



PROYECTO: GESTIÓN DE VENTAS EN UNA PASTELERÍA

Curso: Programación avanzada

Docente: MSc. Ing. Hugo Manuel Barraza Vizcarra

Integrantes:

Diesdanderson Dudu Vela Laurente 2025-119060

Elvis Juan Yana Cayo 2025-119048

Wilfredo Valdivia Vera 2025-119007





Antecedentes

En el Perú, muchos emprendimientos de repostería, principalmente liderados por jóvenes, gestionan sus procesos de forma manual, lo que genera errores y dificultades en la administración. Por ello, es necesario implementar sistemas informáticos simples que optimicen la gestión de inventarios y ventas y fortalezcan la sostenibilidad del negocio.



Planteamiento del problema

PROBLEMA

Los emprendimientos de repostería en el Perú, especialmente los liderados por jóvenes, presentan deficiencias en la gestión de inventarios y ventas debido al uso de métodos manuales. Esta situación genera errores en el control de stock, pérdida de información y dificultades en la planificación de la producción.

JUSTIFICACIÓN

La implementación de un sistema informático de gestión se justifica por la necesidad de automatizar y organizar los procesos internos del emprendimiento. Un sistema permitirá reducir errores humanos, mejorar el control de insumos y facilitar la toma de decisiones.

ALCANCE

El sistema permitirá controlar el stock actual y mínimo, registrar la producción mediante recetas, administrar insumos en el almacén y registrar las ventas, incluyendo la generación y almacenamiento de boletas.



OBJETIVOS

GENERAL

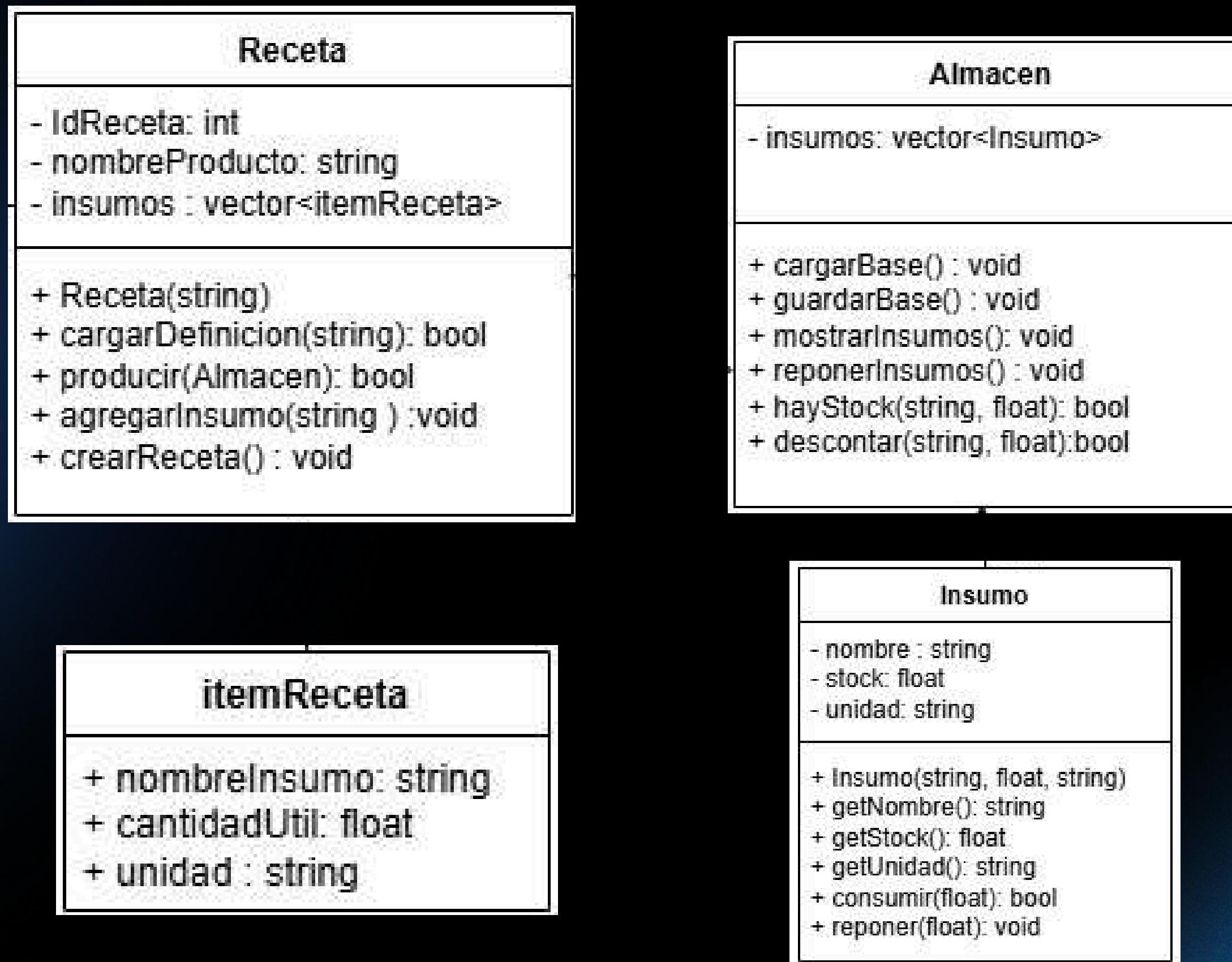
Desarrollar un sistema basado en programación orientada a objetos que automatice la gestión de inventario, producción y ventas de un emprendimiento de repostería, mejorando la organización y la toma de decisiones.

