



## Cátedra Tecnología de Sistemas

Programación Intermedia

Código: 00824

### Proyecto 2. Valor 2%

#### Temas de Estudio

1. Tema 1
  - a. Programación orientada a objetos: herencia
  - b. Programación orientada a objetos: polimorfismo e interfaces
2. Tema 2
  - a. Interfaces gráficas de usuario
  - b. Archivos, flujos de entrada/ salida

#### Objetivo

Desarrollar una aplicación en java que permita demostrar el conocimiento adquirido sobre Herencia y polimorfismo, manipulación de estructuras, manejo de archivos e interfaz gráfica de usuario, así cómo, aplicar estos conocimientos dentro de una aplicación.

#### Software de Desarrollo

NetBeans

#### Desarrollo

Debido al éxito que la empresa ha tenido con la renta de bicicletas desean evolucionar su negocio, por lo que lo que se le contrata para implementar nuevos requerimientos basados en el proyecto #1.

Además de la renta de bicicletas se desean alquilar Scooters y Patines por lo que se requiere:

Crear una clase principal llamada **"Transporte"** que contenga los siguientes campos y métodos:

- ✓ **Código:** campo de texto con un mínimo de 5 caracteres.
- ✓ **Tipo:** campo de texto que indique si el transporte es eléctrico o normal
- ✓ **TarifaBase:** campo entero.
- ✓ **CalcularMonto:** método abstracto.

Crear una clase **"Bicicleta"** que hereda de **"Transporte"** que contenga los siguientes campos

- ✓ **TarifaBase:** atributo heredado que debe inicializar en 3000 colones
- ✓ **CantidadCambios:** campo entero

Crear una clase “**Scooter**” que hereda de “**Transporte**” que contenga los siguientes campos

- ✓ **TarifaBase**: atributo heredado que debe inicializar en 2000 colones
- ✓ **Color**: campo de texto, cantidad de caracteres mayor a 5 caracteres.

Crear una clase “**Patines**” que hereda de “**Transporte**” que contenga los siguientes campos

- ✓ **TarifaBase**: atributo heredado que debe inicializar en 1500 colones
- ✓ **CantidadRuedas**: campo entero.

Cada una de las clases debe de implementar el método CalcularMonto utilizando la formula indicada en el proyecto#1 y el siguiente cuadro

Distancia recorrida	Porcentaje Bicicleta	Porcentaje Scooter	Porcentaje Patines
0 a 1000	50 %	40 %	20 %
1001 a 2000	35 %	30 %	15 %
2001 a 3000	20 %	25 %	10 %
+ de 3000	10 %	15 %	5 %

La clase “**Factura**” del proyecto#1 debe modificarse de la siguiente forma:

- Eliminar el atributo tarifaBase
- Eliminar el método que realizaba el cálculo final.
- Renombrar el atributo CodigoBicicleta por CodigoTransporte.
- Agregar un atributo fecha donde se guarde la fecha en la que se genera la factura del alquiler.

Debe crear un archivo “**Transporte.txt**” donde guarde la información del transporte que la empresa tiene disponible para el alquiler. El sistema debe permitir ingresar, modificar y eliminar transportes y cada acción debe de reflejarse en el archivo “**Transporte.txt**”.

El sistema no debe permitir crear una factura si no hay transportes creados en el archivo “**Transporte.txt**”.

El sistema no debe permitir crear facturas si el código del transporte que se está ingresando no existe en el archivo “**Transporte.txt**”.

Debe crear un archivo llamado “**Alquiler.txt**”, donde guarde la información de las facturas generadas sobre los transportes alquilados.

El sistema debe permitir, ingresar y buscar facturas.

Debe generar 2 reportes

1. Uno que le permita al usuario filtrar el archivo por cada tipo de transporte

2. Uno que le permita al usuario filtrar por fecha los alquileres de los transportes.

La información que debe mostrar en cada reporte es: Nombre y ID del cliente, tipo y código del transporte y monto cancelado por cada alquiler realizado.

Debe tener en cuenta el uso y manejo de excepciones con try..catch y validaciones indicadas para los atributos de las clases.

El proyecto debe de presentarse con interfaz gráfica, donde debe de utilizar al menos 3 componentes GUI.

## Honestidad Académica



<https://audiovisuales.uned.ac.cr/play/player/23048>

Nota Importante

Cada estudiante es responsable del contenido que entrega, si no es el archivo correcto, no podrá entregarlo posterior a la fecha establecida.

Si el contenido del archivo coincide con algún otro estudiante, o se comprueba que no es de su autoría, se aplicaría lo indicado en la plataforma en el documento [Lineamientos ante casos de plagio](#)

## Indicaciones Importantes

- Es obligatorio que incluya todo el directorio donde se encuentra Proyecto 2.
- El **Proyecto 2** debe estar desarrollado en **NetBeans** que es la herramienta oficial del curso.
- El programa debe ser modular, utilizando de la mejor manera funciones definidas por usted.
- Los trabajos deben realizarse en forma individual. Dentro del código del programa debe de indicar la documentación que explique cómo fue realizado el programa.
- Si utiliza código de algún ejemplo del libro, o de otra fuente que no sea de su autoría, debe de indicarlo.

- Comprima todos los archivos en un solo archivo .zip o .rar.
- **Nombre del archivo que envía:** debe ser nombre y primer apellido del estudiante, y nombre de la tarea. **Ejemplo: JuanRojas-proyecto2.**
- La entrega del **Proyecto 2** en las fechas establecidas en la plataforma de aprendizaje en línea Moodle en el apartado que se indique.
- Si no concluyó a tiempo la tarea, debe entregar lo que pudo hacer e incluir una carta explicando las razones por las cuales no finalizó.

### Rúbrica de Evaluación

Criterio	Cumple a satisfacción lo indicado en la evaluación	Cumple medianamente en lo indicado en la evaluación	Cumple en contenido y formato, pero los aportes no son significantes	No cumple o no presenta lo solicitado
Presentación del código: Correcto uso de variables, nombres de métodos significativos, no presenta métodos vacíos.	4	2	1	0
Comentarios en código: No comete errores gramaticales u ortográficos dentro de los comentarios del código y son significativos para cada método y atributo	4	2	1	0
Correcta creación de clase "Transporte"	4	2	1	0
Correcta creación de clase "Bicicleta"	10	5	3	0
Correcta creación de clase "Scooter"	10	5	3	0
Correcta creación de clase "Patines"	10	5	3	0
Correcta modificación de clase "Factura"	4	2	1	0
Desarrollo de los diferentes métodos y cálculos solicitados	4	2	1	0
Manejo de archivo Transporte.txt	10	5	3	0
Manejo de archivo Alquiler.txt	10	5	3	0
Utiliza al menos 3 componentes GUI	5	3	2	0

Manejo de excepciones con try..catch	5	3	2	0
Reporte por transporte	10	5	3	0
Reporte por fecha	10	5	3	0
TOTAL	100			