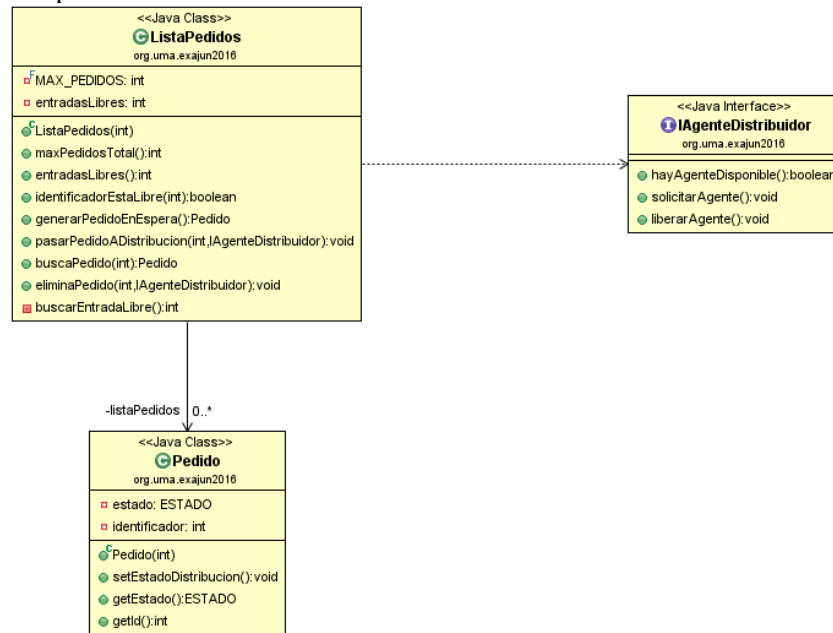


Enunciado del ejercicio de testing

El almacén de logística de la tienda online “Orinoco” utiliza una *lista de pedidos* unificada para gestionar todos los pedidos que se realizan en el sitio *e-commerce* en un momento determinado. Esta lista de pedidos tiene un tamaño acotado, que se delimita por el número máximo de pedidos que la empresa es capaz de atender en cierto periodo de tiempo. Un pedido puede estar en dos estados: en ESPERA, cuando se hace el pedido y se busca sitio en la lista, y en DISTRIBUCION, cuando tras estar en ESPERA, ha sido asignado a un *agente de distribución*.

El funcionamiento de la lista es como sigue: cuando se va a generar un pedido, el método **generarPedidoEnEspera()** busca el primer hueco libre de la lista a partir de la posición cero y se lo asigna al nuevo pedido, inicialmente en estado de ESPERA. La posición en la lista será el identificador del pedido.



El paso del estado en ESPERA a DISTRIBUCION se realiza mediante el método **pasarPedidoADistribucion()**, que accede a un objeto que implementa la interfaz **IAgenteDistribuidor** que gestiona las operaciones de logística. Otros métodos de la lista son **eliminaPedido()**, que dado un identificador de pedido borra su entrada en la lista y, si estaba en distribución, libera al agente correspondiente; **buscaPedido()**, que dado un identificador de pedido lo busca en la lista, y **identificadorEstaLibre()** que comprueba si un identificador de pedido (posición en la lista) está disponible. El diagrama de clases del enunciado representa las relaciones entre las clases que implementan la aplicación de logística.

SE PIDE: A partir de la implementación de las clases que está disponible en el campus virtual y de una plantilla de la clase **ListaPedidosTest**, escribir el código de los siguientes casos de prueba:

1. *siSeGeneraUnPedidoEnEsperaElEspacioLibreDeLaListaDisminuyeEnUno()*
2. *siSeGeneraUnPedidoEnEsperaYLaListaEstaVacíaSuIdentificadorEsCero()*
3. *siSeGeneraUnPedidoEnEsperaYLaListaEstaLlenaSeElevaUnaExcepcion()*
4. *siSePasaUnPedidoADistribucionYHayAgentesEntoncesElEstadoDelProcesoEsDistribucion()*
5. *siSePasaUnPedidoADistribucionYNoHayAgentesEntoncesElEstadoDelProcesoEsEnEspera()*
6. *siSePasaUnPedidoADistribucionYSeEliminaEntoncesSeHaLiberadoAlAgenteDistribuidor()*
7. *siSeEliminaUnPedidoEnEsperaNoSeLiberaNingunAgenteDistribuidor()*
8. *siSeBuscaUnPedidoQueSeHaEliminadoSeElevaUnaExcepcion()*
9. *siSeEliminaPedidoYNoExisteSeElevaUnaExcepcion()*
10. *siSeEliminaPedidoExistenteEntoncesSuIdentificadorQuedaLibre()*
11. *siSePasaUnPedidoADistribucionYHayAgentesEntoncesSeSolicitaAgente()*

En todos aquellos tests en los que se colabore con un agente distribuidor, se debe emplear JMockit para verificar la interacción. En los casos en los que se eleve una excepción, ésta será de la clase **ListaPedidosException**.