Proyecto Integrador de Programación: Avance 2

Integrantes: Héctor Joaquín Pérez Can, Luis Jonathan Quevedo Zaldivar, Andrea Isabel Torres Pérez, Álvaro de Jesús Xool Canul.

Actividades:

1. Identificación o establecimiento del problema a resolver (Documentarlo).

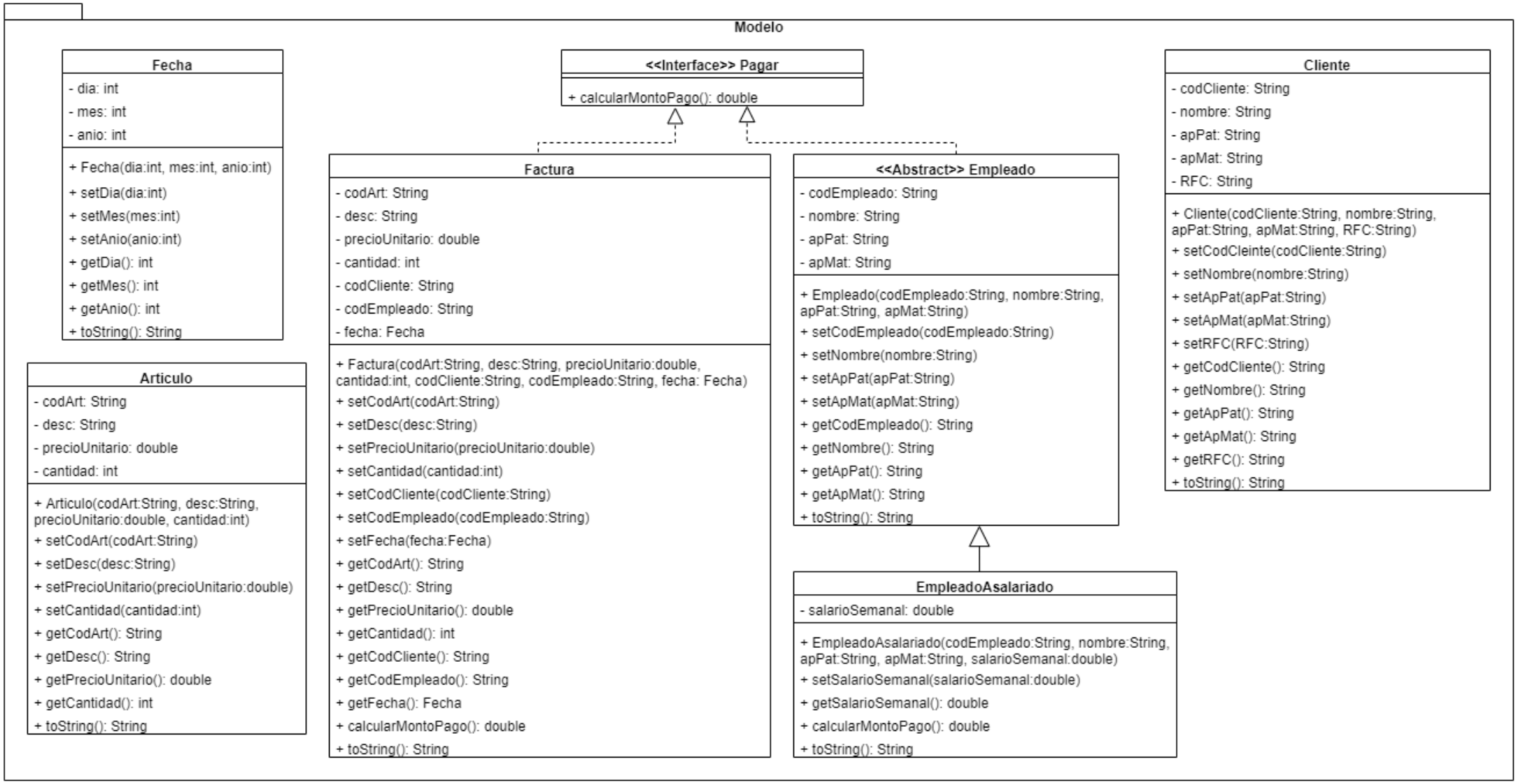
Se requiere un programa para una tienda que ayude a llevar un control de todos los artículos que se encuentran a la venta, además de poder agregar nuevos productos y aumentar el numero de unidades en caso de reabastecimiento. También es necesario que se pueda tener un registro de los empleados, así como de los clientes.

2. Aplicación del proceso de Análisis del problema, para identificar entradas, procesos y salidas necesarias para la resolución del problema. (requisitos documentados).

Debe tener:

1. Inventario: El sistema debe ser capaz de llevar un control de inventario de productos y poder agregar un nuevo producto.
2. Clientes: El sistema debe ser capaz de tener una lista de sus clientes, así como agregar un nuevo cliente.
3. Empleado: El sistema debe ser capaz de tener un registro de todos sus empleados.
4. Facturas: El sistema debe ser capaz de generar una factura de las compras del cliente.

3. Aproximación a una solución software.



4. Descripción dónde o como implementarían los temas de herencia, polimorfismo y/o interfaces (Documentación basada en el diagrama UML correspondiente).

Para las interfaces, se puede apreciar como la clase “Pagar” tiene definido un método abstracto, en esta clase solo se definirá el método “calcularMontoPago” pero no se implementará.

Lo siguiente que vemos en el diagrama de clases es la clase “Factura” que es una clase que utilizará el implements, en otras palabras, la interfaz “Pagar”.

Para la clase “Empleado”, al ser una clase abstracta, no es necesario implementar el método de la interface pero sí será obligatorio implementarlo en sus clase hija (EmpleadoAsalariado).

Por herencia, la clase “EmpleadoAsalariado” heredará de la clase padre (Empleado) todos los atributos y métodos para así reducir líneas de código.

El poliformismo se usa en la clase padre “Empleado” al ser abstracta y heredarle el método de la interfaz “Pagar” a su subclase.