



En general, podemos encontrarnos con clases que representan a un ente de la realidad, del contexto, si nos enfocamos en el mundo empresarial seguramente nos encontraremos con clientes y para resolver ciertos problemas necesitamos codificar la clase Cliente.

Un cliente tiene un número de cliente, un nombre y una deuda. Para crear un cliente, tengo que tener su número de cliente y su nombre. Un cliente puede incrementar su deuda o pagar el total de su deuda.

Actividad: Programar la clase Cliente teniendo en cuenta el siguiente UML.

Cliente
<ul style="list-style-type: none">- numeroCliente: Integer- nombre: String- deuda: Double
<ul style="list-style-type: none">+ Cliente(numero: Integer, nombre: String)+ incrementarDeuda(valor: Double): void+ pagarDeuda(): void