La DGT nos ha pedido ayuda para gestionar el carnet por puntos. Los conductores están identificados de manera unívoca por su DNI y la cantidad de puntos de un conductor está entre 0 y 15 puntos inclusive. La implementación del sistema se deberá realizar como un TAD `CarnetPorPuntos` con las siguientes operaciones:

- `crea()`: Crea un sistema de gestión de carnets por puntos vacío.
- `nuevo(dni)`: Añade un nuevo conductor identificado por su DNI (un string), con 15 puntos. En caso de que el DNI esté duplicado, la operación lanza un error (excepción `EConductorDuplicado`).
- `quitar(dni, puntos)`: Le resta puntos a un conductor tras una infracción. Si a un conductor se le quitan más puntos de los que tiene, se quedará con 0 puntos. En caso de que el conductor no exista, lanza un error (excepción `EConductorNoExiste`).
- `recuperar(dni, puntos)`: Le añade puntos a un conductor enmendado. Si debido a una recuperación un conductor supera los 15 puntos, se quedará con 15 puntos. En caso de que el conductor no exista, lanza un error (excepción `EConductorNoExiste`)
- `consultar(dni)`: Devuelve los puntos actuales de un conductor. En caso de que el conductor no exista, lanza un error (excepción `EConductorNoExiste`).
- `cuantos_con_puntos(puntos)`: Devuelve cuántos conductores tienen un determinado número de puntos. En caso de que el número de puntos no esté entre 0 y 15 lanza un error (excepción `EPuntosNoValidos`).
- `lista_por_puntos(puntos)`: Produce una lista con los DNI de los conductores que poseen un número determinado de puntos. La lista estará ordenada por el momento en el que el conductor pasó a tener esos puntos, primero el que menos tiempo lleva con esos puntos. En caso de que el número de puntos no esté entre 0 y 15 lanza un error (excepción `EPuntosNoValidos`).

La implementación de las operaciones debe ser lo más eficiente posible. Por tanto, debes elegir una representación adecuada para el TAD, implementar las operaciones y justificar la complejidad resultante.