



Departamento de TIC
Algoritmos y Programación II
Laboratorio de la Unidad 1

Objetivos

- OE2.1 Manipular archivos de texto para implementar requerimientos del cliente relacionados con persistencia.
- OE2.2 Leer entradas e imprimir salidas de programas que interactúan directamente con archivos de texto y no con usuarios finales.
- OE2.3 Hacer persistir el estado del modelo de solución del problema durante la ejecución de un programa y restaurarlo cuando se requiera usando la técnica de serialización.
- OE2.4 Implementar algoritmos clásicos de ordenamiento de datos en estructuras de datos lineales y aplicarlos en la solución de un problema.
- OE2.5 Implementar algoritmos clásicos de búsqueda de información en estructuras de datos lineales y aplicarlos en la solución de un problema.
- OE2.6 Hacer uso de las interfaces Comparable y Comparator para definir relaciones de orden total sobre objetos.
- OE2.7 Calcular el tiempo de ejecución de un algoritmo por medio de las operaciones de tiempo del sistema
- OE2.8 Implementar métodos que permitan generar muestras con datos aleatorios.

Preparación

- Lea cuidadosamente el enunciado, la documentación suministrada y cada uno de los puntos que debe desarrollar antes de empezar su desarrollo. Pregunte a su profesor cualquier duda respecto al enunciado o a los requerimientos funcionales que debe desarrollar.
- El trabajo debe ser realizado de forma **individual**.
- El trabajo será entregado en la fecha y hora establecida en Moodle.

Su entrega debe incluir los siguientes elementos

- Requerimientos Funcionales (donde se explica la funcionalidad que ve el usuario).
- Requerimientos No Funcionales (donde se detallan las restricciones internas de funcionamiento que el usuario del programa no puede ver, por ejemplo: que un ordenamiento es burbuja, que la persistencia es utilizando serialización, etc).
- Diagrama de Clase.
- Implementación en un proyecto de eclipse.
- Los requerimientos funcionales, no funcionales y el diagrama de clases deben presentarse de un documento o informe del laboratorio que además debe indicar donde se implementaron los algoritmos de ordenamiento, las interfaces de comparación, los ordenamientos utilizando las interfaces y los algoritmos de búsqueda.

Sobre el laboratorio anterior del sistema de turnos, usted debe adicionar las siguientes funcionalidades. El sistema debe estar en la capacidad de:

- Gestionar tipos de turnos con duraciones preestablecidas en minutos. Todo tipo de turno tiene un nombre y su duración es de tipo punto flotante pues la duración podría ser por ejemplo: 1 minuto y medio (1.5), 3 minutos y cuarto (3.25), etc.
- Asignar, desde su creación, tipo de turno y duración del turno.
 - La duración del turno estará basada en el tipo de turno seleccionado.
- La hora de atención se establecerá basado en que hay una sola persona atendiendo, y se espera que, la atención de todos los turnos ocurra en los tiempos establecidos en los tipos de turno, y que los cambios de turno tienen una demora de 15 segundos (0.25 segundos).
- Desplegar visualmente la fecha y hora actual del sistema, indicando: año, mes, día, hora, minuto y segundo.
- Actualizar la fecha y hora del sistema.
 - La actualización del sistema puede ser de dos maneras:
 - Manualmente, a través de valores indicados por el usuario.
 - Utilizando la fecha y hora actual del sistema de cómputo.
 - Aunque la fecha y hora del sistema hayan sido actualizados manualmente, dicha fecha y hora deben seguir corriendo internamente de forma normal, con base en los valores actualizados. Por ejemplo, si la fecha y hora fueron actualizados al 12 de Noviembre de 2021 a las 10:25 AM y luego transcurren siete minutos, entonces, la fecha y hora del sistema ahora es 12 de Noviembre de 2019 a las 10:32 AM.
 - Solo es posible actualizar la fecha y hora del sistema a una mayor.
 - Cuando ocurra una actualización de la fecha y hora del sistema, se deben procesar como atendidos, todos los turnos, y aleatoriamente indicar si dichas personas estaban presentes o no en el momento en que fueron llamadas.
- Generar un reporte con todas 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.
- 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.
- Generar un reporte con todas las personas que han llegado a tener un turno indicado por el usuario (por ejemplo: A07, o C58).
- Generar aleatoriamente personas registradas en el sistema. La cantidad de personas a generar debe ser indicado por el usuario.
- 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.

Condiciones y restricciones:

- Usted debe tener definida la clase fecha y hora en su modelo.
- El sistema debe ser persistente usando serializable.
- La generación aleatoria de personas debe tomar los valores de los nombres de varios archivos de texto base (uno para nombres, otro para apellidos) generado a partir de herramientas de generación de estos valores que existen en la web.
- Los reportes deben tener la posibilidad de ser visualizados en consola y también de ser enviados a un archivo de texto. El usuario debe elegir qué se debe hacer al generar el reporte.
- Cada vez que el usuario lleve a cabo una acción en el sistema, además de la respuesta final que arroje (por ejemplo: agregar una persona o turno, generar un reporte, generar personas aleatoriamente, etc) que diga cuánto tiempo se demoró llevar a cabo dicha operación.
- En alguna parte del desarrollo de los requerimientos anteriores, debe llevar a cabo al menos un ordenamiento utilizando **cada uno** de estos algoritmos:

• En alguna parte del desarrollo de los requerimientos anteriores, debe llevar a cabo al menos un ordenamiento utilizando cada uno de estos algoritmos: burbuja, selección e inserción, sobre primitivos o sobre objetos.

