

Proyecto de Curso

# Planificación automática de horas de descargue para un puerto naval comercial

Programación por Restricciones

Escuela de Ingeniería de Sistemas y Computación

Facultad de Ingeniería



Profs. Juan Francisco Díaz Frias, Robinson Duque

Abril del 2021

## 1. Introducción

El presente proyecto tiene por objeto enfrentar a los estudiantes del curso:

- Al análisis de un problema abordado desde la óptica del paradigma de Restricciones y la elaboración de un Modelo de Restricciones producto de dicho análisis,
- A la escogencia de una o varias heurísticas que ayuden a la reducción o poda de los dominios de las variables involucradas en el problema e implementación de las heurísticas elegidas.
- A la escogencia de una estructura de datos adecuada al Modelo de Restricciones para la implementación en **Minizinc** de la solución al problema.

## 2. El problema de la Planeación automática de horas de descargue para un puerto naval comercial

Una actividad natural y frecuente en todo puerto, consiste en la descarga de mercancía transportada por los barcos cargueros. Debido a la gran cantidad de barcos que atiende un determinado puerto comercial, al tiempo necesitado para descargar cada barco y al número limitado de muelles de cada puerto, esta actividad debe ser programada con suficiente tiempo de anticipación para que no se presenten incrementos en los costos por demoras en la atención y descarga de los barcos ni inconvenientes graves como la pérdida de mercancía por descargas a destiempo, o el encallamiento de un buque por bajas de marea o el deterioro de la carga por causa del clima, entre otros. Este es entonces un problema de planificación (scheduling) cuya solución sería de gran utilidad en la práctica.

Para soportar ese trabajo de planeación se propone desarrollar una aplicación capaz de planear automáticamente

las horas de llegada de los barcos diariamente, a partir de las características de la carga, y de las restricciones de tiempos, clima y mareas impuestas por cada barco. La aplicación debe calcular soluciones en un tiempo razonable y debe ser capaz de reorganizar esta planeación en cualquier momento, puesto que la tasa de imprevistos es muy alta.

### 3. Definición Formal del problema

Para delimitar claramente el alcance del proyecto, en esta sección se describen la entrada, la salida y las restricciones que la aplicación debe tener en cuenta.

1. La información de entrada al sistema contiene los siguientes datos:
  - Número de muelles de descarga del puerto.
  - Número de barcos que planean llegar el día que se debe planificar.
  - Cada barco tiene la siguiente información:
    - Código del barco.
    - Descripción de la carga:
      - Tiempo máximo de espera: Es el tiempo que puede pasar entre el momento de llegada del barco y el momento en que comienza la descarga.
      - Lista de estados del tiempo posibles, por ejemplo [seco húmedo] o [seco húmedo lluvia]. Esta lista especifica las condiciones del clima que deben garantizarse durante la estadía del barco en el puerto, para que no se deteriore la carga.
      - Demora de descarga: Es el tiempo que se toma descargar el barco.
    - Lista de estados de marea posibles, por ejemplo [media alta] o [alta]. Esta lista especifica las condiciones de la marea que deben garantizarse durante la estadía del barco en el puerto, para que el barco no corra el riesgo de encallar en el puerto.
  - Cada muelle tiene la siguiente información:
    - Código del muelle.
  - El puerto tiene la siguiente información:
    - Descripción del estado del tiempo en el puerto por cada cinco minutos (seco, húmedo o lluvia), del día que se debe planificar.
    - Descripción del estado de la marea por cada cinco minutos (baja, media, alta), del día que se debe planificar.
2. La forma en que es pasada la entrada al sistema podría ser definida por el(los) desarrollador(es). Hay que tener en cuenta que esta aplicación puede ser un módulo de un proyecto más grande; entonces la entrada es generada por otro módulo, no directamente por el usuario final y por lo tanto esta entrada puede ser un archivo, una base de datos, o lo que sea mas fácil. Sin embargo, para tratar de estandarizar los datos de entrada, proveemos algunos ejemplos de cómo formatear los datos para lograr hacer competencias entre los grupos de trabajo.
3. La unidad de tiempo debe ser 5 minutos.
4. La salida del sistema debe contener la información: código del barco, código del muelle y hora de llegada, hora de inicio de la descarga, para cada barco.
5. Las restricciones que se deben satisfacer por defecto son:
  - Todos los barcos deben ser atendidos. Si no es posible, no hay solución.
  - Cada muelle solo puede atender un barco a la vez.
  - Se debe respetar el tiempo máximo de espera definido por la carga del barco.
  - Durante el tiempo de espera y descarga de un barco el estado del tiempo debe estar dentro de la lista de estados posibles de tiempo definidos por el tipo de carga.
  - Durante el tiempo de espera y descarga de un barco el estado de la marea debe estar dentro de la lista de posibles estados de marea definida por el barco.
6. Las restricciones adicionales (impuestas por el usuario si lo desea) son:
  - Preasignación de la hora de llegada de un barco (puede ser débil o fuerte; débil significa que es preferible, pero no necesario, y fuerte que es necesario)
  - Definición de horas hábiles para un muelle.

