

## Proyecto final

# Simulador Supermercado

Kevin Ariel Merino Peña<sup>1</sup>  
Armando Abraham Aquino Chapa<sup>2</sup>

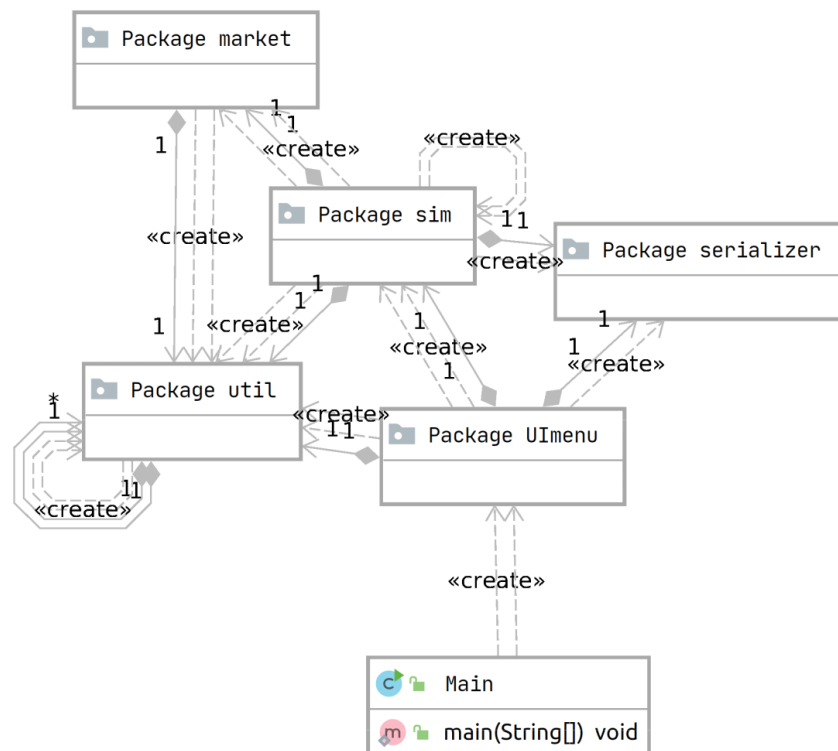
### Modelo del programa:



Figura 1: Simbología para los diagramas de clases

A continuación se presentará una breve descripción del modelo del programa para la simulación del comportamiento de un supermercado con sus respectivos diagramas de clases.

**Main:** Contiene nuestro método principal que permite cumplir con el encapsulamiento y comportamiento del programa.





























---

<sup>1</sup>317031326 $^2 317058163$

**Package UImenu:** Contiene nuestra clase *Menu* dónde modelamos la interacción con la usuaria. Entre sus métodos principales se encuentran:

- *mainMenu*: Cómo su nombre lo indica se encarga de proveer dos opciones principales; "Modo Usuario", "Modo Administrativo" por supuesto una opción para salir.
- *adminMenu*: Para uso administrativo del supermercado, en el cual podemos agregar, resurtir y ver el inventario de productos del mismo.
- *userMenu*: Aquí encontramos las funciones de las que podrá hacer uso cualquier persona, como hacer una simulación automática del supermercado, ó una simulación más personalizada ingresando el número de cajas rápidas y normales.
- *execPlot*: Se crean gráficas a partir del resultado de las simulaciones del supermercado para apreciar el comportamiento y rendimiento de las cajas del mismo para estudios posteriores.

