**SISTEMA DE GESTIÓN DE HISTORIAS CLÍNICAS**

Implementar un sistema que se encargue del almacenamiento y carga de historias clínicas. Estás serán simuladas como archivos de texto .txt que se cargarán cuando el sistema sea encendido. El sistema entonces constará de un menú que en un inicio tendrá tres opciones: (1) Ingresar historia clínica, (2) Observar historia clínica y (3) Salir del sistema. La consulta de historias clínicas se puede realizar ingresando el id asociado a la historia clínica.

Para simplificación de la historia clínica los campos que esta contiene se sugieren que sean: Nombre del paciente, tipo de documento, número de identificación, id de la historia clínica (Este se debe generar automáticamente en el sistema, pero debe quedar registrada dentro del objeto asociado) y grupo sanguíneo. La consulta de la historia clínica se debe realizar con el id, no con el documento de identidad del paciente.

**Solución:**

La solución de lo planteado en este ejercicio es muy parecida a la que se implementa en el ejercicio 1. Esto con respecto al sistema y el manejo de sus menús. Para el manejo de las historias clínicas se generó una clase que captura la información que determina a una, su implementación se muestra a continuación.

|  |  |
| --- | --- |
|  |  |

Cada atributo permite determinar una historia clínica, y cada método asociado a un atributo permite tanto acceder a su contenido como modificarlo. Además, se implementa un método especial \_\_str\_\_, que se emplea con la misma finalidad que en el ejercicio 1.

Con respecto a los atributos usados en el sistema estos son:



|  |  |
| --- | --- |
| self.\_\_baseDeDatos | Lista que se encarga de almacenar las historias clínicas ingresadas dentro del sistema. |
| self.\_\_estado | Variable que ayuda a la gestión del menú. |
| self.\_\_historiaClinica | Variable que captura la historia clínica con la que se esté trabajando. |
| self.\_\_totalHC | Variable que lleva el conteo del total de historias clínicas registradas. Esta también sirve como el Id de cada historia. |
| self.\_\_msj1 | Mensaje del menú. |



El anterior código es el método que está señalado en amarillo en el método constructor del sistema. Este es el encargado de inicializar la base de datos del sistema. La lógica de este es la siguiente:

1. Se recorre por medio de un for el nombre de los archivos .txt presentes en el directorio actual. Para acceder a todos los nombres del directorio actual se emplea os.listdir(). Luego, por medio de la función generadora se crea una lista que tendrá todos los nombres de archivos que contengan la extensión ‘.txt’; si no, se almacenará el valor de None. Esto último se realiza en la sentencia: st if '.txt' in st else None. Para finalizar, se usa la función filter para eliminar los valores None de la lista generada, y por último se convierte en lista esto último para poder ser recorrida.
2. Por medio de la estructura with se abre cada archivo para su lectura. De este, por medio del método read y split se generan los elementos que serán empleados para generar la historia clínica a almacenar. Cuando se ejecuta el método read sobre el archivo, este retorna un string que contiene toda la información presente en el archivo abierto; luego, se genera una lista con toda la información ubicada en una posición de esta. Esto se genera por medio del método split, el cual recibe como argumento el carácter que se empleará para separar la información. En este caso la información siempre es almacenada con la siguiente estructura: Nombre;TipoDeDocumento;NumeroIdentificacion;GrupoSanguineo. Debido a lo anterior, el carácter empleado para la separación de cada información es “;”.
3. Luego se genera el objeto historiaClinica y se personifica en función de los valores cargados.
4. Por último, este se almacena y se aumenta el contador de historias clínicas en el sistema, valor que también funciona como id de la historia clínica correspondiente.

Para explicar la funcionalidad del sistema se mostrará una línea del tiempo donde se resalta y explica qué está sucediendo. Esto se hará con las diferentes opciones del menú general:

**Ingresar Historia clínica:**

****

La lógica empleada es similar a la usada en el Ejercicio 1. Cuando se inicia el sistema este ingresa en el While infinito, en este se entra por defecto en la opción del estado en 0, donde se muestra al usuario el menú y este decide entonces qué tarea desea emplear. Para realizar el ingreso en el sistema se usa el valor de 1, debido a que se está ejecutando un While infinito cuando se modifica el valor del estado a 1, lo siguiente que se ejecuta es el método ingresarHC, ya que ingresa en el debido elif.

En este método se implementa lo siguiente:

|  |  |
| --- | --- |
|  |  |
|  | |

El método ingresarHC llama a otros dos métodos de la clase sistema. Inicialmente se llama al método generarHC, el cual se encarga de pedirle al usuario los datos que se le asignarán a la historia clínica. La historia clínica generada se almacena en el atributo historiaClinica. Luego, este objeto es almacenado en la base de datos del sistema y se actualiza el contador de historias clínicas. Lo siguiente que se realiza es llamar al método almacenarTXT, el cual se encarga de generar el archivo .txt donde quedará guardada la información de la historia clínica generada. En este método se utiliza la estructura with que genera y abre el archivo que se le pone el nombre: “historiaNum\_{# de historia clínica}”. Para decirle a Python que este archivo será creado se emplea el segundo atributo ‘w+’. Luego, se genera el string que almacenará los datos, para esto se usa un f-string. Para finalizar este string se escribe en el archivo abierto. Con esto se finaliza el almacenamiento de los datos asociados a una historia clínica.

Al inicio del método ingresarHC se hace que la variable estado vuelva a 0, esto con el fin de que al finalizar la ejecución de este método, se regrese al estado inicial del sistema donde se le muestra el menú al usuario.

**Ver Historia clínica:**

|  |  |
| --- | --- |
|  |  |

La lógica empleada para ver las historias clínicas es muy simple. En el método verHC, el cual es llamado en el While infinito, inicialmente se verifica si el Id corresponde con alguna historia clínica dentro de la base de datos. Esto se realiza por medio del método buscarHC, el cual recibe el Id ingresado por el usuario y por medio de un for recorre toda la base de datos verificando si el Id corresponde con alguna historia clínica. Si la búsqueda es exitosa, la historia clínica encontrada se almacena en el atributo historiaClinica y luego se retorna True; si no se retorna False. Luego de que esto suceda se evalúa el condicional presente en verHC. Si la historia clínica se encuentra dentro de la base de datos esta se imprime en pantalla y se reinicializa el sistema para regresar a mostrar el msj1(Mostrar el menú). Si no, se le menciona al usuario que el Id no corresponde con ninguna historia clínica, y se reinicializa el valor de estado para regresar al menú.

**Salir del sistema**

Como último, para salir del sistema solo se llama la palabra clave break, la cual hace que se detenga el While infinito.