**Especificación de requerimientos funcionales**

El sistema debe estar en la capacidad de…

**Req1. Agregar** un nuevo usuario con tipo de documento, número de documento, nombres, apellidos, teléfono y dirección, siendo los 4 primeros campos obligatorios. No se permitirá que se agreguen usuarios con el mismo número de documento.

**Req2. Registrar** un nuevo turno a un usuario dado su número de documento. No se permitirá registrar turnos a usuarios que tengan un turno activo pendiente. El primer turno disponible siempre será A00 y cada vez que el número llegue a 99, el siguiente turno será la siguiente letra seguida del número 00 y así sucesivamente.

**Req3. Atender** un turno, mientras que halla turnos pendientes por atender. En caso de que el usuario no esté presente en el momento que se atiende su turno, esto podrá ser indicado por quien usa el programa.

**Req4. Gestionar** tipos de turnos con duraciones preestablecidas en minutos. Todo tipo de turno tiene un nombre y su duración es de tipo float.

**Req5. Desplegar** visualmente la fecha y hora actual del sistema, indicando: año, mes, día, hora, minuto y segundo.

**Req6. Actualizar** la fecha y hora del sistema. La actualización puede ser de dos formas: Manualmente, a través de valores indicados por el usuario o Utilizando la fecha y hora actual del sistema de cómputo.

**Req7. 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.

**Req8. 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.

**Req9. Generar** un reporte con todas las personas que han solicitado un turno.

**Req10. Generar** aleatoriamente personas registradas en el sistema. La cantidad de personas a generar debe ser indicado por el usuario.

**Req11. Generar** aleatoriamente turnos asociados a personas previamente registradas en el sistema.

**Req12. Atender** turnos hasta la hora y fecha actuales del sistema, teniendo en cuenta el tiempo de espera entre la atención de turnos que es de 15 segundos.

**Req13. Guardar** sus datos a través de la serialización de sus objetos (persistencia).

**Req14. Cargar** los datos del modelo desde un archivo externo.

**Especificación de requerimientos no funcionales**

A la hora de generar aleatoriamente turnos asociados a personas previamente registradas en el sistema, el número de turnos no puede exceder al numero de usuarios registrados en el sistema.

Los reportes de usuarios individuales tienen terminación .report y pueden ser visualizados desde Eclipse.

A la hora de generar el reporte de todos lo usuarios que han solicitado un turno, los ordenamientos se llevan a cabo con algoritmos de ordenamiento de selección, burbuja, utilizando el método sort de la clase Arrays o de la clase Collections, utilizando las interfaces Comparable sobre alguna clase propia del modelo, Comparator como una clase externa, Comparator como una clase anónima, un Comparator al revés del método de Collections y un Comparador de una clase ya implementada.

Cuando se actualiza la hora o fecha del sistema, se procesan como atendidos, todos los turnos, y aleatoriamente se indica si dichas personas estaban presentes o no en el momento en que fueron llamadas.

Los datos utilizados para generar usuarios aleatorios fueron creados y descargados desde Mockaroo.

Cada vez que el usuario lleve a cabo una acción en el sistema, se debe indicar cuánto tiempo se demoró llevar a cabo dicha operación