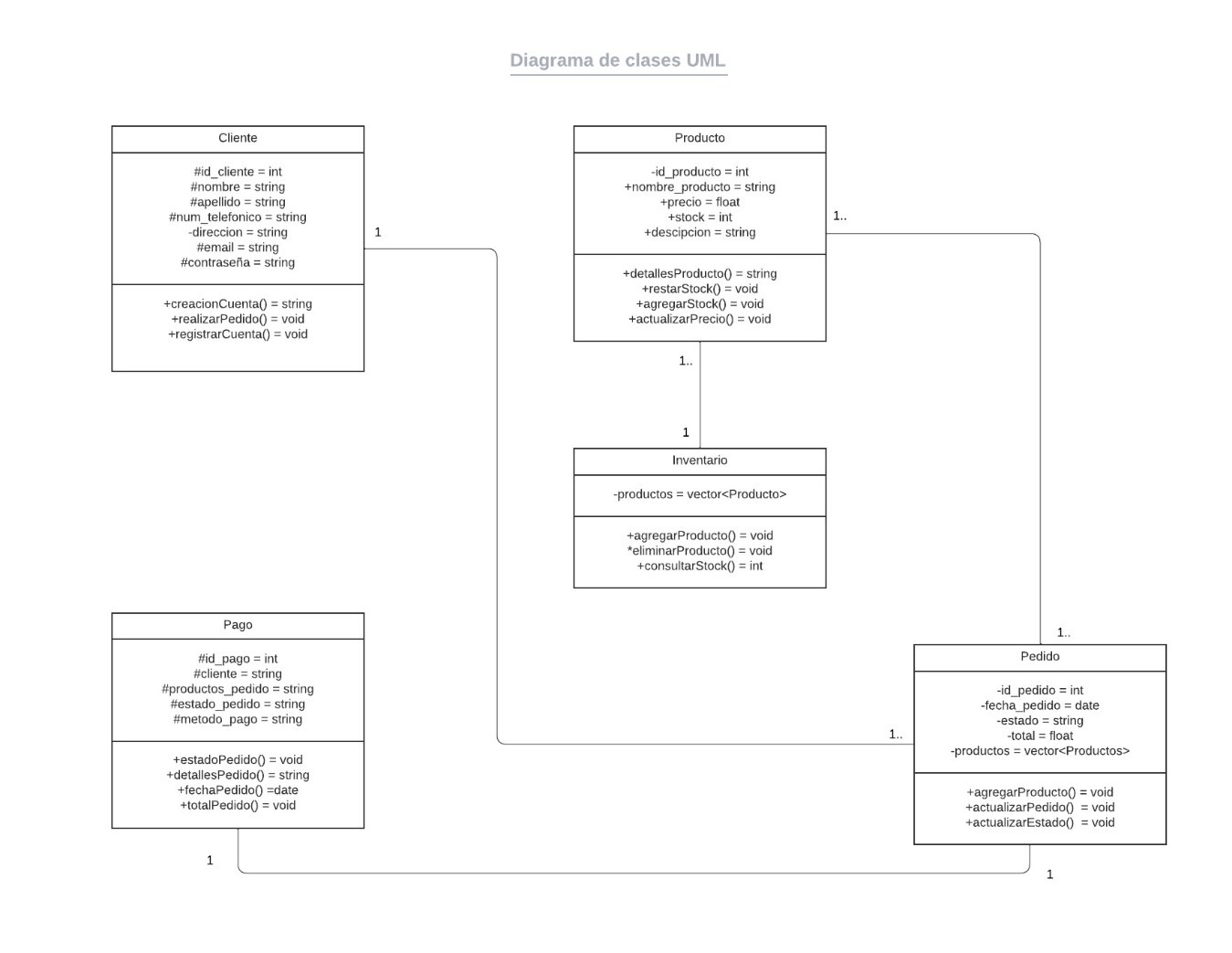
***Diagrama de clases***

Para el diseño de software, se elaboró un diagrama de clases Unified Modeling Language (UML), en el cual, proporcionará una explicación detallada de cada clase establecida en el diagrama, con sus respectivos, métodos, así como sus relaciones. Este sería el diagrama de clases:



Desglosando el diagrama:

* Existen 5 clases, las cuales serían cliente, producto, inventario, pago y pedido.

