

<u>UTN – FR Mar del Plata – TSP</u> <u>Laboratorio III - Primer parcial</u> <u>Comisión 1 Mayo 2021</u>	<i>Nombre y Apellido</i>	<i>Nota</i>
---	--------------------------	-------------

IMPORTANTE:

- Crear un proyecto con su nombre y apellido.
- Realizar todas las funciones que se indican. Añadir comentarios
- Desarrollar un método main() que invoque únicamente a las funciones que compilan correctamente. No se pide ingreso por teclado de ningún tipo.

Tabla de puntuación:

Obtenido	10	20	30	40	50	60	70	80	90	100
Nota	1	2	3	4	5	6	7	8	9	10

Una empresa de distribución de golosinas nos pide un sistema para la gestión de su RRHH. Los empleados de esta empresa son vendedores o administradores. De ambos empleados nos gustaría almacenar su legajo, su nombre, apellido, dirección y fecha de nacimiento. Los vendedores tendrán un monto de venta mensual como particular mientras que los administradores un área (finanzas, ventas, marketing, logística, etc)

- Crear constructores vacíos, completos.
- Crear getters/setters según crea conveniente.
- Sobreescibir el método toString() en todas las clases.
- Sobreescibir el método equals()

Puntaje: 20 pts

Todos los empleados deben almacenarse dentro de un ArrayList con las funciones de agregar, buscarPorLegajo, contarEmpleados, devolverListado y eliminarPorLegajo.

- Crear la clase envoltorio/contenedora.
- Crear y codificar los métodos correspondientes.
- Invocar a dichos métodos para probar funcionamiento.

Puntaje: 30 pts.

Todos los empleados cobran un sueldo. Los empleados administrativos cobran un fijo de 15 (no importa qué o cómo). Los vendedores cobran un 30% de lo vendido. El cliente nos reveló que en un futuro los repartidores (que no son empleados de él) también van a cobrar un sueldo. Se le paga 3 por cada kilómetro recorrido.

- Crear ????? necesarios para soportar los repartidores.
- Crear la clase Repartidor.

- Crear un método que nos permita calcular el total que tengamos que pagar en términos de sueldo. Se puede hacer de MUCHAS formas pero hay una o varias que son las correctas en términos conceptuales con polimorfismo, herencia, etc)

Puntaje: 30 pts.

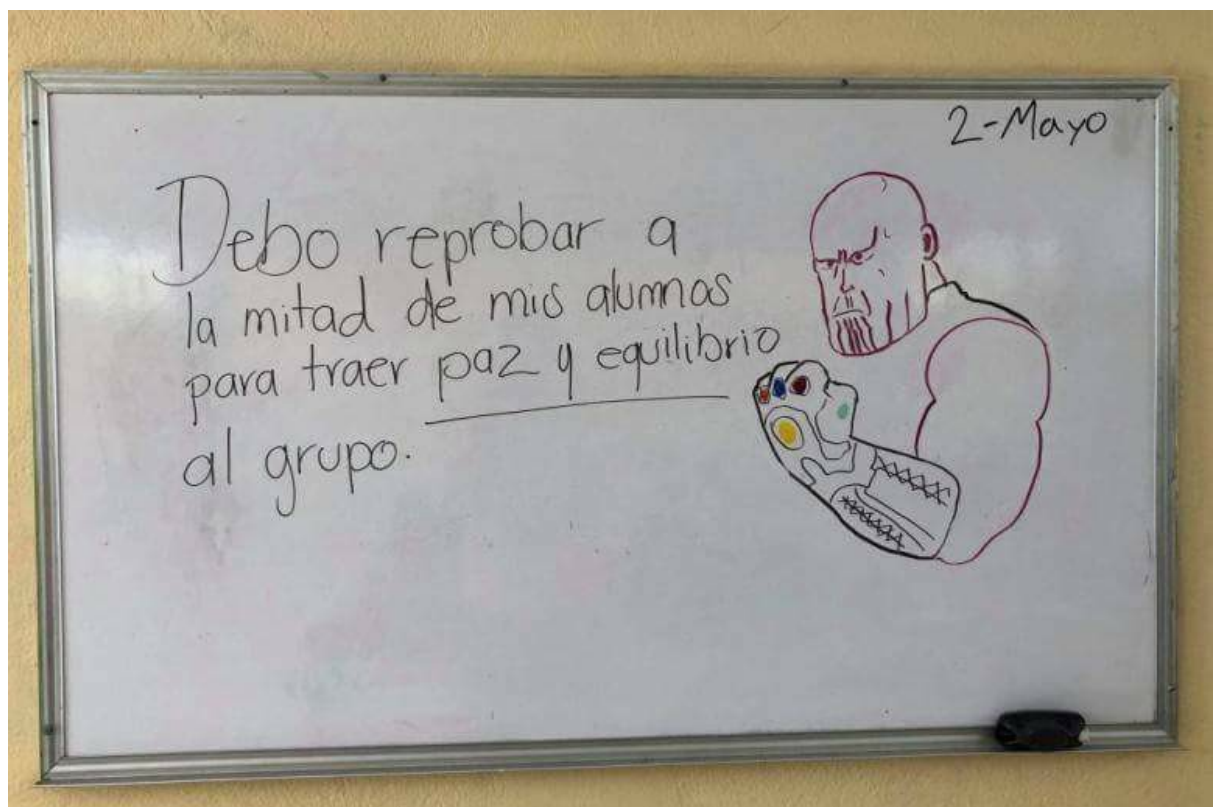
Para probar todo nuestro sistema haremos un lindo main con un switch que deberá estar modularizado (recuerden que los métodos son públicos y static y no deben ir variables de clase (globales) sino pasaje de parámetros). El menú deberá contar con todas las opciones necesarias para demostrar el correcto funcionamiento.

Puntaje: 10 pts.

Utilización de conceptos de la POO. Diseño de clases. Comentarios. Prolijidad. Modularidad, acoplamiento, cohesión. Entrega del parcial. Originalidad.

Puntaje: 10 pts.

¡Los mejores éxitos!



**WRITING CODE FOR AN HOUR
AND COMPILE**

VIA 9GAG.COM

**COMPILES WITH NO
ERRORS**

