

Apellido y nombre: _____ Legajo: _____

PRIMER EXAMEN PARCIAL

00/00/0000

Resolvé el siguiente ejercicio utilizando el lenguaje **Java** y el paradigma orientado a objetos. Lee el enunciado al menos dos veces antes de intentar confeccionar la solución.

ENUNCIADO

TodoLibre es un nuevo sitio dedicado a la compra/venta de productos y sus respectivos envíos. El prototipo de la aplicación cuenta con una lista de usuarios registrados en la app. También tiene una lista con las órdenes de compra que deben enviarse.

De los usuarios se sabe su ID, nombre, apellido, mail y domicilio (calle, altura y número de comuna).

De todas las órdenes se sabe el número, el monto, el usuario vendedor y el usuario comprador.

Todas estas órdenes deben ser entregables, lo que significa que deben ser capaces de retornar, de forma independiente, el domicilio de origen y el domicilio de destino.

En particular, de las órdenes Puerta a Puerta se sabe si su envío es o no gratis. El domicilio de destino será el del comprador. De las órdenes de Retiro en Persona se saben los datos de la sucursal de correo donde se debe dejar el paquete. El domicilio de destino será el de la sucursal de correo.

El domicilio de origen es siempre el del vendedor.

Se desea que, al correr la aplicación, se muestren de forma clara y ordenada los siguientes resultados:

- A) Porcentaje de órdenes con destino/origen en la misma comuna.
- B) Cantidad de órdenes Puerta a Puerta con envío gratis.

FORMATO DE ENTREGA

Archivo **.zip** o **.rar** nombrado de la siguiente manera (reemplazando lo que está en verde):

INSPT-ProgII-p1Modelo-TodoLibre-TUAPPELLIDO-TUNOMBRE

El **.zip** o **.rar** debe contener el proyecto Java, el cual debe tener la misma nomenclatura.

No cumplir con alguna indicación del formato de entrega disminuye la nota del examen.

CRITERIOS DE EVALUACIÓN

Para considerar aprobado el examen, el mismo debe demostrar la correcta aplicación de los siguientes conceptos de la programación orientada a objetos y el lenguaje Java:

- Definición de clases y asignación adecuada de sus responsabilidades.
- Encapsulamiento, ocultando detalles de implementación y utilizando métodos getters y setters sólo cuando corresponda.
- Modularización reutilizable y mantenible, usando funciones con correcta parametrización, aplicando alta cohesión y bajo acoplamiento.

- Correcta implementación de los constructores.
- Validación de los datos que ingresan al sistema.
- Aplicación de herencia y polimorfismo, incluyendo interfaces.
- Aplicación de los principios S.O.L.I.D
- Algoritmos precisos, finitos y eficientes.
- Buenas prácticas de programación:
 - Uso de constantes en lugar de números mágicos.
 - Nombres de variables, métodos y clases descriptivos, que sigan la convención del lenguaje.
 - Uso adecuado de los ciclos. No alterar la naturaleza de los mismos utilizando, por ejemplo, instrucciones de control como **break** y **continue**.
 - Código ordenado e indentado correctamente.

% Correcto	0 a 20	25 a 45	50 a 55	60	65 a 70	75	80	85 a 90	95	100
Nota	1	2	¿4?	4	5	6	7	8	9	10

(Escala solo con fines orientativos)