

# Matemáticas discretas

## Tarea 1

Prof: Fabián Riquelme Csori  
Ayudante: Diego Monsalves Cabello  
2018-II

### 1. Objetivos

- Solucionar el problema propuesto mediante la utilización de un algoritmo recursivo.
- Implementar un algoritmo con interfaz gráfica en Java.
- Desarrollar competencias de trabajo grupal.
- Aplicar los conceptos teóricos de algoritmos recursivos.

### 2. Descripción

- Se realizará en **grupos de 4 integrantes**, definidos a su elección.
- Se debe entregar una solución ejecutable, sin errores y ajustado a las exigencias del problema, no se debe entregar informe ni presentación.
- Se interrogará a todos los integrantes del grupo, para la defensa de su trabajo.

#### 2.1. Sistema de descarga portuaria

La actividad portuaria en un país es vital para su economía, y es por esta razón que Chile tiene en su larga soberanía marítima, más de 10 puertos de carga, entre ellos el puerto de Valparaíso. En este último, está presente la empresa “Terminal Pacífico Sur Valparaíso” o TPS, quienes en 2016 y 2017 presentaron una pérdida considerable de dólares por motivos de daños en maniobras de descargas. Las razones de ello fueron apilar contenedores sin considerar los pesos y dimensiones de cada uno.

Es por ello que la empresa TPS nos solicitó desarrollar un algoritmo que solucione dicho problema. Para ello nos entrega datos importantes de sus instalaciones y del procedimiento que se debe hacer.

### 2.1.1. Datos

- Los contenedores están apilados de forma correcta en la embarcación (desde el contenedor más pesado como base, hasta el más liviano en su punta).
- Para desembarcar, el Puerto de Valparaíso presenta 2 plataformas para dejar los contenedores, una auxiliar y una de descarga definitiva. Ambas plataformas deben seguir respetando las reglas de apilamiento del punto anterior.
- Para desembarcar los contenedores de cada barco solo se dispone de una grúa. Por tanto solo se puede tomar un contenedor por movimiento, esto es, solo se puede desplazar el contenedor que se encuentre arriba en cada plataforma.

### 2.1.2. Trabajo que deben realizar

- Implemente un algoritmo recursivo que permita trasladar todos los contenedores desde la plataforma de embarcación hasta la plataforma descarga definitiva, de modo que se mantenga la regla de apilamiento: no se puede dejar en ninguna de las tres plataformas un contenedor pesado sobre uno más liviano. Puede ayudarse de la plataforma auxiliar para cumplir con su cometido. La solución debe presentarse de manera gráfica, Utilizando el lenguaje de programación JAVA.
- Se debe tener como mínimo 3 clases: la clase `Contenedor.java`, la cual describirá el objeto contenedor (peso, origen, prioridad), la clase `Descargarecursiva.java`, en donde se debe realizar la acción recursiva para solucionar el problema, y la clase `Main.java`, que instanciará la clase `Descargarecursiva.java`. Esto no impide que para la solución de su problema pueda crear más clases.
- Su algoritmo debe permitir trasladar a lo menos 4 contenedores, pero la implementación debe ser escalable y permitir modificar esta cantidad.

## 2.2. Entrega

Deben entregar los archivos fuente `.java` y todo recurso utilizado por la aplicación, tales como ejemplo archivos, bibliotecas externas, etc. Se pondrá a disposición un “recurso tarea” en el Aula Virtual, donde deberán subir un archivo `.zip` con el formato:

Grupo NUM\_GRUPO-Tarea1.zip  
**Ejemplo: Grupo5-Tarea1.zip**

- **Descripción de Tarea:** 29 de octubre de 2018, en hora de clases.
- **Fecha de Entrega en Aula Virtual:** 26 de noviembre de 2018, 10:00 horas.
- **Fecha de Interrogación:** 26 de noviembre de 2018, en hora de clases.  
(en caso de no alcanzarse a interrogar a todos los grupos, se coordinará un horario adicional entre los grupos y el ayudante).

### 3. Sistema de evaluación

Esta tarea vale un 20 % de la nota final del curso. La nota de la Tarea 1 (NT1) estará dada por una nota grupal (NG1) resultante del algoritmo implementado, una nota individual (NI1) resultante de una pequeña interrogación referente al código, y un factor de participación (FP1) entregado por sus pares de forma anónima:

$$NT1 = (NG1 \cdot 0,8 + NI1 \cdot 0,2) \cdot FP1$$

#### 3.1. Nota grupal (NG1)

Criterio	Puntos
<b>Representación gráfica.</b> Se visualiza claramente el estado inicial, desarrollo paso a paso (posiciones intermedias de contenedores) y solución del problema planteado (estado final).	20 pts
<b>Calidad del código.</b> El código está implementado exclusivamente en Java; es claro, está bien documentado y utiliza al menos las 3 clases solicitadas.	20 pts
<b>Correctitud.</b> El algoritmo funciona correctamente para 4 contenedores.	40 pts
<b>Escalabilidad.</b> El algoritmo es escalable a más contenedores.	20 pts

#### 3.2. Nota individual (NI1)

Criterio	Puntos
<b>Comprensión de código.</b> A partir de preguntas realizadas por el Ayudante, el alumno demuestra comprender el código implementado.	60 pts
<b>Comprensión teórica.</b> A partir de preguntas realizadas por el Ayudante, el alumno demuestra comprender el problema y la solución planteada por el equipo.	40 pts

#### 3.3. Factor de participación (FP1)

Nivel de trabajo	Factor
<b>Excelente.</b> Mi compañero/a fue indispensable para la realización del trabajo. Sin él/ella no se podría haber realizado.	1.1
<b>Bueno.</b> Mi compañero/a se desempeñó de manera satisfactoria y responsable en el equipo.	1.0
<b>Insuficiente.</b> Mi compañero/a no participó lo suficientemente en el trabajo. Su desempeño no fue el ideal esperado.	0.9
<b>Pésimo.</b> Mi compañero/a no fue un aporte al equipo. El equipo habría funcionado mejor sin él/ella.	0.8