

## Recuperatorio 1er Parcial: Types, types, types...

Un nuevo desarrollador implementó la funcionalidad necesaria para determinar, al momento de importar un Customer (método importCustomer de CustomerImporter), si el tipo de identificación es DNI o CUIT, con sus validaciones correspondientes.

Lo mismo hizo al momento de importar un Address (método importAddress de CustomerImporter), con el objetivo de poder diferenciar el viejo código postal (solo 4 dígitos, ej: 1636) del nuevo código postal (ej: B1336BBE).

Debido a estos tipos de documento y de código postal es que agregé protocolo en Customer y Address para diferenciarlos.

Por suerte el desarrollo lo hizo por medio de TDD y hay tests para todo lo nuevo. Sin embargo parece que desconocía buenas prácticas de diseño y utilizó IF en lugares que podría haber usado polimorfismo, y además dejó bastante código repetido.

Tu tarea es sacar los IF y el código repetido que consideres necesario para dejar un buen diseño. Se puede hacer cualquier cambio de diseño e implementación que consideres necesario, pero hay que mantener los tests como están.

Usar el archivo CustomerImporter-Recu-1er-Parcial como punto de partida.

## Recuperatorio 2do Parcial: CustomerImporter - El regreso

A la empresa le ha ido muy bien con el sistema de importación de clientes y ¡Hay más trabajo para hacer!

Debemos importar las ventas realizadas a los clientes. Para ello, hay que utilizar un archivo CSV como el que sigue:

S, 2022/08/10,0001-00000022,1000	← Línea que indica que se hizo una venta (Sale)
EC,D,1122	← Comprador. Es un "Existing Customer"
P,1 kilo de huevos,700	← Producto vendido (Product)
P,1 kilo de manzanas,300	
ES	← Indica el fin de los datos de la venta (End Sale)
S, 2022/08/11,0001-00000023,2500.00	← Línea que indica que se hizo una venta
NC,Juan, Perez,D,3344	← Comprador. Es un New Customer
P,1 Kilo y medio de lomo,2500.00	← Producto vendido
ES	← Indica que es el fin de los datos de la venta

Los tipos de registro son los siguientes:

- 1) S: Indica que es un registro de venta, con la fecha de la misma, el número de factura y total de la venta. Toda venta es realizada a un cliente, el cual ya puede existir (o no) en el sistema.

- 2) EC: Indica registro de cliente ya existente. Es el cliente de la venta que se está importando, y que ya existe en el sistema, por eso solo viene el tipo y número de identificación importado previamente.
- 3) NC: Indica registro de cliente nuevo. Es el cliente de la venta que se está importando, y que aún no existe en el sistema, por lo que hay que importarlo completo, como si fuese del archivo de clientes.
- 4) P: Indica registro de producto. Es el producto que se vendió con la descripción y el precio.
- 5) ES: Indica que se terminó con los datos de la venta.

La venta sólo puede ser creada si es válida. Para que una venta sea válida debe tener un único cliente relacionado (comprador) y uno o más productos.

Desarrollar por medio de TDD y las heurísticas de diseño vistas durante el cuatrimestre, el importador que permita cargar al sistema esta información.

El importador debe ser robusto respecto a errores de formato del archivo, sin embargo no se deben validar temas de consistencia de información (ej. que el total de la venta sea la suma de los productos) ni de tipos de datos (ej. que el total sea un número válido)

El importador debe funcionar con el sistema transient únicamente.

Usar el archivo CustomerImporter-Recu-2do-Parcial como punto de partida.

Consejos:

- 1) Empezar creando un importador de CSV genérico a partir del CustomerImporter. Recordar los refactorings Insert Superclass y Push up
- 2) Utilizar el CustomerSystem para guardar las ventas también. Renombrarlo de manera acorde.
- 3) La venta debe ser un objeto válido y completo desde el momento que se crea. Debido a que la información de la misma viene en varios registros, se recomienda utilizar un Builder o similar para crear la venta al momento de encontrarse con el registro ES.
- 4) La fecha de la venta puede ser un String, no es necesario convertirla a Date.
- 5) No hay que validar que los números sean correctos.
- 6) No hay que hacer validaciones sobre el producto (ej. que la descripción no esté vacía)

## Entrega:

1. Entregar por mail el fileout de la categoría de clase donde está la solución que debe incluir toda la solución (modelo y tests).
2. Entregar también por mail el archivo que se llama **CuisUniversity-nnnn.user.changes**
3. Probar que el archivo generado en 1) se cargue correctamente en una imagen “limpia” (o sea, sin la solución que crearon) y que todo funcione correctamente. Esto es fundamental para que no haya problemas de que falten clases/métodos/objetos en la entrega.

4. Realizar la entrega enviando mail a la lista de Docentes: [ingsoft1-doc@dc.uba.ar](mailto:ingsoft1-doc@dc.uba.ar) con el **Subject: LU nnn-aa - Recuperatorio 1ro/2do Parcial 1c2022** (poner 1ro o 2do parcial según corresponda)
5. Subir a sus repos grupales los archivos **CuisUniversity-nnnn.image** y **CuisUniversity-nnnn.changes**. Debe **zippearlos** previamente para reducir su tamaño o podría dejar sin espacio disponible a sus compañeros. **Pueden eliminar las imágenes del 1er parcial y 2do para liberar espacio.**
6. Deberán subirlos al main branch de sus respectivos repos (tenga en cuenta hacer pull antes de ser necesario), y al subdirectorio **/Recu/LUnnn-aa/**

### IMPORTANTE:

No retirarse sin tener el ok de los docentes de haber recibido el mail con la resolución.