# Isabella Montiel Reyes

#### Producto

- id: int
- nombre: string
- tipo: string
- + get\_id(): int
- + get\_nombre(): string
- + get\_tipo(): string
- + precio\_total(): double\*
- + to\_string(): string\*

#### **Producto Perecedero**

- precio\_unitario: float
- cantidad: int
- fecha\_caducidad: string
- + precio\_total(): float
- + to\_string(): string

#### **ProductoNoPerecedero**

- precio\_unitario: float
- cantidad: int
- marca: string
- + precio\_total(): float
- + to\_string(): string

#### ProductoBebida

- precio\_unitario: float
- cantidad: int
- contiene\_alcohol: bool
- + precio\_total(): float
- + to\_string(): string

#### ProductoLimpieza

- precio\_unitario: float
- cantidad: int
- tipo\_superficie: string
- + precio\_total(): float
- + to\_string(): string

#### Supermercado

- inventario[1000]
- num\_productos: int
- + agregar\_producto\_perecedero(...): void
- + agregar\_producto\_no\_perecedero(...): void
- + agregar\_producto\_bebida(...): void
- + agregar\_producto\_limpieza(...): void
- + mostrar\_productos(): void
- + mostrar\_productos(tipo: string): void
- + calcular\_total\_ventas(): float
- + calcular\_total\_por\_tipo(tipo: string): float

# **SUBCOMPETENCIAS:**

# Explicación del diseño y su relación con el problema

**Objetivo del proyecto**: Modelar un sistema para administrar productos en un supermercado, organizándolos por tipo, y calcular ventas.

# **Clases principales:**

- **1.Producto (abstracta)**: Representa cualquier tipo de producto. Define interfaz común (precio\_total, to\_string).
- **2.ProductoPerecedero**: Añade fecha\_caducidad. Representa frutas, lácteos, etc.
- **3.ProductoNoPerecedero**: Añade marca. Ej: arroz, frijoles.
- **4.ProductoBebida**: Añade si contiene alcohol. Ej: jugo, cerveza.
- **5.ProductoLimpieza**: Añade tipo de superficie. Ej: detergentes, limpiavidrios.
- **6.Supermercado**: Administra un inventario de productos, permite agregar y mostrar productos y calcular totales.

# Casos que podrían hacer que falle:

- •Agregar productos con precio o cantidad negativa (se valida en código).
- •Exceder el límite de 1000 productos.
- •No liberar memoria (se maneja en el destructor).