- En alguna parte del desarrollo de los requerimientos anteriores, debe llevar a cabo al menos dos búsquedas, una secuencial y una binaria.
- En alguna parte del desarrollo de los requerimientos anteriores, debe llevar a cabo al menos 5 ordenamientos 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 del API de Java como Integer o String (en este último no debe implementar el Comparator, solo ordenar un arreglo o lista de objetos de una clase de Java que ya viene con el Comparator implementado).

Recomendaciones y Aclaraciones

- La fecha y hora actual del sistema puede (y debería) mostrarse de dos formas:
 - Al inicio del menú, para que cada vez que el menú se despliegue, se muestre cuál es la fecha y hora actual del sistema de turnos.
 - Una opción del menú llamada X. Mostrar fecha y hora, que al ser elegida muestre la fecha y hora actual del sistema de turnos.
- La hora del sistema de turnos inicia siendo la misma hora del sistema (es decir, la hora del computador). Pero aunque el usuario del programa cambie la hora del sistema de turnos, la hora del computador siempre será el referente para conocer la hora del sistema de turnos. Por tanto, cuando se actualice la fecha y hora en el sistema de turnos, lo que se debería almacenar no es la nueva fecha y hora, sino la **diferencia** de la nueva fecha y hora respecto de la fecha y hora del computador.

Por ejemplo, si la hora actual del computador es 9:44 AM y la nueva hora que se quiere establecer es 10:12 AM, lo que se debería guardar es que hay una diferencia de 28 minutos.

De esta manera, en cualquier momento que el usuario solicite la fecha y hora actual, dicha fecha y hora actual del sistema de turnos será **calculada** con base en la hora actual del computador y la diferencia almacenada. Por ejemplo, si después del ejemplo anterior, pasa un tiempo, la hora del computador ahora es 11:07 AM y el usuario del programa elige la opción de mostrar hora actual del sistema de turnos. La hora del sistema sería calculada como 11:07 + **28 minutos** = 11:35 AM.

Tenga en cuenta que lo anterior, respecto de guardar las diferencias, también aplica para fechas.

- Ya no se requiere la opción atender turno para atender el siguiente turno (uno solo) y preguntar si la persona se encuentra o no en el momento para ser atendida. Esto debido a que los turnos ahora se atenderán de forma automática, teniendo en cuenta la duración del tipo de turno. La opción de atender turno anterior, deberá ser reemplazada por una opción de atender todos los turnos hasta la fecha y hora actual.

Por ejemplo, el sistema de turnos está siendo usado en una cafetería y creamos dos tipos de turnos con los siguientes nombres y duración: SERVIR ALMUERZO con duración de 2.5 y SERVIR REFRIGERIO con duración de 3.5. En el sistema son las 17:36 y en ese momento se crean los siguientes turnos:

Turno A00 de tipo SERVIR ALMUERZO solicitado a las 17:36.
Turno A01 de tipo SERVIR REFRIGERIO solicitado a las 17:36.
Turno A02 de tipo SERVIR ALMUERZO solicitado a las 17:36.
Turno A03 de tipo SERVIR ALMUERZO solicitado a las 17:36.
Turno A04 de tipo SERVIR REFRIGERIO solicitado a las 17:36.
Turno A05 de tipo SERVIR REFRIGERIO solicitado a las 17:36.

Pasa varios minutos y ahora son las **17:48**. Aunque ha pasado tiempo y los turnos debieron ser atendidos ya, el programa no ha hecho nada pues no se le ha dado ninguna orden. En ese momento, el usuario da la orden de atender todos los turnos hasta la fecha y hora actual. Lo que ocurre es lo siguiente:

- El Turno **A00** fue atendido a las 17:36 y su duración fue de 2.5 (porque el tipo es SERVIR ALMUERZO), por tanto, la hora después de atender ese turno es 17:38.5, luego espera 15 segundos (que es la espera entre turnos obligatoria -recuerde que 15 segundos son 0.25 minutos-) y la hora queda en 17:38.75.
- Ahora atiende el turno **A01** cuya duración es de 3.5 y después de atenderlo la hora es 17:42.25. Se esperan 15 segundos y ahora la hora es 17:42.5.
- Se atiende el turno **A02** con duración 2.5. Después de atenderlo la hora es 17:45, y después de la espera de 15 segundos, la hora es 17:45.25.
- Se atiende el turno **A03** con duración 2.5. La hora ahora es 17:47.75 más la espera queda queda en 17:48.

Termina. No se atienden más turnos por ahora porque en este momento son justamente las **17:48**. Los turnos A04 y A05 no se atienden aún porque no ha llegado la hora de atenderlos.

En amarillo los requerimientos funcionales

En azul y verde no funcionales