ESPECÍFICOS

El proyecto consiste en aplicar principios de Programación Orientada a Objetos, diseñar el sistema mediante diagramas UML e implementar la gestión de productos, ventas y control de stock, incluyendo la generación y almacenamiento de boletas de venta.



```
"container">
  <div class="row">
    <div class="col-md-6 col-lg-8"> <!-- Main Content -->
      <nav id="nav" role="navigation">
        <ul>
          <li><a href="index.html">Home</a></li>
          <li><a href="home-events.html">Home Events</a></li>
          <li><a href="multi-col-menu.html">Multi Column Menu</a></li>
          <li class="has-children"> <a href="#">Tall Buttons & Headers </a>
            <ul>
              <li><a href="tall-button-headers.html">Tall Button Headers</a></li>
              <li><a href="image-logo.html">Image Logo</a></li>
              <li class="active"><a href="#">Active Tall Buttons & Headers</a></li>
            </ul>
          </li>
          <li class="has-children"> <a href="#">Variable Width Sections & Grids </a>
            <ul>
              <li><a href="variable-width-sections.html">Variable Width Sections</a></li>
            </ul>
          </li>
        </ul>
      </nav>
    </div>
    <div class="col-md-6 col-lg-4"> <!-- Sidebar -->
      <div>
        <h3>Recent Posts</h3>
        <ul>
          <li>Recent Post One</li>
          <li>Recent Post Two</li>
          <li>Recent Post Three</li>
        </ul>
      </div>
    </div>
  </div>
</div>
```





Inventario
- productos : vector<Producto*> - contadores : map<char,int> - formatearId() : string
+ agregarProducto(Producto*): void + actualizarStock(): void + producir + buscarPorId(int) : Producto* + venderProducto (int,int) : void + mostrarProductos() : void + cargarProductos() : void + guardarProductos() : void + mostrarBajoMinimo() : void

Producto
- idProducto: string - nombre: string - stockActual : int - stockMinimo : int - precio : float
+ Producto() + Producto(string, int , int) + virtual ~Producto() = default + getStockActual(): int + getStockMinimo(): int + getPrecio(): float + getIdProducto(): string + getNombre(): string + setId(string): void + {virtual} actualizarStock(int) = 0: void + {virtual} prefijoId() = 0: char + {virtual} mostrarDatos(): void + {virtual} ssdatos(): string + {virtual} cargarDesdeStream(stringstream&) = 0: void + producir(int cant, Almacen&) : int + estaBajoMinimo() : bool + mostrarDatos() : void

