



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN  
IIC2233 - PROGRAMACIÓN AVANZADA

# Actividad 02

2º semestre 2017  
17 de Agosto de 2017

## Herencia, Polimorfismo y Clases Abstractas

### Introducción

El supermercado Programarc requiere tus servicios para modelar y programar el funcionamiento de uno de sus locales.

El supermercado vende muchos productos, cada uno con un nombre, un SKU<sup>1</sup> único generado automáticamente al ser registrado, y un precio unitario. Todo producto, al ser revisado, despliega un mensaje con el SKU, el nombre y el precio normal como el siguiente:

[987654321] Papas Fritas Mami's \$ 990.

Los productos se dividen en categorías:

- Comida: Cada alimento contiene una tabla con información nutricional por cada 100 gramos (calorías, carbohidratos, proteínas y grasa) y una fecha de vencimiento<sup>2</sup>. De quererlo, un cliente puede acceder a la información nutricional correcta para cualquier cantidad de gramos (o predeterminadamente, 100 gramos). Por ejemplo, esta tabla para 100 gramos es mostrada así:

Calorías	= 120 (kcal)
Proteínas	= 10 (g)
Carbohidratos	= 5 (g)
Grasa	= 25 (g)

Opcionalmente, los alimentos se pueden clasificar en las siguientes subcategorías:

- Lácteos: tienen un contenido de calcio (mg) en su información nutricional.
- Verduras: tienen un contenido de vitamina C (mg) al final de su información nutricional.
- Carnes: tienen un tipo que indica de qué animal provienen. Estos pueden ser pescado, vacuno, pollo o cerdo.

---

<sup>1</sup>Stock Keeping Unit

<sup>2</sup>Puede ser un string.

- Vestuario: Cada pieza de ropa tiene una talla y una categoría (femenino, masculino, o neutro). Un cliente puede ver la información de la ropa de la siguiente forma:

```
Talla      = XXXS
Categoría = F
```

- Adicionalmente, existen productos que no caben estrictamente en solo una de estas categorías. Es su deber modelar de alguna forma categorías que permitan instanciar un pantalón de queso y zapatos de pescado.<sup>3</sup>. Estos últimos entran en la categoría de comida para aplicar sus respectivos descuentos.

Toda persona tiene un nombre y una fecha de nacimiento. A cada persona se le puede preguntar su edad, pero no se podrá saber su fecha de nacimiento exacta.

Este supermercado funciona de la misma manera que los de su competencia: los clientes tienen un monto de dinero (no público), entran con un carro y agregan los productos que deseen llevar (considerando la cantidad que lleven de cada producto). Cuando estén listos, pasan por la caja para que les cobren. Si un cliente no tiene dinero suficiente al ser cobrado, se cancela la compra y no paga monto alguno. Cabe mencionar que existen clientes frecuentes inscritos, los que acceden a un 10% de descuento en ropa y 15% en comida. Además, los clientes de tercera edad (con al menos 60 años) acceden a un 10% de descuento sin necesidad de ser clientes frecuentes<sup>4</sup>. Cada producto le puede entregar a un cliente en particular su precio considerando estos descuentos. Por otro lado, los clientes tienen distintas tendencias alimenticias. Existen clientes veganos, quienes no consumen ningún producto de origen animal y vegetarianos, que no consumen carne. Estos clientes no podrán agregar productos de estas categorías a sus carros.

Para reducir costos, el supermercado sólo posee cajeros como trabajadores, quienes están entrenados para saludar cordialmente a todo cliente en cualquier momento ~~o serán despedidos~~, calcular el total de la compra y cobrar. Por otro lado, el cliente también puede saludar a los trabajadores .

Tu programa debe ser capaz de simular el proceso, desde la elección de productos hasta la compra en caja.

## Notas

- Esta actividad tiene muchas clases. Tal vez quieras pensar la modelación en papel antes de programar.
- Se espera que uses correctamente herencia, polimorfismo y clases abstractas en tu modelación.
- En esta actividad, no es estrictamente necesario implementar *properties*.
- Los productos sin categoría no existen en este supermercado.
- Recuerda que puedes definir el método `__str__` para establecer qué string se imprime al ejecutar `print(objeto)`. Por ejemplo, se espera que `print(alimento)` muestre algo como lo siguiente:

```
[987654321] Plátanos $ 1990
Calorías      = 120 (kcal)
Proteínas     = 10 (g)
Carbohidratos = 5 (g)
Grasa         = 25 (g)
```

- Recuerda que puedes definir el método `__repr__` de una clase para que sus instancias tengan un string imprimible que lo represente en todos los casos (incluso dentro de listas).

<sup>3</sup>Cuidado con el problema del diamante.

<sup>4</sup>Los descuentos no se acumulan. Se usa el mejor que esté disponible.

## Requerimientos

- (3.0 pts) Modelar correctamente los productos.
  - (0.2 pts) Correcta representación de los productos.
  - (1.0 pts) Se modelan correctamente las distintas categorías de comida.
  - (0.2 pts) Se puede leer correctamente la tabla nutricional al imprimir un alimento.
  - (0.4 pts) Se modela correctamente el vestuario.
  - (0.2 pts) Se puede leer correctamente la información de una prenda al imprimirla.
  - (0.4 pts) Se implementan correctamente el pantalón de queso y el pepino de mar.
  - (0.6 pts) Dado un cliente, se obtiene desde el producto su precio con descuentos correctamente.
- (2.4 pts) Modelar correctamente las personas.
  - (1.0 pts) Se modelan correctamente las clases necesarias.
  - (0.2 pts) Se obtiene correctamente la edad de una persona.
  - (0.4 pts) El cliente agrega correctamente un producto a su carro con la cantidad que lleva de éste, considerando sus restricciones alimenticias.
  - (0.4 pts) El cajero calcula correctamente el precio total de la compra de un cliente.
  - (0.2 pts) Las personas pueden saludar correctamente dependiendo a si son clientes o cajeros.
  - (0.2 pts) El cajero cobra correctamente el monto de la compra al cliente.
- (0.6 pts) Poblar el sistema con al menos: 1 pepino marino, 1 pantalón de queso, 1 cajero, 2 clientes (al menos uno de tercera edad), 1 lácteo, 1 verdura, 1 carne, 1 alimento de otro tipo y 3 prendas de ropa.

## Entrega

- **Lugar:** En su repositorio de Github en la **carpeta Actividades/AC02/**
- **Hora:** 16:55
- Si está trabajando en pareja, basta con que un miembro suba la actividad. Si se suben actividades distintas, se corregirá una de las dos al azar.