**5) Un supermercado organiza las cajas con las siguientes estructuras:**

TDA CAJAS:

* + Lista de Cajas
  + Cantidad de CajasCola de Tareas

Lista de Cajas:

* Numero de Caja (Clave ordenamiento)
* Recaudación
* Clientes atendidos
* Clientes por atender
* Cola de Clientes:

Nro de Cliente

Monto

Realizar procedimientos para este TDA que:

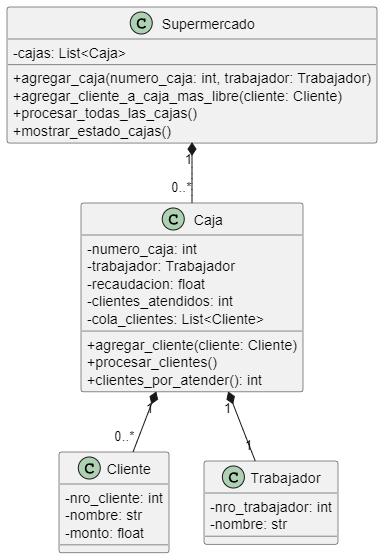
* Permita procesar todos los clientes de todas las cajas. Para ello se deberá sacar todos los clientes de cada una de las cajas, actualizando el monto recaudado en cada caso.
* Permita agregar un cliente en la caja con menor cantidad de clientes por atender.

# Desarrollo

Este modelo simula cómo funciona una fila de cajas en un supermercado, los clientes se distribuyen a las cajas más vacías, cada caja tiene un trabajador que los atiende y acumula recaudación, todo el sistema es manejado por la clase Supermercado.

* Un **Supermercado** contiene varias **Caja.**
* Una **Caja** contiene varios **Cliente** en su cola**.**
* Cada **Caja** está atendida por un solo **Trabajador**

Como podemos visualizar en la siguiente imagen:



Para el código de Python, utilizamos la Clase Cliente:  
Esta clase representa a una persona que realiza una compra y esta cuenta con atributos:

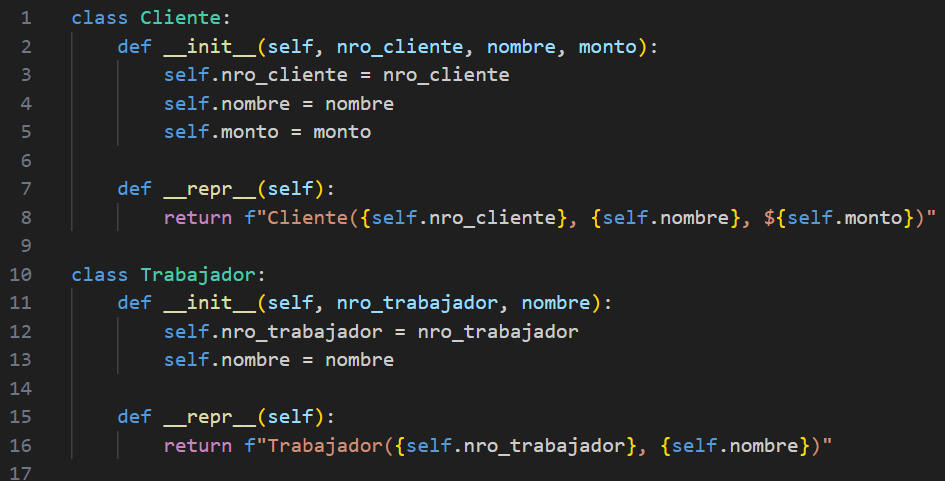
* nro\_cliente: número identificador único del cliente.
* nombre: nombre del cliente.
* monto: cantidad de dinero que debe pagar.

Este objeto se guarda en una lista (cola\_clientes) dentro de una caja.

**Clase Trabajador**

Esta clase representa al empleado asignado a una caja, cada caja tiene asignado un trabajador mediante este objeto y sus atributos son

* nro\_trabajador: ID único del trabajador.
* nombre: nombre del trabajador.