Cada clase esta compuesta de la siguiente manera:

* La clase ‘Cliente’ cuenta con todos sus atributos de tipo de protección (#), incluyendo el ‘id\_cliente’ de tipo entero, ‘nombre’, ‘apellido’, ‘num\_telefonico’, ‘direccion’, ‘email’ y ‘contraseña’ de tipo cadena de caracteres. Sus métodos serian 3 y todos de tipo de protección público (+), el método ‘creacionCuenta()’ retorna una cadena caracteres, y los métodos ‘realizarPedido()’ y ‘registrarCuenta()’ que son de tipo vacío.
* La clase ‘Producto’ se tienen atributos de visibilidad publica, los cuales son: ‘id\_producto’, y ‘stock’ de tipo entero, seguido de ‘nombre\_producto’, y ‘descripcion’ de tipo cadena de caracteres y ‘precio’ de tipo flotante. Los métodos que componen la clase son todos de visibilidad publica, ‘detallesProducto()’ retorna una cadena de caracteres, mientras que los métodos ‘restarStock()’, ‘agregarStock()’ y ‘actualizarPrecio()’ son de tipo vacio.
* La clase ‘Inventario’ este compuesto por un solo atributo de visibilidad privada (-), el cual es ‘productos’ que es un vector que contiene el objeto Producto. Y tiene 3 métodos, 2 de ellos de tipo vacío, los cuales serían: ‘agregarProducto()’ y ‘eliminarProducto()’, mientras que el método de ‘consultarStock()’ retorna un entero.
* La clase ‘Pago’ esta compuesto por atributos de visibilidad protegida, los cuales son: ‘id\_pago’ de tipo entero, ‘cliente’, ‘productos\_pedido’, ‘estado\_pedido’ y ‘metodo\_pago’ de tipo cadena de caracteres, teniendo de métodos ‘estadoPedido()’ y ‘totalPedido()’ de tipo vacio, ‘detallesPedido()’ que retorna una cadena de caracteres, y ‘fechaPedido()’ que retorna una variable de tipo fecha.
* Y, por último, la clase ‘Pedido’ esta compuesto por los atributos: ‘id\_pedido’ de tipo entero, ‘fecha\_pedido’ de tipo date, ‘estado’ de tipo cadena de caracteres, ‘total’ de tipo flotante y ‘productos’ un vector que contiene objetos de la clase Producto. Sus métodos de visibilidad publica y todos de tipo vacío: ‘agregarProductos()’, ‘actualizarPedido()’ y ‘actualizarEstado()’.

Una vez teniendo las clases definidas, las relaciones entre clases se definieron de la siguiente manera:

* La clase ‘Cliente’ está asociada con la clase ‘Pedido’.
* La clase ‘Pedido’ está asociada con la clase ‘Pago’ y la clase ‘Producto’.
* La clase ‘Inventario’ está asociada con la clase ‘Producto’.

Estas relaciones son bidireccionales, por lo que sus valores de multiplicidad serían los siguientes:

* Un cliente puede realizar varios pedidos, para cada pedido está asociada a un único cliente. Siendo una relación de 1 a muchos (1..).
* Un pedido puede contener múltiples productos, y un producto puede estar incluido en varios pedidos. Siendo una relación de muchos a muchos (n..m).
* Un inventario este compuesto por varios productos, pero cada producto pertenece a un único inventario. Siendo una relación de 1 a muchos (1..).
* Y, por último, cada pedido tiene asociada a un único pago, y cada pago corresponde a un único pedido. Siendo una relación de 1 a 1 (1..1).