La práctica consiste en crear un programa que permita extraer unos valores de un archivo .txt y crear pacientes con diferentes atributos para posteriormente distribuirlos dependiendo de sus características en una de las 4 diferentes colas de espera que dan acceso a la cola. Este proceso dura hasta que todos los pacientes cargados en el .txt hayan sido atendidos.

- ***Instalación y Mecanismo de Ejecución***:

Para poder ejecutar el programa es necesario disponer de un compilador de Python, así como tener instaladas las librerías pandas (para la creación y gestión de DataFrames) y sys (para leer el .txt con los datos de los pacientes),

El programa se inicia descomprimiendo el fichero .zip y ejecutando el archivo main.py. Este accede a la ruta (indicada en el código) donde se encuentra el archivo .txt y carga los parámetros indicados. Para su correcta lectura, cada paciente del txt debe seguir la siguiente estructura (los datos de cada instancia de la clase paciente deben estar agrupados en una misma línea y cada campo separado por un espacio en blanco):

Atributo	**Tipo de Dato**	**Descripción**
ID del Paciente	str	Código alfanumérico único del paciente.
Tipo de Consulta	str	Puede ser "general" o "specialist".
Urgencia	bool	"True" si es urgente, "False" en caso contrario.
Tiempo Estimado	int	Minutos que estará en consulta.

Una vez leídos estos parámetros, el main ejecuta la función cargar_pacientes(), que carga y genera a los pacientes desde el txt para posteriormente ejecutar el bucle principal, que llama a las funciones y objetos de los archivos “gestor”, “clase_paciente” y “colas” para el correcto funcionamiento del programa.

Nicolás García Iglesias – nicolas.garcia.iglesias@udc.es
Manuel Omil Barreiro – manuel.omil.barreiro@udc.es

- ***Fases de desarrollo.***

El programa trata con objetos creados a partir de la clase pacientes (definida en el archivo clase_paciente). Estos pacientes presentan una serie de atributos (que se cargan a la hora de leer el archivo .txt) y un diccionario que alberga los tiempos particulares de cada paciente para su posterior uso en el apartado de estadísticas.

Estos pacientes se almacenan en instancias de la clase "cola", provenientes del archivo cedido por el profesorado "colas". Estas colas siguen los principios de las colas circulares vistas en las clases teóricas y cuentan con los métodos típicos tales como dequeue(), enqueue(), is_empty()… etc.

Por último, tanto las instancias de Cola, como la de Paciente son administradas dentro de la clase Gestor; esta clase se encarga de almacenar las 5 colas necesarias (Admisión, y otras 4 para cada tipo de consulta y urgencia). La clase Gestor cuenta con métodos para clasificar y pasar a los pacientes de una cola a otra y de ahí a sus respectivas consultas. También cuenta con un sistema de priorización para aquellos pacientes que esperen mas de 7 horas en una cola sin ser atendidos, almacenándolos y pasándolos la siguiente vez que necesiten ser tratados a una cola de urgencia prioritaria.

El archivo main es el encargado de ejecutar el programa: Se encarga de leer y cargar los pacientes desde el .txt, de iniciar y parar la simulación, y de mostrar las estadísticas generadas (a través de la librería pandas).

Fases de la Simulación (Se ejecutan cada turno)

1) *Distribución de Pacientes*

Una vez cargados todos los pacientes en la cola de Admisión, la función distribuir_pacientes() se encarga de, cada 3 turnos, escoger al primer paciente de la cola de Admisión (siguiendo un orden FIFO) y dependiendo de los valores que tomen sus properties, "consulta" y "urgencia", lo mueve a la cola de espera de su consulta correspondiente.

2) *Retirar de Consulta:*

Esta parte consiste en comprobar para todas las consultas con pacientes siendo atendidos, si el paciente ya ha cumplido el tiempo estimado en consulta. Si esta condición se cumple, entonces los datos del paciente se guardan para su uso posterior en una lista antes de descartarlo.

3) *Pasar a Consulta:*

Durante esta fase el gestor se encarga de comprobar si las consultas están vacías y, de estarlo, pasa al siguiente paciente de la cola de espera .

Sistema de Priorización

El sistema implementa un sistema de compensación para aquellos pacientes que lleven esperando más de 7 horas antes de pasar a consulta. Cuando esta condición se cumple, el ID del paciente se guarda en una lista y la próxima vez que el paciente vuelva al hospital, el gestor comprobará que tiene acceso prioritario y lo pasará a la cola de urgencia para que sea tratado ahí

- ***Estadísticas***

Al final de la simulación también se calculan una serie de estadísticas mediante el uso de la librería pandas. Con esta librería se genera una serie de DataFrames a partir de los datos de los pacientes almacenados en la lista "almacenados". Dichos DataFrames agrupan a los pacientes y muestran el número medio de priorizaciones aplicadas a pacientes (no dadas) dependiendo del tipo de consulta. El segundo DataFrame devuelve el tiempo medio de espera para cada tipo de consulta diferente.

- ***Errores y Mejoras***

El principal problema de esta práctica ha sido intentar aplicar el uso de diccionarios para guardar datos y relacionar cada cola con su consulta; pese a que la idea principal era escribir menos líneas de código, también lo ha hecho más contraintuitivo y difícil de entender, además de presentar problemas a la hora de trabajar con el pandas.

IMPORTANTE:

En el archivo .txt original los pacientes vienen catalogados como "priority" o "no priority". En este caso es necesario modificar este campo por los valores True or False antes de ejecutar el código o dará error. También recomendamos usar el archivo txt provisto en el .zip. Este cambio ha sido autorizado por el profesorado.
