Laboratorio 2

**Requerimientos funcionales**

El programa esta en la capacidad de:

* Agregar un nuevo usuario en caso de que no esté registrado previamente con su nombre, apellido, tipo de documento de identidad, numero de documento, teléfono y dirección. Los campos obligatorios para que agregar un nuevo usurario son: nombre, apellido, tipo de documento y número de identificación.
* Registrar un tipo de turno a un usuario que ya este registrado en el sistema, de lo contrario no se le asignara ningún turno, este turno cuenta con el tiempo en el que se demora en atenderse y el nombre del tipo.
* Mostrar al funcionario la fecha actual implementado: día, mes, año, hora, minuto, segundo. Asimismo, de actualizar la fecha en el sistema.
* Generar un reporte con los turnos de algún usuario que haya solicitado desde su primera asignación de turno, indicando el código del turno, si se atendió el turno y si la persona estaba presente en ese momento.
* Suspender a un usuario por no estar presente en el momento de atender el turno en dos ocasiones consecutivas, esta suspensión consiste en que el usuario no se le puede asignar un turno después de dos días.
* Generar un reporte con todos los usuarios que han tenido un turno en específico. Ejemplo (A01 o G89)
* Generar aleatoriamente turnos a personas ya registradas, indicando cuantos días de turnos se van a generar, cuantos turnos por día se generan, donde la asignación se hace desde el mismo día.
* Generar nuevos usuarios aleatoriamente con base a los que ya están registrados, indicando cuantos desea generar.

**Requerimientos no funcionales**

En programa debe incluir:

* La clase fecha y hora en el modelo
* Persistencia utilizando la serialización
* Un texto base con los nombres y otro con los apellidos de los usuarios donde a partir de este se generan los nuevos usuarios con la información ya expuesta en el sistema.
* Los reportes mostrados por consola o un archivo de texto o en su defecto ambos, es a elección del usuario.
* El tiempo con el que se demoró la acción a realizar.
* Al menos tres ordenamientos, uno de cada uno (burbuja, inserción, selección) en primitivos o en su defecto objetos.
* Dos búsquedas una secuencial y una binaria.
* Cinco ordenamientos utilizando el método sort de la clase Arrays o Collections, incluyendo las interfaces Comparable, Comparator como clase externa, Comparator como clase anónima, Comparator al revés del modelo de Collections y una clase Comparator ya implementada en el API de java.