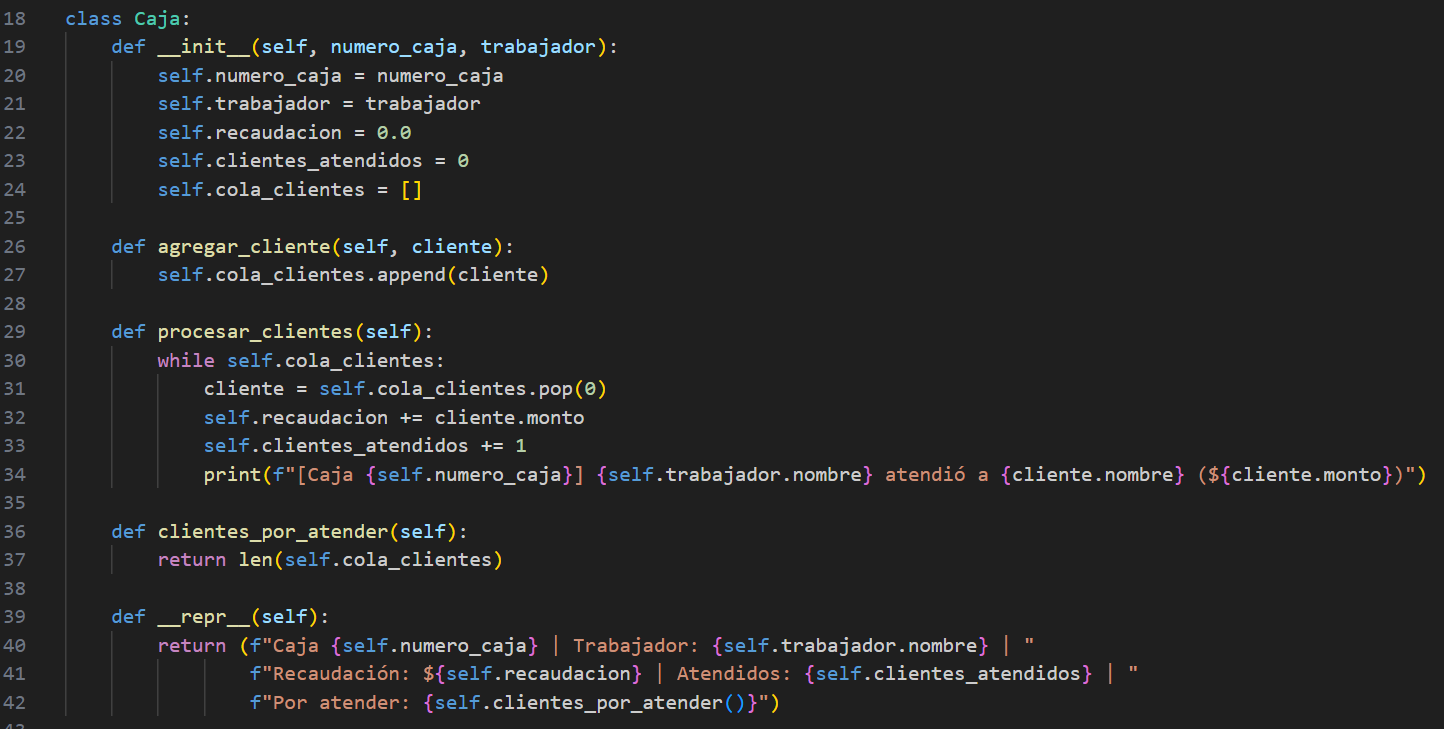
**Clase Caja**

Simula una caja registradora en donde los sus atributos son los siguientes:

* numero\_caja: identificador de la caja.
* trabajador: objeto de tipo Trabajador que atiende en la caja.
* recaudacion: total recaudado hasta el momento.
* clientes\_atendidos: número de clientes ya procesados.
* cola\_clientes: lista de clientes pendientes por atender.

Métodos:

* agregar\_cliente(cliente): agrega un cliente a la cola.
* procesar\_clientes(): atiende todos los clientes de la cola, sumando los montos a la recaudación.
* clientes\_por\_atender(): devuelve la cantidad de clientes en espera.



