	Estructura de Datos Proyecto Final	Prof: MSc. José Alvarado Moreira
		Código curso: SC-304, 2023-3C

1. Especificación

Proyecto programado grupal a desarrollar para su entrega y exposición en Semana 14. Con entregas de avance parciales en semana 7 y semana 11.

Los grupos deben estar conformados por 4 estudiantes. Los problemas entre compañeros del grupo deben solucionarse a lo interno de cada grupo.

2. Código

El código debe entregarlo durante la primera hora de clase en la semana 14, después de esto no se recibirán proyectos, y perderá la calificación de esta evaluación. Bajo ninguna condición existen prórrogas.

Adicionalmente el código deberá ser administrado por un repositorio en GitHub, con acceso al profesor cómo admin: jalvarado50778@ufide.ac.cr. No se permiten modificaciones del Repo una vez entregado el proyecto. El nombre del repositorio debe tener el siguiente formato: 2023_3C_SC-304_HE_DIA_GRUPO dónde DIA será K para martes y J para jueves y GRUPO será el grupo asignado.

Un sólo integrante por grupo subirá al campus virtual un único archivo .zip con su proyecto con todos los módulos necesarios. Dentro de ese archivo Zip, agregue un archivo de texto con el nombre **README.md** (formato Markdown, documentación en: <https://www.markdownguide.org/> , Ejemplos en: <https://dillinger.io/>) donde indique:

1. Número de grupo asignado
2. Integrantes finales del grupo. A los que se les asignará la nota del proyecto
3. Documentación necesaria para hacer la corrida con ejemplo
4. Documentación del menú de usuario y sus opciones (en lista con prosa de explicación)
5. Prosa explicando los algoritmos utilizados, estructura de datos, formato de variables, estándares de programación y cualquier detalle técnico a discreción. Se recomienda extractos de código usando Markdown
6. Capturas de pantalla en formato jpeg de la corrida


Nota Final de Proyecto

El proyecto será calificado según la rúbrica que se presenta en el programa del curso.

Se evaluarán los temas según hayan sido vistos en clase. Si cumple con los requisitos especificados se asignan todos los puntos del tema, sino se descuentan según se incumpla.

Adicionalmente se evaluará el uso de documentación interna en todo el código. Modulación de código es recomendada para tener mejor orden en el mismo.

No está permitido ningún *framework*, código previo o librerías sin previa aprobación del profesor. Todo código debe ser generado por los estudiantes desde cero. El código de cada grupo debe ser desarrollado por cada grupo por separado.

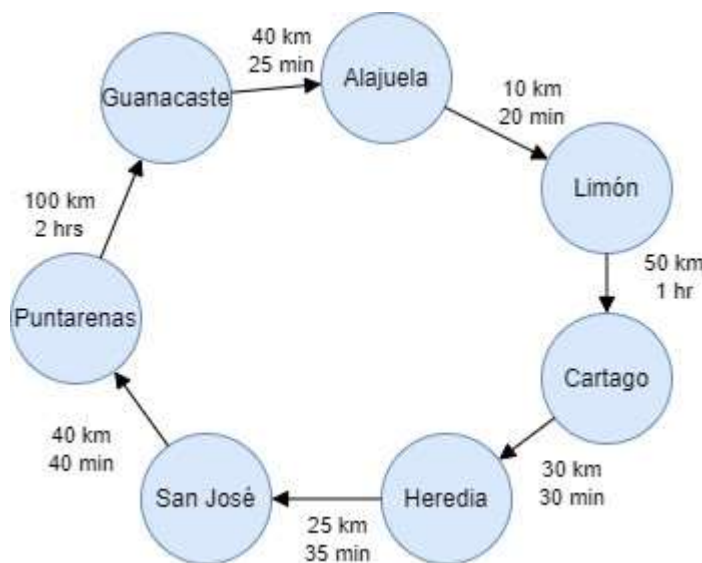
	Estructura de Datos Proyecto Final	Prof: MSc. José Alvarado Moreira
		Código curso: SC-304, 2023-3C

3. Enunciado

Una empresa administradora de trenes requiere de su ayuda para implementar una solución para organizar el recorrido y la distribución de los asientos de su tren periférico. Esta empresa requiere de un sistema que organice las rutas que debe tomar el tren en su recorrido por las diferentes provincias, considerando las distancias entre las diferentes estaciones que debe recorrer.

Además de esto, necesita de un sistema que le permita organizar los pasajeros que hacen uso de este tren, de acuerdo a la cantidad de asientos disponibles, considerando además los asientos que están reservados de forma preferencial para personas con algún tipo de discapacidad.

