**DEFINICON DE ALGUNOS MÉTODOS**

PRESENTADO POR:

Néstor Augusto Marmolejo 1088356715

Agustín Salazar Arenas 1225091225

PRESENTADO A:

Julio Cesar Chavarro Porras

PROGRAMACIÓN IV

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

INGENIERIA DE SISTEMAS Y COMPUTACIÓN

20 DE ABRIL DEL 2019- I

**Asesor -> + disponibilidadAsesor (Boolean): void**

Este método verifica el estado de un Asesor, si está activo puede atender Turnos.

**Asesor -> + guardarAsesor (RandomAccessFile): void**

Este método permite guardar en un archivo toda la información de un Asesor (id, nombre, nroCelular, estado, habilidadServicio).

**Asesor -> + recuperarAsesor (RandomAccessFile): void**

Este método permite recuperar en cualquier momento toda la información de un Asesor ya existente en el archivo.

**Turno -> + addTurno (int, String, Boolean, Usuario, Servicio): int \***

Este método guarda toda la información dada por el Usuario que le pide el sistema, y lo agrega a la lista de espera.

**Puesto -> + configurarPuesto (int, Servicio, Servicio, Servicio, Asesor): void**

Este método permite configurar los Puestos de Atención a Usuarios, con el número de Puesto de Atención, cada servicio que se puede prestar en ese Puesto de Atención y el Asesor que estará en ese Puesto.

**Asignacion -> + turnoLlamar (String, Puesto, Turno, Asesor): void**

Este método valida que turno se atenderá de la cola, teniendo como primordial las prioridades de cada cola.

**Asignacion -> + turnoLlamar (String, int, Turno, Asesor): void**

Este método escoge el turno a atender donde asigna a que puesto de atención debe pasar y con qué Asesor.

**Asignacion -> + historialAtendido (String, Puesto, Turno, Asesor): void**

Este método permite guardar y recuperar los detalles de un turno atendido (fecha, asesor que lo atendió, usuario, servicio que solicitó, puesto de atención, etc).

**Asignacion -> + listaColas(): void**

Este método permite listar los turnos que hay en cada cola de cada servicio.