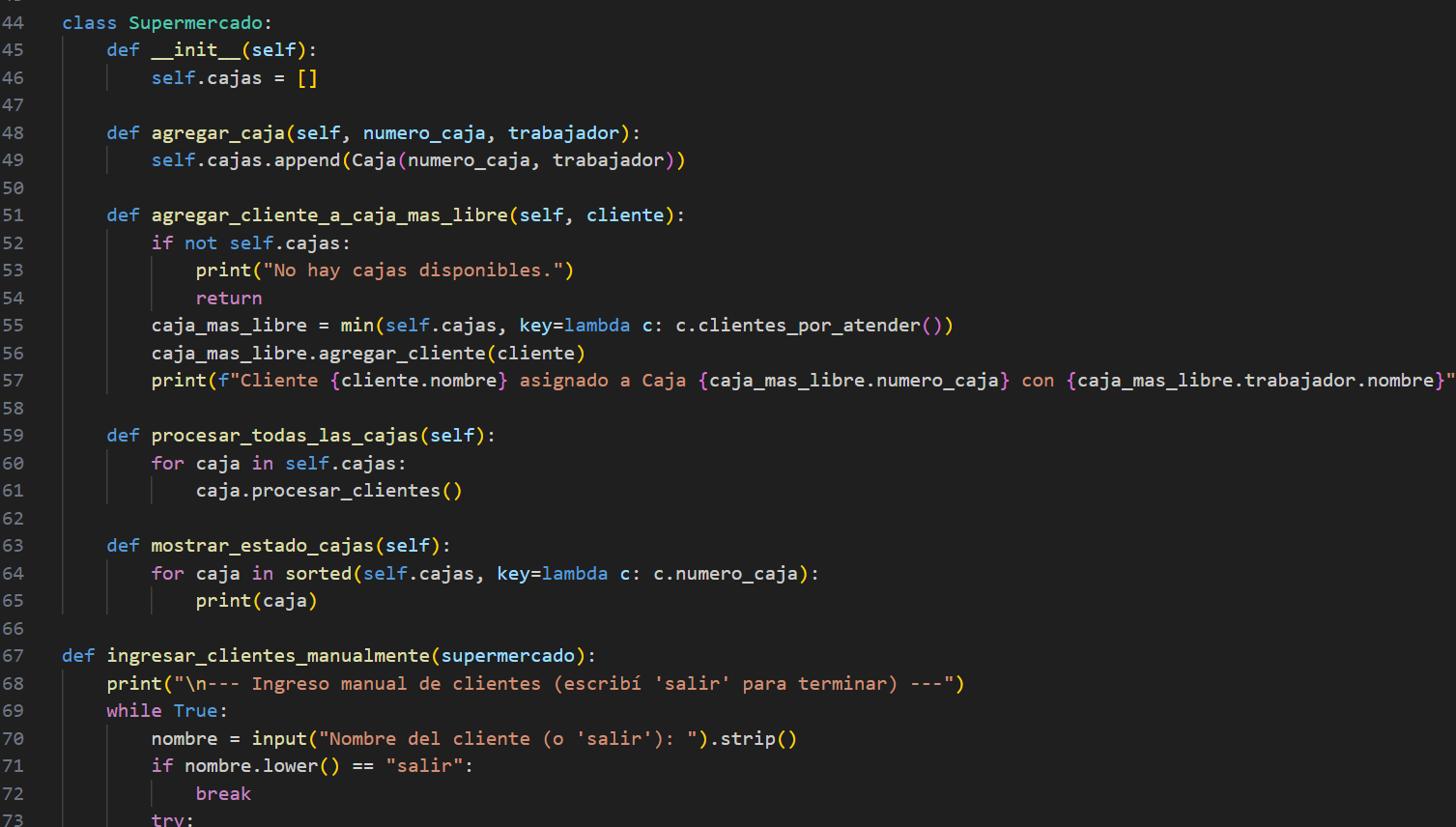
**Clase Supermercado**

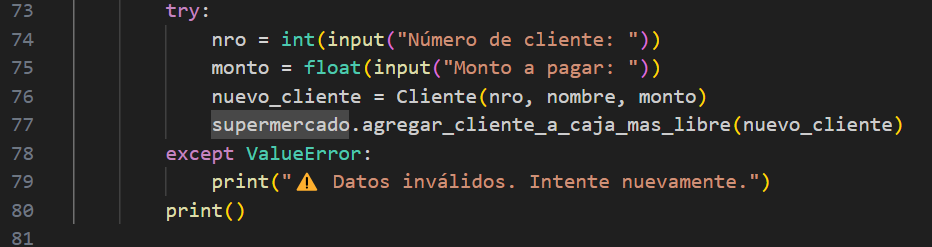
Esta clase administra todas las cajas del supermercado en donde los sus atributos son los siguientes:

* cajas: lista de objetos Caja.

Métodos:

* agregar\_caja(numero, trabajador): agrega una nueva caja..
* agregar\_cliente\_a\_caja\_mas\_libre(cliente): busca la caja con menos clientes y le asigna el cliente.
* procesar\_todas\_las\_cajas(): llama al método procesar\_clientes() de todas las cajas
* mostrar\_estado\_cajas(): imprime la información actual de cada caja (clientes, recaudación, etc.).





Con este código ingresamos manualmente los clientes desde la consola y el cliente se asigna automáticamente a la caja con menos personas en espera y el bucle se repite hasta que se escriba la palabra salir

En donde se crearán varias cajas con distintos trabajadores y se agregarán clientes de prueba donde se permite el ingreso de nuevos clientes por teclado para que finalmente se procesen las cajas y se muestra el estado final