## 4. Técnicas utilizadas

Todo grupo de trabajo es libre de diseñar el modelo del problema de la manera que considere más adecuada e implantar dicho modelo con el algoritmo que desee, siempre y cuando se ajuste al estilo de programación del paradigma de restricciones y específicamente a la forma de programación en **Minizinc**.

## 5. Implementación

Ud. deberá implementar una interfaz de entrada/salida, ojalá gráfica. La entrada se leerá de archivos de texto; la interfaz de salida debe permitir ver claramente que la solución sí satisface el pedido de entrada.

Todos los programas deben correr bajo un mismo ambiente; se debe entregar el programa **ejecutable** y el código fuente. **La fecha de entrega es el 18 de Mayo de 2021 a las 23:59. La sustentación será el 25 de mayo.**

Debe entregar un informe en formato pdf, los archivos fuente, un archivo Readme.txt que describa todos los archivos entregados y las instrucciones para ejecutar la aplicación. Todo lo anterior en un solo archivo empaquetado cuyo nombre contiene los apellidos de los autores y cuya extensión corresponde al modo de compresión. Por ejemplo Diaz.zip o Diaz.rar, o Diaz.tgz o ...

## 6. Conclusiones

Esta es una de las partes más interesantes del trabajo (pero no por ello la que más vale). En ella se espera que usted analice los resultados obtenidos y **justifique** claramente sus afirmaciones.

## 7. Sustentación y calificación

El trabajo debe ser sustentado por los autores en día y hora por definir. La calificación del proyecto se hará teniendo en cuenta los siguientes criterios:

1. Informe (Modelo, Pruebas y Conclusiones)
2. Implementación.
3. Desempeño personal en la sustentación, lo cual incluye la capacidad de cada uno de navegar en el código y realizar cambios rápidamente en él.
4. **Reta a tus compañeros con una instancia:** diseña una entrada del problema y reta un grupo a que la solucione en menor tiempo que tu modelo. Si el grupo que es retado no logra resolver la instancia en menos tiempo, obtendrás 0.5pts adicionales en la nota del proyecto, de lo contrario obtendrás 0.2pts.

En todos los casos la sustentación será pilar fundamental de la nota asignada. Cada persona de cada grupo, después de la sustentación, tendrá asignado un número real (el factor de multiplicación) entre 0 y 1, correspondiente al grado de calidad de su sustentación. Su nota definitiva será la nota del proyecto, multiplicada por ese valor. Si su asignación es 1, su nota será la del proyecto. Pero si su asignación es 0.9, su nota será 0.9 por la nota del proyecto. La no asistencia a la sustentación tendrá como resultado una asignación de un factor de 0.

La idea es que lo que no sea debidamente sustentado no vale así funcione muy bien!!!

Éxitos!!!