Se ha definido la clase Usuario la cual es abstracta. De ella heredarán diferentes clases para implementar diferentes tipos de usuarios con distintos roles. Está la clase Administrador, que ya se ha visto en el enunciado que va a tener permiso para realizar más gestiones con la base de datos que otras clases que implementen otros tipos de usuario.

A efectos prácticos, esa diferencia en la gestión de las peticiones a la base de datos, o a las funciones a las que cada perfil esté autorizado a ejecutar en general, en función del rol se llevarán a cabo mediante el polimorfismo visto en clase. Así un objeto de la clase Administrador podrá implementar ciertas queries por ejemplo sobre la base de datos y otro de la clase Cliente, (también subclase de la clase abstracta Usuario), mediante el polimorfismo lanzará una excepción, (también visto en clase), en el mismo método si el rol no está autorizado a hacer esas consultas.

Otro ejemplo de entidades que se están trabajando dentro del paradigma de la POO es la clase Operación la cual, como se comenta en el enunciado es la base de todo el flujo de información. Esta clase sí que se ha decidido implementarla frente a hacerla abstracta o una interfaz porque se ha considerado que una implementación para esta clase tenía sentido. Sin embargo, respecto de esta clase a la que se añaden las distintas rutas y la información en general del flujo de mercancías vinculado a esta operación, se ha hecho que otras clases, como Facturación, hereden de ella, ésta última por ejemplo para añadir la funcionalidad concreta de las transacciones monetarias vinculadas a la operación en cuestión.

Finalmente, el otro gran grupo de entidades que se ha trabajado en esta actividad han sido las que contienen información de las rutas. Se ha creado la clase Ruta como una clase abstracta ya que se considera que una Ruta no tiene sentido en sí misma. Es sólo el resultado de aglutinar características comunes de las rutas marítimas, terrestres o aéreas pero nunca va a ser necesario que exista en el código un objeto de la clase código como tal. Las clases RutaAerea, RutaTerrestre y RutaMaritima heredan, por tanto, todas ellas de la clase abstracta Ruta y, por un lado añaden parámetros característicos de cada una de las rutas en forma de atributos, (la ruta marítima tiene un booleano para indicar si el contenedor es open top o no mientras que la ruta aérea tiene otro para indicar si el bulto viaja de pie), y por el otro utilizan el polimorfismo para que cada método de la clase abstracta Ruta se implemente de forma diferente según si el trayecto es aéreo, marítimo o terrestre.