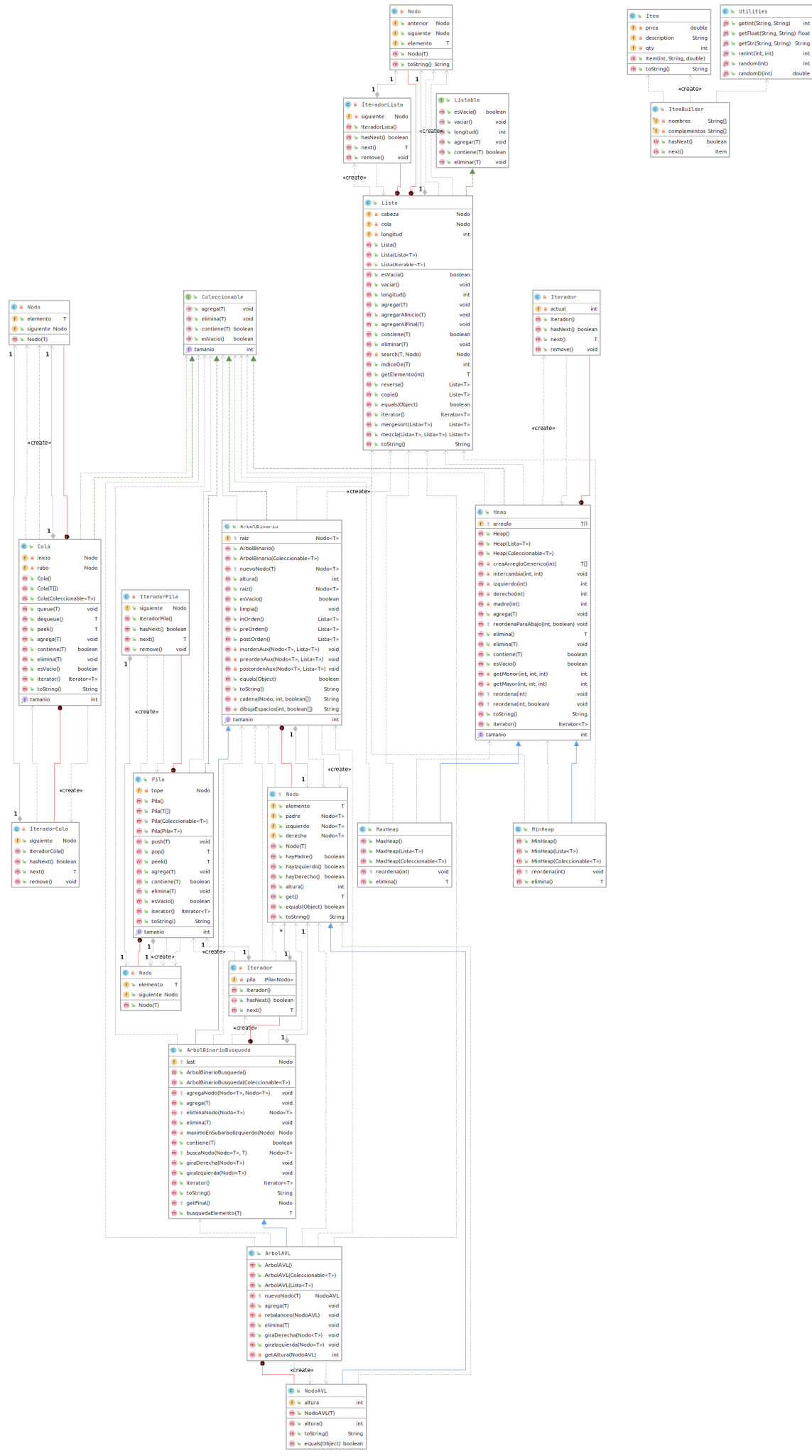
		Menu	
		serializer	Serializer
		simulation	Simulation
		plotting	Lista<String>
		Menu()	
		mainMenu()	void
		save()	void
		checkExistences()	void
		adminMenu()	void
		userMenu()	void
		addProduct()	Product
		resurtirProducto()	boolean
		execPlot()	void

**Package util:** En este paquete encontramos diversas clases, entre las cuales destacan las estructuras de datos utilizadas para modelar el supermercado y todo lo que lo engloba. Las estructuras de datos utilizadas son: *Árbol Binario*, *de Búsqueda*, *Árbol AVL*, *Cola*, *Heap*, *Lista*, *Pila*.

También encontramos una clase llamada *Utilities*, la cuál tiene la funcionalidad de contar con métodos auxiliares de la clase *Menu* para que el programa sea persistente.

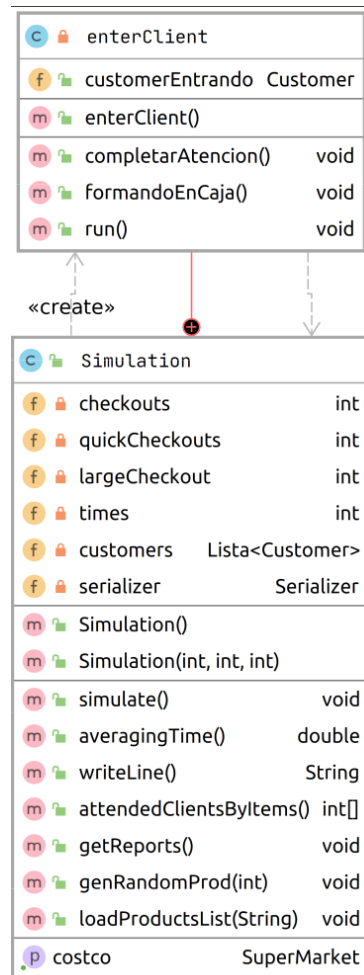
Por último encontramos dos clases llamadas *Item*, *ItemBuilder* que generan productos aleatorios para el supermercado.

A continuación la representación en diagrama de clases.