Respecto a la ruta del tren periférico, deberá recorrer una a una cada una de las estaciones ubicadas en cada provincia, considerando tanto distancia como tiempo. Esto debe hacerse a través de un grafo (utilizando matriz de adyacencia) representado de la siguiente manera:




El sistema deberá simular el movimiento del tren según imagen anterior. Por cada estación se llevará una cola de pasajeros la cual deberá esperar a ingresar al vagón hasta que llegue en la estación. Una vez que el tren llegue a la estación, el sistema deberá atender cada pasajero en la cola y validar espacios disponibles según requerimientos del pasajero. Es requerido el uso de hilos para sumar el movimiento del tren en cada estación

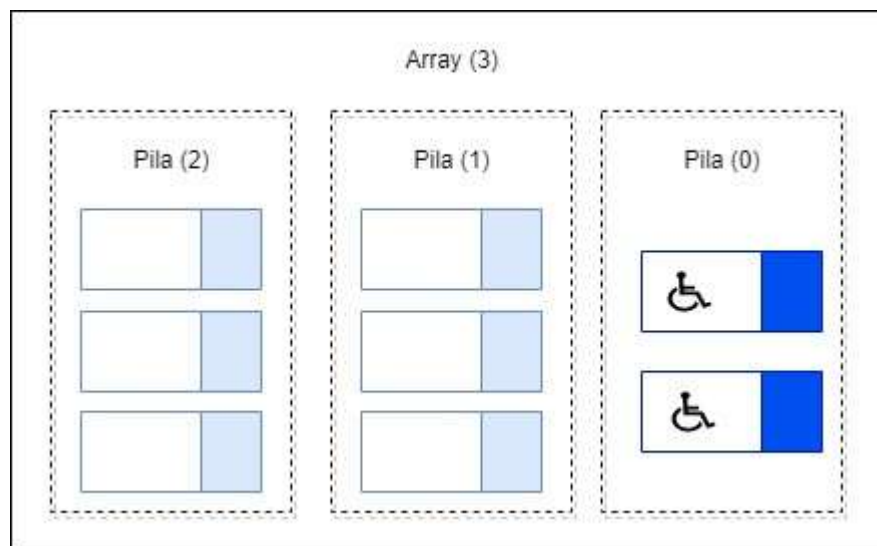
Una vez finalizado cada viaje, se debe registrar el costo del boleto, que será calculado mediante la suma de cada kilómetro recorrido a un costo especificado en configuración, se debe utilizar el algoritmo de ruta corta *Dijkstra* para el cálculo

Para la distribución de los pasajeros en los asientos del tren, debe considerarse lo siguiente:

- Hay 3 filas de asientos en el vagón del tren. El vagón del tren debe corresponder a una lista que tiene dentro 3 pilas, una para cada fila.
- La fila 1 son preferenciales para personas con discapacidad y tiene 2 asientos. Esto corresponde a 1 pila de 2 nodos.

	Estructura de Datos Proyecto Final	Prof: MSc. José Alvarado Moreira
		Código curso: SC-304, 2023-3C


- Las filas 2 y 3 tienen 3 asientos cada una, lo que corresponde a 2 pilas de 3 nodos cada una (En el caso de estar llenas)
- Debe existir un atributo en clase pila que determine el máximo de nodos permitidos, más no utilizar nodos con dato en null
- En cada estación, hay una cantidad de personas que suben al tren y a la vez, hay una cantidad de personas que bajan del tren. Se deberá ejecutar el desalojo primero y después el ingreso
- Se llevarán dos colas de espera para los pasajeros, por cada estación y llenar el tren según espacios disponibles, si el tren está en su máxima capacidad, dejar pasajeros en espera hasta siguiente llegada del tren. Una cola será de discapacitados y otra de usuarios NO discapacitados
- Se deberá respetar la fila de asientos en comportamiento de *Pila*, en el caso de que un pasajero al fondo de la pila requiera su salida, se debe desapilar (pop) todos los pasajeros en una pila auxiliar y volver a apilar en el mismo orden, dejando siempre espacios disponibles al tope de la pila.
- Cada persona debe tener un atributo que indique si tiene alguna discapacidad o no, de forma que tiene prioridad en el asiento de la fila 1.
- Si una persona que no tiene ninguna discapacidad no se debe permitir ubicar en fila 1
- Cada Pila debe iniciarse vacía (*null*), con restricción de capacidad según puntos anteriores



Módulo 1: Módulo de configuración

Módulo será necesario para almacenar la configuración del sistema. NO es requerido administración vía pantallas, si no en archivo.

1. Se puede utilizar librerías externas para manejar de archivos de configuración, tipo ini, archivo plano, con formato KEY=VALUE. Se recomienda el uso de ini4j.
2. El archivo ini será sólo lectura y se leerá una vez iniciado el sistema
3. Se debe almacenar:
 - a Número de filas por vagón CON discapacidad, por defecto 1
 - b Número de filas del vagón SIN discapacidad, por defecto 2
 - c Número de asientos de fila CON discapacidad, por defecto 1
 - d Número de asiento de fila SIN discapacidad, por defecto 3
 - e Nombre de la empresa
 - f Tiempo en segundos en que el tren avanza entre paradas, por defecto 1 segundo
 - g Costo por kilómetro recorrido: 100 colones
4. El costarchivo ini estará almacenado en la raíz para su modificación

	Estructura de Datos Proyecto Final	Prof: MSc. José Alvarado Moreira
		Código curso: SC-304, 2023-3C

