REALIZACION DE UML  
Nos encontramos en charlas constantes para evaluar el TP, leyéndolo tanto individual como grupalmente en un lapso aproximado de 2 o 3 dias, realizando de manera gradual el UML considerando para nosotros las mejores opciones de diseño dentro del marco que nos indica la metodología SOLID

PATRON STATE (fase de buques)

Las terminar dependiendo de la fase de los buques tiene que enviar en total mensajes a los clientes y 2 mensajes a los buques  
Inicia en esta Outbound, al estar a menos de 50 Km pasa al siguiente estado   
En estado Inbound el buque le avisa a la terminar y esta le avisa a los Importadores  
En estado Arrived la terminal gestionada le avisa al buque (carga y descarga)  
En estado Working la terminal gestionada enviar un mensaje “depart” y buque cambia al siguiente estado  
En estado Departing el buque avisa a la terminal y esta avisa a los exportadores  
Se consideró usar un patrón State, ya que es necesario que Buque vaya cambiando de estado según se cumplan ciertas condiciones y los estados se conocen entre ellos, esto nos permitiría en un futuro poder agregar otro esta de ser necesario.  
**-Buque cumple el rol de Contexto  
-EstadoBuque cumple el rol de Estado  
-Clases implementadas por EstadoBuque cumplen el rol de EstadoConcreto**

PATRON STRATEGY (Mejor circuito)  
Como una de las funcionalidades de la terminal portuaria es devolver el mejor circuito entre un puerto origen y un puerto destino, en base a distintos criterio para determinar este circuito consideramos el uso de un patrón Strategy ya que cada algoritmo de búsqueda es distinto y no se conocen entre ellos.  
**- MejorRuta cumple el rol de Estrategia  
-Subclases de MejorRuta cumplen el rol de EstrategiaConcreta  
-TerminalPortuaria cumple el rol de cliente**

PATRON COMPOSITE (Busqueda marítima)   
En esta búsqueda es necesario evaluar diferentes elementos, donde algunos (elemento compuesto) pueden contener otros del mismo tipo o elementos de comportamiento sencillo, relacionándose a través de operadores booleanos lo que nos da a la idea de una estructura de árbol y utilizar la recursión para obtener el resultado esperado, por estas razones decidimos utilizar un patrón composite  
**-Componente cumple el rol de Componente  
-OperadorAND y OperadorOR cumplen el rol de Compuesto  
-Filtros cumplen el rol de Hoja**

PATRON OBSERVER (envio de emails de terminal a clientes)  
Dependiendo de la fase de buque, este envía un mensaje de aviso a la terminal y a su vez, este último envía un mensaje a diferentes destinatarios(Clientes), cuando llega el buque se le debe avisar a los importadores relacionados a ese viaje y cuando el buque sale del puerto gestionado se le debe avisar a los exportadores que su carga partió, por esta razón se tomó la decisión de usar un patrón observer para poder enviar el mensaje al destinatario correspondiente  
**-Cliente cumple el rol de observador**