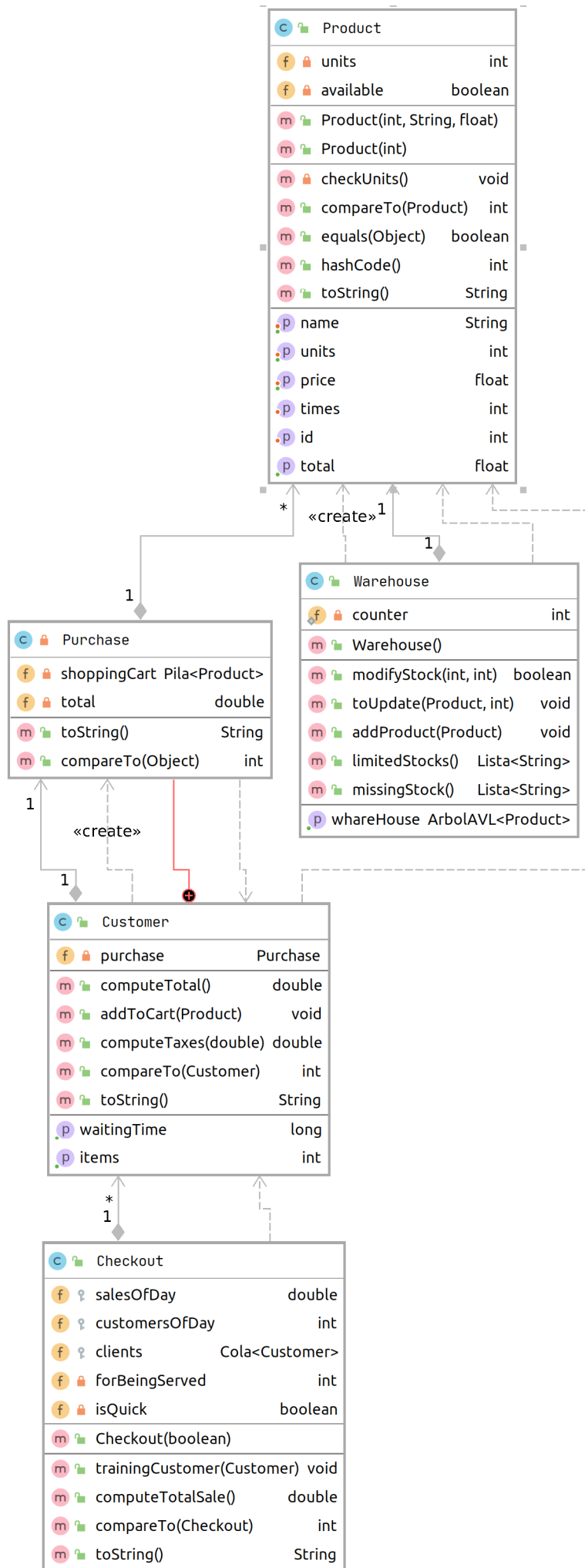
**Package sim:** Este paquete contiene las clases necesarias que se encargan de gestionar la simulación del supermercado a través de hilos.

- *enterClient*: Simula el estado de un cliente en el supermercado, desde su ingreso a éste, formarse en alguna caja y pagar sus productos.
- *Simulation*: Engloba y ejecuta todos los aspectos necesarios para el comportamiento del supermercado. Un método a destacar es *getReports* el cuál crea diversos directorios dónde se almacena el estado del supermercado después de las simulaciones, es decir, que productos tienen pocas existencias, los tickets generados de las diversas compras de los clientes, etc.

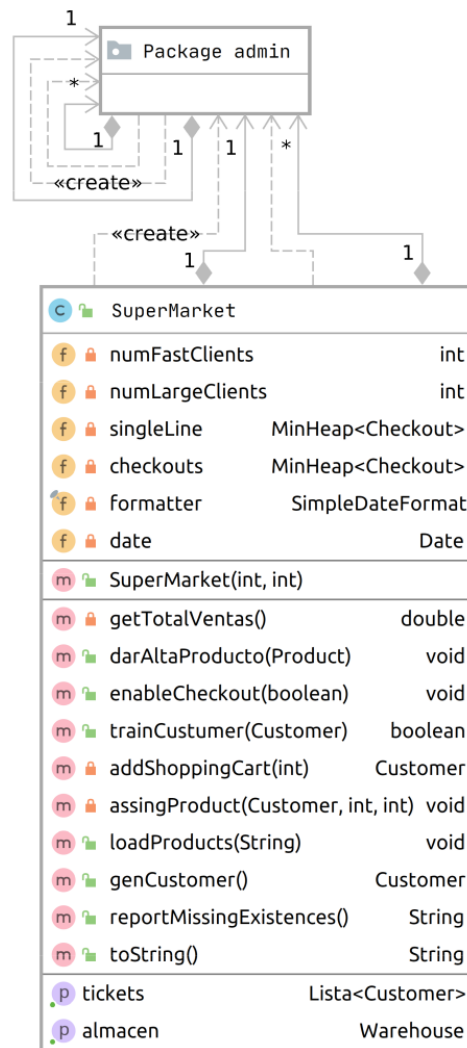


### Package market:

- **Package admin:** Aquí se encuentran las clases para modelar los objetos que componen a un supermercado, tales como sus cajas (hay dos tipos: normal y rápida), sus productos (los cuales poseen un ID generado aleatoriamente, unidades en el inventario, nombre, precio) y un almacén dónde podemos consultar el inventario, modificar, ver los productos ue contiene, etc.



Por último tenemos la clase *Supermaket*: que permite relacionar del paquete *admin* y le otorga una estructura al supermercado.



Reporte de resultados: