

<b>Nombre:</b>	
<b>Código:</b>	<b>Fecha:08/Abril/2019</b>

### Enunciado

La mayor parte de las importaciones y exportaciones que realiza nuestro país viajan a través de puertos marítimos, y Cali, esta ubicada cerca del puerto marítimo más importante de Colombia, el puerto marítimo de Buenaventura. La empresa “Camarón que se Duerme” que controla las operaciones en Buenaventura quiere poder administrar la información de los contenedores que llegan a su puerto. Por esto, la empresa ha contactado a Ragnar Lodbrok estudiante del curso APO1 y le ha solicitado ayuda para crear un programa sencillo que les ayude a resolver sus requerimientos.

Del puerto administrado por la empresa es importante conocer el nombre, la dirección, el teléfono y el nit de la empresa. El puerto es responsable de administrar muchos contenedores, pero de cada contenedor requieren conocer la siguiente información: el identificador único del contenedor, compuesto por letras y números, el peso, el tipo de contenedor (esta empresa maneja 3 tipos, Dry Van o contenedor seco de 20' y 40', y Tank o contenedor cisterna para transportar líquidos), una descripción general de los productos que viajan en el contenedor, un indicador de si el contenedor sale o entra al puerto, días que lleva en el puerto y un costo que depende del indicador, si sale del puerto, el costo es calculado como \$2'000.000 \* el peso registrado del contenedor más los días que lleve en el puerto por \$100.000 y si entra al puerto, el costo es calculado como los días que lleva en el puerto \* \$150.000.

Por el momento se necesita que usted implemente el modelo del primer prototipo modelado por Ragnar , el cual se puede ver en la figura 1.

Los requerimientos para este primer prototipo son:

<b>Nombre:</b> R1 - Agregar un contenedor al puerto
<b>Resumen:</b> Permite registrar un contenedor al puerto, ya sea que vaya de salida o entrada al país.
<b>Entradas:</b> El identificador único del contenedor, El peso, El tipo, contenedor va de entrada o salida, los días y el costo.
<b>Resultado:</b> Se añade un nuevo contenedor.

<b>Nombre:</b> R2 - Listar contenedores que van a salir del puerto
<b>Resumen:</b> Lista los contenedores que van a salir del país a través del puerto de la empresa.
<b>Entradas:</b> Ninguna
<b>Resultado:</b> Una lista con los contenedores que van a salir del país.

<b>Nombre:</b> R3 - Consultar la cantidad de contenedores tipo cisterna que entraron al puerto.
<b>Resumen:</b> Consulta la cantidad de contenedores tipo cisterna que entraron al país a través del puerto de la empresa.
<b>Entradas:</b> ninguna
<b>Resultado:</b> El número de contenedores tipo cisterna que entraron al puerto.

## Diagrama de Clases:

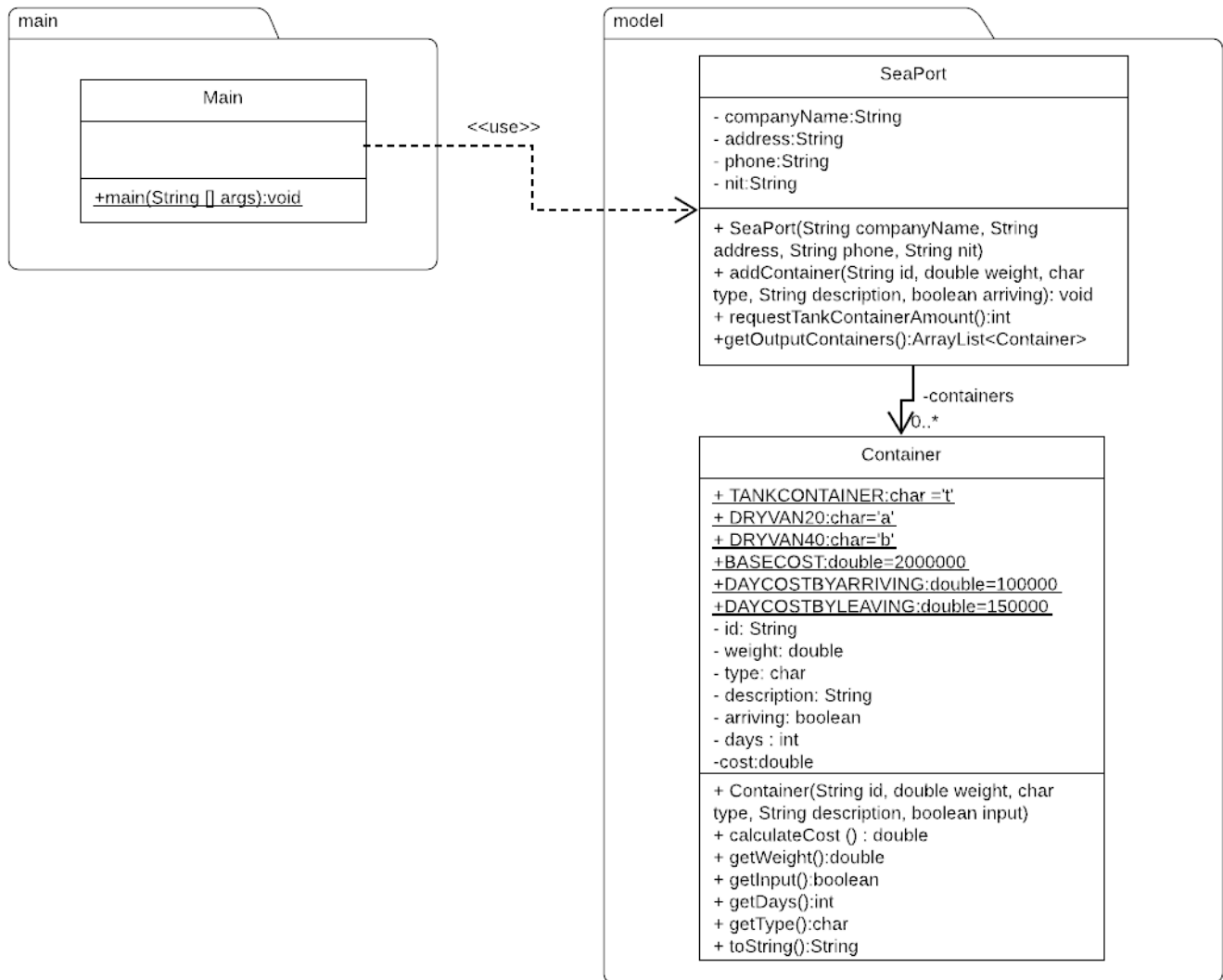


Figura1. Diagrama de clases

### Datos de prueba:

Utilice el método `addContainer` de la clase `SeaPort` desde la interfaz para agregar por lo menos 3 contenedores y con ellos probar la funcionalidad del aplicativo.

### Actividades

1. [15%] Implementa correctamente en Java el modelo del mundo.
2. [15%] Implementa correctamente en Java la interfaz de usuario.
3. [10%] Realiza los contratos de las clases del modelo.
4. [10%] Crea los datos de prueba para realizar pruebas funcionales.
5. [50%] Funcionalidades:
  - a. [5%] Compila correctamente desde la consola.
  - b. [13%] El requerimiento R1 añade un contenedor.
  - c. [13%] Entrega los resultados esperados de R2.
  - d. [14%] Entrega los resultados esperados del R3.
  - e. [5%] Genera el javadoc correctamente según lo explicado en clase.

### Entrega

Usted debe entregar un archivo en formato .zip cuyo nombre debe tener el formato PRIMERAPELLIDO\_PRIMERNOMBRE.zip en el cual se encuentre una carpeta con el nombre API\_PRACTICO2\_PRIMERAPELLIDO\_PRIMERNOMBRE, dentro de la cual se encuentren las carpetas de los respectivos paquetes definidos en el diagrama de clases y éstas a su vez contengan los archivos de las clases implementadas y la documentación.

En el método main usted puede implementar cualquiera de las dos siguientes alternativas:

1. Definir un escenario a través de la creación de los objetos con valores predeterminados e imprimir en consola el resultado respectivo del requerimiento 2 y 3.
2. Leer por consola los valores de cada uno de los objetos que se requieren crear para el funcionamiento apropiado del programa y finalmente imprimir en consola el resultado respectivo del requerimiento 2 y 3.