



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
(ICAI)

Máster en Industria Conectada (MIC)

**Desarrollo de un Gemelo Digital para la
Optimización de una Refinería**

Autora

Blanca Ferrer Sánchez del Villar

Dirigido por

Pedro Fernández Gaspar

Madrid

Julio 2024

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título *Desarrollo de un Gemelo Digital para la Optimización de una Refinería* en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2023/2024 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.



Fdo.: Blanca Ferrer Sánchez del Villar

Fecha: 12 / 07 / 2024

Autorizada la entrega del proyecto
EL DIRECTOR DEL PROYECTO



Fdo.: Pedro Fernández Gaspar

Fecha: 12 / 07 / 2024

Índice

1. Introducción	1
2. Estado de la técnica	1
3. Objetivos	1
4. Conceptualización	2
5. Etapas del Proyecto	2
6. Lógica del algoritmo de ventas	3
6.1. Gasolinas	3
6.2. Destilados	5
6.3. GNL	6
6.3.1. Propano	7
6.3.2. C3 Mix	7
6.3.3. nC4	9
7. Conclusiones	9

Índice de figuras

1.	GANTT del proyecto	3
2.	Diagrama funcionamiento Gasolinas	5
3.	Diagrama funcionamiento Diesel	6
4.	Diagrama funcionamiento Jet A	6
5.	Diagrama funcionamiento Propano	7
6.	Diagrama funcionamiento C3 Mix	8
7.	Diagrama funcionamiento nC4	9

Índice de cuadros

1. Introducción

En la era digital, la implementación de tecnologías avanzadas es crucial para que las empresas sigan siendo competitivas y optimicen sus procesos. Una de estas tecnologías es el "gemelo digital", una copia virtual precisa de un sistema físico. Esta innovación está revolucionando diversas industrias y ofrece notables mejoras en eficiencia, seguridad y sostenibilidad en el sector de las refinerías. Este Trabajo de Fin de Máster (TFM), realizado en colaboración con The Boston Consulting Group (BCG), crea un gemelo digital para una refinería, analizando sus beneficios, desafíos y el impacto potencial en la industria.

2. Estado de la técnica

El concepto de gemelo digital, que consiste en una copia virtual exacta de un sistema físico, es una tecnología emergente en la digitalización industrial. Facilita la supervisión y simulación en tiempo real de los procesos y equipos de una refinería, permitiendo optimizar operaciones, predecir fallos y aumentar la seguridad y la eficiencia.

La aplicación de gemelos digitales en la industria del petróleo y gas es motivada por la necesidad de disminuir costos operativos y aumentar la eficiencia energética. Además, ayuda a cumplir con estrictas regulaciones ambientales mediante una gestión exacta de emisiones y desechos, y a manejar la complejidad de las operaciones de refinería a través de sistemas de información avanzados.

3. Objetivos

El propósito central de este Trabajo de Fin de Máster es desarrollar un gemelo digital para una refinería. Los objetivos específicos son los siguientes:

1. **Desarrollar una copia virtual exacta de la refinería**
2. **Optimizar los procesos operativos**
3. **Mejorar la seguridad y la gestión de riesgos**
4. **Incrementar la sostenibilidad y el cumplimiento ambiental**
5. **Facilitar la capacitación y el desarrollo del personal**

6. **Evaluar la viabilidad y el impacto del gemelo digital**
7. **Crear un modelo que pueda ser replicado**

4. Conceptualización

La implementación del gemelo digital consta de tres herramientas:

1. **Planificación estratégica mensual:** Desarrolla un plan mensual que determine las cantidades de crudo a adquirir y los productos a fabricar y comercializar.
2. **Planificación diaria detallada:** Elabora un plan diario, basado en la planificación mensual, para gestionar las cantidades necesarias de recepción y despacho de productos.
3. **Programación horaria específica:** Diseña una programación detallada para las operaciones diarias, desglosada por horas.

Cada herramienta toma información de la herramienta de nivel superior y proporciona retroalimentación sobre la factibilidad de las propuestas, estableciendo un proceso iterativo que ajusta la planificación hasta alcanzar la solución óptima.

5. Etapas del Proyecto

El desarrollo de esta herramienta es esencial para optimizar las operaciones y generar visualizaciones efectivas. El proyecto se divide en las siguientes fases:

1. **Recepción de Crudo:** recepción y almacenamiento del crudo en tanques específicos, utilizando tuberías, camiones y barcos.
2. **Procesamiento de Crudos:** separación y tratamiento de crudos para obtener componentes valiosos como nafta, keroseno y coque, seguidos de su almacenamiento.
3. **Producción de Componentes para Gasolinas:** fabricación y control de componentes como benceno, tolueno y reformado, utilizados en la mezcla de gasolinas.
4. **Producción de Componentes para Diésel:** procesos dedicados a la producción de componentes de diésel, asegurando eficiencia y calidad.

5. **Producción de Componentes para Gasoil:** procedimientos para obtener componentes de gasoil, cumpliendo con las especificaciones necesarias.
6. **Mezcla de Productos:** combinación de componentes para producir productos finales que cumplan con las especificaciones de calidad.
7. **Distribución de Productos:** eficiente distribución de productos a través de tuberías, camiones y barcos, maximizando los beneficios.

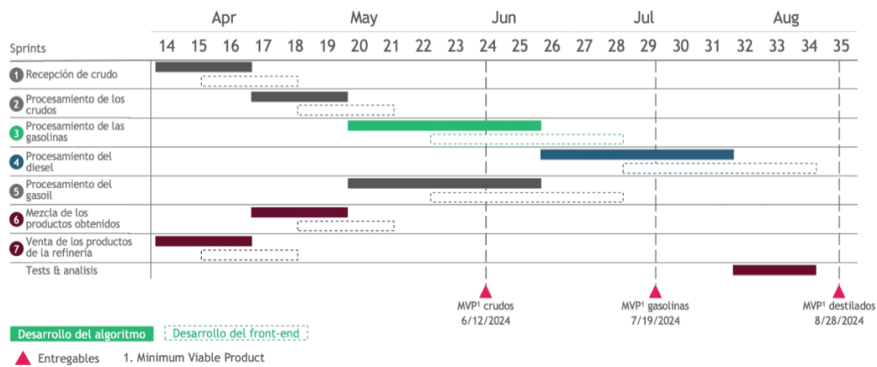


Figura 1: GANTT del proyecto

6. Lógica del algoritmo de ventas

6.1. Gasolinas

Primero, abordaremos el tema de las gasolinas. Nuestra refinería produce principalmente tres tipos: Gasolina A, Gasolina M y Gasolina D.

Además de estas tres variedades, producimos otras gasolinas bajo pedidos específicos. Estas no se modelarán debido a que su producción es ocasional y solo se lleva a cabo cuando se recibe una solicitud específica.

Las gasolinas producidas por la refinería se almacenan en tanques llamados tanques finales, desde donde se realiza la venta al cliente final. Es importante señalar que los tanques finales de gasolina no pueden despachar a varios destinos al mismo tiempo; deben hacerlo de manera secuencial o bien despachar desde dos

tanques finales diferentes.

Las gasolinas se distribuyen según el siguiente orden de prioridad:

1. Distribución de Gasolina por Camión

La distribución de gasolina mediante camiones se lleva a cabo desde los tanques finales hacia unos tanques intermedios o tanques de despacho, que reciben el producto de los tanques finales y desde ellos se realiza la venta a los camiones. Solo se comercializan dos tipos de gasolina por camión: Gasolina A y Gasolina D. La lógica específica del funcionamiento de las ventas por camión está detallada en el anexo C.

2. