

<p>Universidad Nacional de Costa Rica ESCUELA DE INFORMÁTICA EIF 204 PROGRAMACIÓN II</p>
<p>PROYECTO 1 EJERCICIO DE CÁTEDRA N° 1</p> <p>FECHA DE ENTREGA: SEMANA 9 (Domingo 15 de Mayo) PUNTAJE TOTAL: 100 Puntos PORCENTAJE: Corresponde a un 15% de la nota final del curso</p>

CONDICIONES DEL PROYECTO

El Proyecto de Aprendizaje del curso EIF 204 Programación II pretende que el estudiante ponga en práctica los siguientes temas:

- Principios de Diseño
- Jerarquías de Clases
- Relaciones entre Clases

Se presenta un problema que el estudiante debe implementar en el IDE seleccionado por el profesor para desarrollar un proyecto en C + +, y posteriormente subirlo al aula virtual en la sección que el profesor habilite para esto. El ejercicio se trabaja en grupos de máximo dos personas. Cualquier plagio se calificará con 0, como lo establece el artículo 24 del Reglamento General sobre los procesos de Enseñanza y Aprendizaje de la Universidad Nacional.

Recuerde que además, como es de esperar los proyectos tienen una naturaleza de investigación por cuanto, es importante que usted como parte del proyecto aplique los conocimientos adquiridos e investigue en el proceso de desarrollo y solución del proyecto.

TEMAS EVALUADOS:

1. Mejores prácticas de programación
2. Programación Orientada a Objetos (POO)
3. Relaciones entre clases
4. Manejo de excepciones
5. Programación Genérica

ENUNCIADO DEL PROBLEMA

La empresa de buses **BusCar Costa Rica S.A.**, se encarga de brindar el servicio de transporte a destinos turísticos de Costa Rica.

Durante la pandemia por el COVID-19 y a raíz de las restricciones sanitarias impulsadas por el Ministerio de Salud para minimizar el riesgo de contagio del virus, instruye que todas las empresas de transporte tienen como requisito temporal que los buses puedan brindar el servicio de transporte a un 80% de su capacidad total, este requerimiento crea un problema de asignación de espacios al momento de emitir un boleto. Los siguientes datos indican las unidades disponibles de la empresa y la capacidad de transporte al 100%.

Tipos de buses:

- Buseta: Hasta 40 pasajeros
 - Placa: BUS001
 - Placa: BUS002
 - Placa: BUS003
- Coaster: Hasta 21-27 pasajeros
 - Placa: COA001
 - Placa: COA002
- Hiace: Hasta 12 pasajeros
 - Placa: HIA001
 - Placa: HIA002
 - Placa: HIA003
 - Placa: HIA004
- H1: Hasta 6 pasajeros
 - Placa: HON001
 - Placa: HON002

Rutas de viajes:

- Volcán Arenal, San Carlos
- Río celeste, San Carlos
- Tamarindo, Guanacaste
- Puerto Viejo, Limón
- Cóbano, Puntarenas
- Volcán Irazú, Cartago

Se solicita desarrollar un sistema que permita gestionar los buses, las rutas, y la reserva de tickets, considerando las restricciones adicionales impuestas por el ministerio de salud.

Para solucionar este problema, la empresa ha contactado a los estudiantes del curso de Programación II, para hacer un sistema en C++ con el fin de:

1. Asignar uno o varios tipos de bus a una ruta de viaje
2. Permitir reservar boletos por viaje, considerando las restricciones de capacidad máxima por unidad de transporte.
3. Las rutas de viaje podrían tener uno o más buses asignados. Si un bus llega a su máxima capacidad de pasajeros, el sistema automáticamente lo asigna al siguiente bus.

Requerimientos Específicos

1. Construya una interfaz llamada Object, esta es la superclase por excelencia de todo el proyecto así que todas las clases heredan de ella.
 - a. Debe tener como métodos virtuales puros
 - i. Método toString que retorna un string con la representación del objeto
2. Configuración de la flota de transporte
 - a. Tipos de buses
 - i. Permitir la carga de distintos tipos de buses
 - b. Rutas de viaje
 - i. Permitir la carga de las rutas de viaje
 - ii. Tener la opción de asignar los buses a las rutas de viaje, por ejemplo: la ruta al Volcán Arenal podría tener dos tipos de buses asignados: una Buseta y un H1.
3. Reserva de tickets
 - a. Reserva de tickets
 - i. Permitir reservar y emitir un ticket de viaje, el mismo debe llevar el número de asiento y el bus correspondiente.
4. Reportes
 - a. Mostrar todos los tickets emitidos indicando la ruta y el bus
 - b. Mostrar la disponibilidad de pasajeros por ruta de viaje
5. Menú o Pruebas de Unidad
 - a. Esta sección de la aplicación maneja toda la interacción con el usuario
 - b. Se evalúan los procesos y la interacción de los resultados mediante las pruebas de unidad

Distribución de pasajeros

El sistema debe tener la capacidad de distribuir eficientemente los pasajeros entre los buses que tiene disponible la empresa.

Técnicamente se debe buscar un algoritmo de distribución (ejemplo: bin packing problem).

Anotaciones

Usted debe aplicar los siguientes conceptos para cumplir con los requerimientos y obtener la puntuación asignada

1. Utilización de la delegación
2. Los contenedores deben utilizar programación genérica
3. Crear una clase para manejar la interfaz de usuario
4. Las excepciones deben estar incluidas en el manejo de archivos y el ingreso de datos vía teclado.
5. El sistema debe incluir persistencia de datos mediante la utilización de archivos.
6. Se recomienda hacer la segregación de funciones
 - a. Uso de clase controladora
 - b. Separación de interfaz de usuario (vista)
 - c. Separación del modelo o lógica del negocio

Diseño del Menú

<div><div>BusCar Costa Rica S.A.</div><div><div>Menú Principal</div><div><div>1. Configuración de la flota de transporte</div><div>2. Reserva de tiquetes</div><div>3. Reportes</div><div>4. Salir</div></div></div></div>	<div><div>BusCar Costa Rica S.A.</div><div><div>Menú Configuración</div><div><div>1. Tipos de buses</div><div>2. Restricciones sanitarias</div><div>3. Rutas de viaje</div><div>4. Regresar</div></div></div></div>
<div><div>BusCar Costa Rica S.A.</div><div><div>Tipos de buses</div><div><div>Tipo de Bus: Buseta</div><div>Cantidad max de pasajeros: 40</div><div>Placa: BUS001</div></div></div></div>	<div><div>BusCar Costa Rica S.A.</div><div><div>Rutas de Viaje</div><div><div>Nombre Ruta: Volcán Arenal, San Carlos</div><div>Placa de Bus: BUS001</div></div><div>Nota: las rutas de viaje podrian tener 1 o más buses asignados</div></div></div>
<div><div>BusCar Costa Rica S.A.</div><div><div>Reserva de tiquetes</div><div><div>Cédula Pasajero: 1-1060-0482</div><div>Cantidad pasajeros: 2</div></div></div></div>	<div><div>BusCar Costa Rica S.A.</div><div><div>Mostrar todos los tiquetes emitidos</div><div><div>Número tiquete: 12345, ruta: Volcán Arenal, San Carlos, bus: BUS001</div><div>Número tiquete: 23423, ruta: Río celeste, San Carlos, bus: HIA002</div><div>Número tiquete: 234, ruta: Tamarindo, Guanacaste, bus: HON001</div></div></div></div>

BusCar Costa Rica S.A.

Restricciones Sanitarias

% Máximo de aforo: 80

BusCar Costa Rica S.A.

Menú Reportes

- 1. Mostrar todos los tiquetes emitidos
- 2. Mostrar la disponibilidad de pasajeros por ruta de viaje
- 3. Regresar

BusCar Costa Rica S.A.

Mostrar la disponibilidad de pasajeros por ruta de viaje

Ruta: Volcán Arenal, San Carlos. Disponibilidad actual 20%
Ruta: Río celeste, San Carlos. Disponibilidad actual 80%
Ruta: Tamarindo, Guanacaste. Disponibilidad actual 80%

BusCar Costa Rica S.A.

Mostrar todos los tiquetes emitidos

Número tiquete: 12345, ruta: Volcán Arenal, San Carlos, bus: BUS001
Número tiquete: 23423, ruta: Río celeste, San Carlos, bus: HIA002
Número tiquete: 234, ruta: Tamarindo, Guanacaste, bus: HON001

Evaluación

Detalle	Puntaje
[ESTÁNDARES] Código fuente ordenado siguiendo los estándares definidos para los nombres de las variables, documentación del código y estructura de las clases.	5
[UML] Diagrama de clases completo.	10
[POO] Uso correcto de las técnicas de Programación Orientada a Objetos [Herencia, Encapsulamiento y Polimorfismo] (5 puntos cada tema)	15
[RELACIONES DE CLASES] Implementación de relaciones de clases.	10
[ABSTRACCIÓN E INTERFACES] Utilizar de forma correcta la abstracción y las interfaces. (5 puntos cada tema)	10
[PRINCIPIOS DE DISEÑO] Diseñar e implementar utilizando al menos tres principios de SOLID.	20
[PRUEBAS DE UNIDAD O MENÚ VISUAL] Definición de las pruebas de unidad para los métodos más importantes en su sistema, o el menú de las pantallas sugeridas para evaluar la funcionalidad del sistema.	15
[MANEJO DE EXCEPCIONES] Mediante una clase especializada se debe administrar las posibles excepciones generadas ya sea por el sistema o el usuario final para que la aplicación sea robusta. Interfaz de usuario y archivos.	15