Módulo 2: Administración de pasajeros


Módulo de ingreso de pasajeros	<ol style="list-style-type: none"> 1. El usuario debe tener la capacidad de agregar pasajeros según las necesidades y características de este. 2. El usuario debe tener la capacidad de listar todos los pasajeros tanto en estaciones como en vagón del tren 3. Datos del pasajero: <ol style="list-style-type: none"> a Nombre y apellidos b Edad c Estación origen d Estación destino e Discapacitado SI/NO, usando <i>enum</i> en java f Estado: EN_COLA/EN_CAMINO/COMPLETADO
--------------------------------	---

Módulo 3: Visualización de estaciones y vagón

Este módulo funciona para representar el movimiento de pasajeros, en pantalla	<ol style="list-style-type: none"> 1. Deberá permitir la visualización de estaciones y el vagón del tren en UNA sola pantalla 2. La pantalla deberá refrescar según movimiento del tren, por defecto cada segundo 3. Por cada estación, se debe mostrar: <ol style="list-style-type: none"> a Colas de pasajeros por ingresar (Discapacitados y NO discapacitados), con sus características. Especificar ciclos de espera en caso de que no haya podido encontrar espacio para ingresar. b Pasajeros con viaje completado, con sus características incluyendo su costo. Además, debe mostrar un resumen en texto (bitácora) en el caso de que hubiera que des-apilar y volver a apilar para ejecutar la salida del pasajero 4. El vagón del tren debe mostrar: <ol style="list-style-type: none"> a Distribución de asientos, mostrando datos de pasajeros b Estado actual de su ubicación (estación) 5. Presentar en consola los mensajes necesarios por cada ejecución de los algoritmos. Por ejemplo, escribir estación actual del tren, pasajeros que dejaron el vagón y pasajeros que entraron en el vagón.
---	--

4. Entregables

Primera Pre-Entrega (Semana 7)	<ul style="list-style-type: none"> • Creación de repo en GitHub (uno del grupo) con respectivos accesos a los demás integrantes + profesor • Plan de proyecto con responsables por tarea (en Git) • Diagrama de clases (en Git) • Diseño de menú e interfaces de usuario • Componentes a visuales y librerías a utilizar (en Git)
Segunda Pre-Entrega (Semana 11)	<ul style="list-style-type: none"> • Avance del plan del proyecto (en Git) • Imitación de visualizaciones con datos de ejemplo • Entrega de prototipo funcional, básico funcionamiento con opciones de menú e interfaz de usuario (en Git)
Entrega Final (Semana 14)	<ul style="list-style-type: none"> • Entrega del proyecto completa con su defensa (en Git)

	Estructura de Datos Proyecto Final	Prof: MSc. José Alvarado Moreira
		Código curso: SC-304, 2023-3C

5. Presentación y Demostración

5.1 Presentación

La presentación y demostración debe realizarse en semana 14.

Deberá durar 10-15 minutos en total. La exposición se hará escogiendo el orden de los grupos que pasarán al frente según su número asignado en un sorteo aleatorio. Uno tras otro hasta terminar.

El grupo seleccionado deberá presentar en forma oral, sin hacer uso de presentación:

1. Introducción:
 - a. Presentar a todos los integrantes del grupo
 - b. Presentar la temática del proyecto
 - c. Presentar arquitectura de clases
 - d. Visualización de pantallas
 - e. Defensa del sistema en vivo
2. Estructuras de la Solución:
 - a. Presentar las estructuras de datos utilizadas, cómo implementan sus algoritmos y cómo cumplen con las condiciones de cada módulo, mostrando ejemplos específicos de código para demostrar el cumplimiento de los temas y sus condiciones.
3. Retos y lecciones
 - a. Comparta con la clase los retos y lecciones de realizar la visión de las funcionalidades de su proyecto en cuanto a decisiones de diseño, discusiones de implementación, etc.

6. Adicionales

- No se solicita trabajo escrito. El código debe estar actualizado en Git
- Si tiene un problema con un compañero de grupo, comuníquemelo lo antes posible.
- La asistencia al día de la exposición es obligatoria.
- No se puede reponer, solo se puede justificar la ausencia.
- El estudiante deberá mostrar dominio completo de la aplicación desarrollada.
- No puede retirarse de la clase hasta que expongan todos los grupos.
- Si se comprueba la intervención de personas ajenas al grupo en la programación del proyecto, éste será anulado y será causa para la pérdida del curso. Se aplicará la misma sanción en caso de que se compruebe que existe copia parcial o total de códigos de Internet.

"Cree en ti mismo: Protagoniza tu vida. No te victimices. Cree en ti mismo con tanta fuerza, que el mundo no pueda evitar creer en ti también. Ten voluntad. Pon de tu parte. Merece la felicidad. El premio más grande es la sensación de capacidad; para eso, necesitas poner tu mejor esfuerzo. Preocúpate por darte a los demás. A pesar de tu tragedia, otros necesitan de ti. Si aprendes a recibir, pronto sentirás la necesidad de dar. Sé objetivo. La vida no es dura, sólo hay momentos difíciles. Absolutamente todo es pasajero." (Hábitos de las Personas resilientes)