



El diagrama UML consiste en una plataforma para que el usuario pueda comprar boletos de avión en línea.

Comienza con la superclase **Usuario** la cual se conforma por los atributos de tipo protegido: nombre, correo electrónico e id del usuario. Asimismo, el método abstracto de verificar cuenta.

Posteriormente, tenemos las clases hijas **Cliente** y **Administrador**. En lugar de duplicar atributos para nombre, correo electrónico e id del usuario, convertimos la clase **Cliente** y la clase **Administrador** en subclases de la clase **Usuario**, que se convierte en superclase. Ahora existe una relación de herencia: estas subclases heredan todos los métodos y atributos de la superclase. Estas subclases también cuentan con sus respectivos atributos independientes a la superclase. En este escenario, también estamos hablando de abstracción: la clase **Usuario** es una clase abstracta porque en nuestro diagrama UML cada vez que se quiera crear una instancia de una de nuestras clases, esta será **Administrador** o **Cliente**. La clase **Usuario** es una forma de mantener simple el UML y ahorrar código. Por otro lado, tenemos la composición que involucran a las clases **Vuelos** y **Factura**. Esto quiere decir que los objetos derivados no podrían existir sin su objeto principal, este caso, la subclase **Cliente** y la clase **Vuelos**. En la clase **Factura** los atributos son número de factura, precio del boleto, número de asientos y numero de factura. Además, como métodos existe el actualizar información. En la clase **Vuelos** tenemos los atributos de lugar de partida, número de boleto y asiento, fecha del viaje, lugar de partida y lugar de destino. Como métodos tenemos el consultar información del vuelo y comprar el vuelo. Finalmente, tenemos el concepto demultiplicidad, lo cual nos permite definir restricciones numéricas en las relaciones creadas. Que van desde 0 a varios pedidos (**0..\*** a **1**) y una relación **1** a **1**: para cada compra corresponde solo un detalle del boleto