Torta
- maxPorciones : int
+ Torta() + Torta(string, int, int, int, float) + getMaxPorciones(): int + actualizarStock(int) : void + ssdatos() : string + mostrarDatos() : void + cargarDesdeStream() void + producir() : int + prefijoId() = "T": char

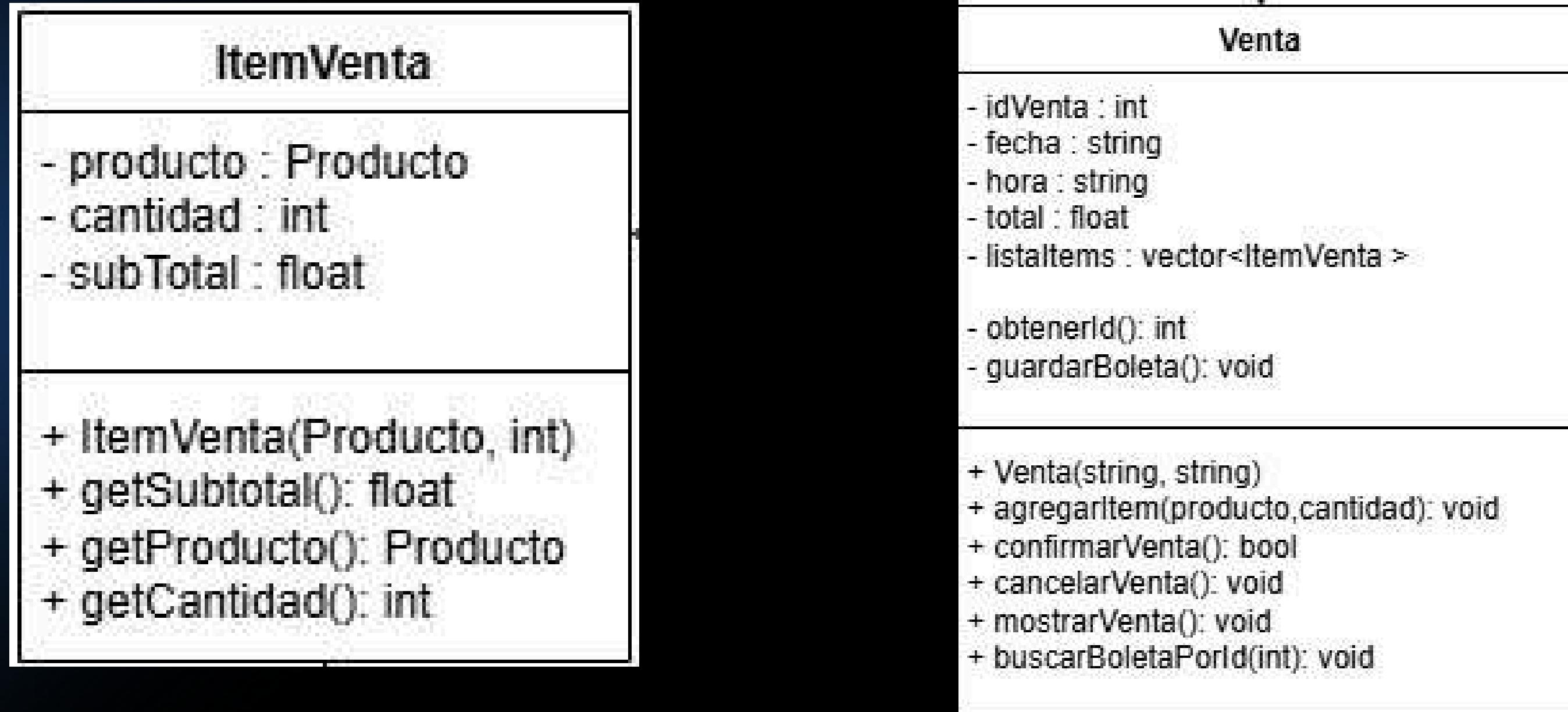
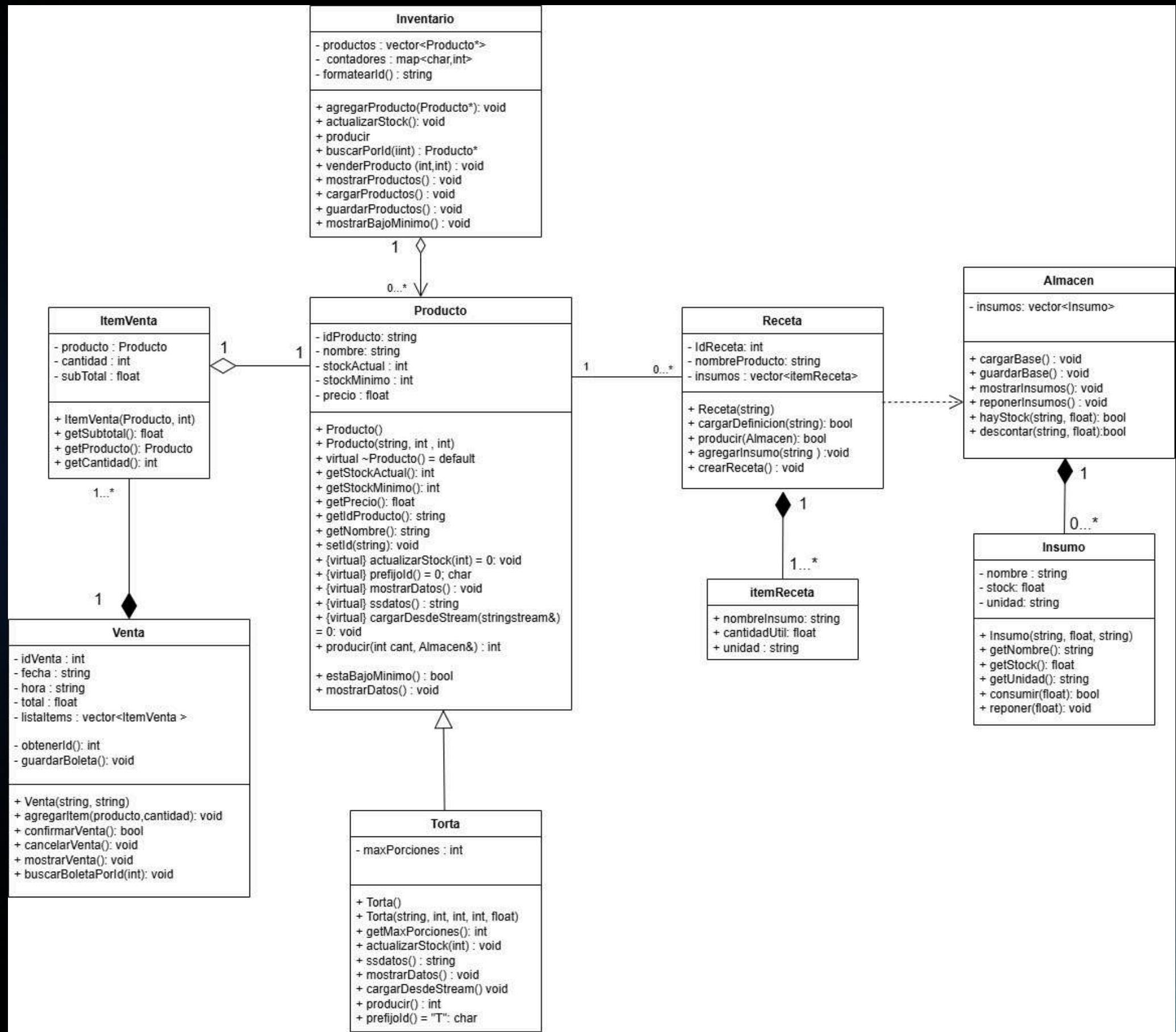


DIAGRAMA UML





THANK YOU