar y guardar el puntaje si es mejor que los anteriores.
* Almacenar y mostrar un ranking con los 5 mejores resultados.
* Preguntar si el usuario desea jugar de nuevo.

Más allá de la consigna, buscamos aplicar los conocimientos aprendidos en clase, compartir ideas entre compañeros y lograr una solución limpia, funcional y lo más prolija posible en equipo.

# *Planificación y Estrategia de Desarrollo*

Nuestra forma de abordar este trabajo fue bastante natural: empezamos resolviendo los puntos más simples que nos permitieran entender cómo funcionaba el juego, como la generación del número secreto y el control de intentos. Eso nos ayudó a visualizar el flujo básico del programa y ganar confianza en los primeros pasos.

Una vez que teníamos funcionando el juego en su versión más básica, fuimos agregando funcionalidades, como la validación de datos y el uso de listas para manejar los puntajes. En esa etapa también detectamos errores y los fuimos corrigiendo en conjunto, muchas veces debatiendo sobre cuál era la mejor forma de plantear cada función.

Al principio, todo el código estaba en una sola función principal, bastante larga. Pero a partir del seguimiento del profesor, reorganizamos el trabajo en funciones más pequeñas, con el objetivo de que el código quedará más claro, ordenado y fácil de mantener. Esa fue una gran mejora, ya que nos ayudó a entender mejor el propósito de cada parte del programa.

Finalmente, en el último tramo del trabajo, buscamos sumar algo más que lo básico. Decidimos usar los módulos json y pathlib para guardar los puntajes de forma persistente. Fue un desafío que implicó investigar un poco más y aprender sobre archivos, rutas y estructuras de datos, pero nos gustó mucho haber podido dar ese paso adicional.

En resumen, nuestra estrategia fue avanzar por etapas:

* Empezar con lo más simple para entender el juego.
* Incorporar validaciones y listas para mejorar la lógica.
* Modularizar el código por sugerencia del profesor.
* Agregar guardado de puntajes con archivos para completar el trabajo con algo más avanzado.

# *Diagrama General del Flujo*

1. Iniciar el juego
2. Generar el número secreto
3. Solicitar número al usuario
4. Indicar si el número ingresado es mayor o menor que el secreto
5. Repetir el proceso hasta que el usuario acierte o ingrese -1
6. Si el usuario acierta:  
   * Verificar si mejora el puntaje
   * Pedir el DNI si corresponde
   * Guardar el nuevo puntaje
   * Mostrar el ranking de los mejores
7. Preguntar si desea jugar otra vez
8. Reiniciar o finalizar el juego según la elección

Este esquema fue una guía constante para asegurarnos de que cada función cumpliera con su rol dentro del circuito del juego

# *Descripción de las Funciones*

El programa fue organizado en funciones específicas para facilitar la lectura, mantenimiento y reutilización del código. A continuación, se describen las principales funciones utilizadas:

* ***iniciar\_juego()*** Controla el flujo general del juego: genera el número secreto, solicita intentos al usuario, evalúa si se acierta o se abandona, y al finalizar gestiona el registro de puntajes y el reinicio del juego.
* ***solicitar\_numero()*** Solicita un número válido al usuario (de cuatro cifras o -1 para salir). Incluye validaciones para asegurar que se cumpla el formato requerido.
* ***solicitar\_dni()*** Pide el número de documento al usuario en caso de haber logrado un buen puntaje. Solo se acepta un valor numérico de 8 cifras.
* ***comparar\_numeros()*** Compara el número ingresado con el valor secreto. Informa si es mayor, menor, si el usuario gana o si decide salir del juego.
* ***mostrar\_puntajes()*** Muestra el ranking de los cinco mejores puntajes, detallando el DNI y los intentos correspondientes.
* ***ordenar\_puntajes()*** Evalúa si el intento actual es mejor que alguno de los ya registrados. Si corresponde, lo agrega al ranking, lo ordena y guarda los cinco mejores resultados.
* ***guardar\_nuevo\_puntaje()*** Registra el nuevo intento exitoso junto al DNI del usuario.
* ***get\_ranking()*** Lee el archivo JSON con los puntajes registrados. Si el archivo no existe, lo crea vacío para su posterior uso.
* ***save\_ranking()*** Guarda el ranking actualizado en el archivo mejores\_puntajes.json, manteniendo la persistencia entre partidas.
* ***jugar\_de\_nuevo()*** Pregunta al usuario si desea iniciar una nueva partida. Si la respuesta es afirmativa, reinicia el juego

# *Validaciones Realizadas*

* El número ingresado por el usuario debe ser de 4 cifras (1000 a 9999).
* El número de documento (DNI) ingresado debe ser de 8 cifras.
* Control de ingreso de valores no válidos o fuera de rango.
* Permite abandonar el juego ingresando `-1`.
* Controla correctamente los intentos realizados.

# *Manejo de Archivos*

Para conservar la información de los mejores puntajes, utilizamos la biblioteca **pathlib** para manejar archivos y rutas de forma segura y sencilla, evitando problemas de compatibilidad entre sistemas operativos.

El archivo mejores\_puntajes.json se utiliza para almacenar de manera estructurada los datos de los puntajes y los documentos de los usuarios. Si el archivo no existe al iniciar el programa, se crea automáticamente vacío, garantizando que el juego pueda funcionar sin interrupciones desde la primera ejecución.

Este enfoque permite que la información de los mejores resultados persista entre sesiones, ofreciendo al usuario la posibilidad de superar marcas anteriores y ver un ranking actualizado en cada partida.

# *Experiencia del Desarrollo*

En lo grupal, consideramos que este trabajo fue muy útil para afianzar los conocimientos que veníamos adquiriendo en la materia. Nos permitió integrar de manera concreta gran parte de los temas vistos, como estructuras de control, funciones, validaciones y manejo de archivos.

El grupo estuvo formado por personas con distintos niveles de experiencia. Esta diversidad nos permitió complementarnos bien y trabajar con una dinámica colaborativa, en la que cada uno aportó desde su lugar y se generaron aprendizajes compartidos.

También nos enfrentamos a herramientas nuevas para todos, como el módulo json y el uso de pathlib, que aprendimos a incorporar en el proceso mismo del desarrollo.

En resumen, fue un trabajo desafiante pero muy satisfactorio, tanto por lo que logramos técnicamente como por el modo en que lo llevamos adelante como equipo