**Link de git**

**Requerimientos funcionales:**

[**https://github.com/sebastianvilla17/Usser\_Support.git**](https://github.com/sebastianvilla17/Usser_Support.git)

El sistema debe estar en la capacidad de:

1. Registrar un usuario con los siguientes caracteres: documento de identidad, número de documento, nombres, apellidos, teléfono y dirección y a la vez validar que estos datos sean diligenciados y tener en cuenta que no se puede registrar un usuario que ya se encuentra en el sistema, si el usuario se registra y se agrega correctamente se le asigna un turno codificado con una letra y dos números.
2. Buscar un usuario con su número de documento de identidad, entrando a la lista y confirmando que se encuentre registrado, de lo contrario se lanza una excepción. Si el usuario se encuentra en la lista se le asigna un turno codificado y se imprime el teléfono y el nombre del usuario junto con el código del turno. Para que este requerimiento se ejecute es necesario que hallan previamente creados los usuarios.
3. Gestionar tipos de turnos con duraciones preestablecidas en minutos, es decir recibe una fecha de fin del turno por lo tanto se crea un turno con su nombre, hora de inicio y hora de fin, asignado a un turno que tiene un usuario
4. Actualizar la fecha y hora del sistema.
   1. La actualización del sistema puede ser de dos maneras:
      1. Manualmente, a través de valores indicados por el usuario.
      2. Utilizando la fecha y hora actual del sistema de cómputo.
      3. También se muestra en el menú una opción para conocer la fecha y la hora y cada vez que se inicia el menú de nuevo se imprime esta.
5. Generar un reporte con todos los turnos que una persona ha solicitado alguna vez, indicando el código del turno, si ya fue atendido y si la persona estaba presente cuando fue llamada para ser atendida, recibiendo como parámetro el documento de la persona y retornando el informe con la opción de verlo en pantalla o guardarlo en un archivo
6. Suspender a una persona no ha estado presente en los últimos dos turnos, de tal manera que no pueda solicitar nuevos turnos por dos días, esto realiza comprobando el estado de los turnos anteriores.
7. Generar aleatoriamente personas registradas en el sistema. La cantidad de personas a generar debe ser indicado por el usuario. Esto se hace leyendo un archivo de texto con nombres y apellidos aleatorios, pera registrarlos y agregarlos al sistema
8. Generar aleatoriamente turnos asociados a personas previamente registradas en el sistema. El usuario debe indicar cuantos días de turnos se generarán. Y siempre se empieza la generación a partir del día actual. El usuario debe indicar también cuántos turnos se generarán por cada día. Para generar los turnos se toman usuarios previamente creados y se les asigna un turno con una duración y nombre aleatorio
9. Atender turnos automáticamente, de manera que se presenta en el menú la opción atender turnos y esta atiende todos los turnos que se han registrado hasta la hora, cambiando su estado ha atendido y aleatoriamente coloca si el usuario estaba o no

**Requerimientos no Funcionales:**

El sistema permite:

* 1. Generar un reporte de usuarios ordenados por numero de documento, implementando el compareTo y el método de ordenamiento burbuja, tomando la lista de usuarios ordenándola y mostrando por pantalla los usuarios ordenados
  2. Generar un reporte de usuarios ordenados por nombre de usuario, implementando el compareTo y el método de ordenamiento selección, tomando la lista de usuarios ordenándola y mostrando por pantalla los usuarios ordenados
  3. Generar un reporte de usuarios ordenados por Apellido, implementando el compareTo y el método de ordenamiento inserccion, tomando la lista de usuarios ordenándola y mostrando por pantalla los usuarios ordenados
  4. ser serializable de tal forma que la información del sistema se guarda y puede volver a ser cargada al momento de iniciar el programa. (como comentario al inicio del programa se puede presentar una excepción que no pude atrapar por lo tanto se recomienda parar cerrar y volver a ejecutar de esta manera funciona.
  5. El sistema utiliza la clase math random para generar los datos necesarios para la creación de usuarios y turnos
  6. Se utilizan búsquedas secuenciales para los métodos que la necesiten(searchDocument, repeat, report…etc)

Cometarios adicionales:

1. El diagrama de clases se encuentra en la carpeta del proyecto llamada “Diagrama de clases”,en su ejecutable con la app recomendada por el docente
2. Los contratos de los métodos no están completos
3. La prueba unitaria de agregar usuario si funciona pero como se le coloca a imprimir el tiempo que tarda , esta se demora diferente en cada uno