**REQUERIMIENTOS FUNCIONALES**

**REQUERIMIENTO 1:** Agregar un nuevo usuario con tipo de documento de identidad, número del documento, nombres, apellidos, teléfono y dirección, de entre los cuales es obligatorio ingresar tipo de documento, número de documento, nombres y apellidos, el teléfono y correo se pueden dejar sin rellenar. Hay que revisar que el número de documento no tenga letras ni símbolos, que los nombres y apellidos no contengan números ni símbolos, si el usuario ya existe en el programa, en cuyo caso no se le dejará agregarse por segunda vez, también se debe revisar si ya hay usuarios en el sistema o si no se ha ingresado ninguno.

**REQUERIMIENTO 2:** buscar un usuario con base en su número de identidad. Se debe revisar si el usuario ya existe o si no ha sido ingresado a programa, además de si ya hay usuarios en el programa o si no hay ninguno.

**REQUERIMIENTO 3:** asignar un turno a un usuario, este turno consta de una letra desde la **A** hasta la **Z** y dos números, desde **00** hasta **99**, una vez se llegue al límite numérico con una letra se procederá a usar la siguiente, por ejemplo, luego de **A99** se procederá a usar **B00**. Aquí hay que revisar si el usuario ya tiene un tuno activo, en cuyo caso no se le dará otro turno y si el usuario no tiene un turno activo se le dará un turno, además hay que revisar que el programa este trabajando correctamente y si esté aumentando entre turnos, por ejemplo, que pase de **A00** a **A01**, también hay que tener en cuenta que una vez que se llegue al turno **Z99**, se inicie la cuenta otra vez desde **A00**. Además, desde se le asigna un tipo de turno y su duración.

**REQUERIMIENTO 4:** reiniciar los turnos. Cuando el empleado lo considere necesario, podrá usar esta opción para reiniciar los turnos. Esto es, que no importa por cual turno vaya, una vez usada esta opción los turnos iniciaran nuevamente desde **A00**.

**REQUERIMIENTO 5:** crear un usuario al azar. Este requerimiento crea un usuario con un documento, tipo de documento, nombres, apellidos, teléfono y dirección al azar, usando para los nombres y los apellidos archivos de texto de los cuales usaran nombres y apellidos al azar, mientras que para el documento se creara un numero al azar desde 1000000000 hasta 9999999999 y para el tipo de este se obtendrá un número al azar y dependiendo de ese número se escogerá un tipo u otro **(explicar cuáles números)**. Para el teléfono se hará algo similar, se generará un número al azar desde 3000000000 hasta 3999999999, esto debido a que se usará para teléfono celular. Finalmente, para la dirección se utilizará un arreglo constante de String para la primera parte, por ejemplo “carrera”, “calle”, “avenida” o “transversal”, y para el resto se generarán dos números al azar, uno desde 00 hasta 99 y el otro desde 0000 hasta 9999, los cuales se unirán usando un “#”, un ejemplo de cómo quedaría una dirección generada al azar seria: “carrera 25#4783”. Además de esto, el usuario creado deberá cumplir con los requisitos del **requerimiento 1**.

**REQUERIMIENTO 6:** generar turnos al azar. Se generarán tunos y se les asignarán a usuarios al azar, a la hora se usar este requerimiento se deberá ingresar la cantidad de días para los cuales se generarán los turnos y la cantidad de turnos por día, el tiempo total que se demore en atender estos turnos no podrá ser mayor al tiempo total de ese día, es decir, 24 horas. Además, los turnos generados cumplirán con las condiciones del requerimiento 3.

**REQUERIMIENTO 7:** llevar una fecha y hora en el programa. El programa deberá llevar internamente una fecha y hora.

**REQUERIMIENTO 8:** mostrar fecha y hora. Muestra en pantalla el año, mes, día, hora, minuto y segundo actuales del programa.

**REQUERIMIENTO 9:** actualizar fecha y hora. Permite la actualización de la fecha y hora del programa de dos maneras distintas. La primera manera es que el usuario ingrese la nueva fecha y hora, mientras que la otra manera es que se actualicen automáticamente con la fecha del computador. Además, las nuevas fecha y hora no podrán ser menores a la actual.

**REQUERIMIENTO 10:** atender todos los turnos hasta la fecha actual. Se atenderán todos los turnos desde la última fecha en la que se atendieron (en caso de ser la primera vez entonces será desde que se inició el programa), hasta la fecha actual del programa, a la hora de atenderlos se elegirá al azar que turnos fueron atendidos correctamente y en que casos no se encontraba el usuario.

**REQUERIMIENTO 11:** crear un tipo de turno. Se crea un tipo de turno el cual tiene un nombre y un tiempo estimado de lo que se demora en ser tendido, el nombre no se podrá repetir y el tempo será en minutos.

**REQUERIMIENTO 12:** generar un reporte con todos los turnos que una persona ha solicitado alguna vez, indicando el turno, si ya fue atendido y si la persona estaba presente cuando fue llamada para ser atendida.

**REQUERIMIENTO 12:** generar un reporte con todas las personas que han llegado a tener un turno indicado por el usuario.

**REQUERIMIENTO 13:** suspender a un usuario que no haya estado presente en los últimos dos turnos y que no pueda volver a pedir turnos por dos días.

**REQUERIMINETO 14:** mostrar el tiempo que una acción demora en ejecutarse. cada que el usuario lleve a cabo alguna acción, se mostrara en consola el tiempo en milisegundos que esta de demoró en realizarse.

**REQUERIMINETO 15:** persistir el estado actual del sistema. Guardar el estado actual del sistema, para que cada vez que se inicie, cargar los usuarios, turnos y todo lo demás que ya se haya creado anteriormente.

**REQUERIMIENTOS NO FUNCIONALES**

**REQUERIMIENTO 1:** el programa deberá realizar la persistencia mediante serialización.

**REQUERIMIENTO 2:** el programa usa ordenamiento de sort cuando los usuarios que han tenido un turno, también usa búsqueda binaria en un método que es llamado por este.

**REQUERIMIENTO 2:** el programa usa ordenamiento de sort cuando va a suspender a un usuario.

Imagen que contiene texto, captura de pantalla

Descripción generada automáticamente

**GitHub link:** [**https://github.com/MarcoFidelVasquezRivera/turn-control-system.git**](https://github.com/MarcoFidelVasquezRivera/turn-control-system.git)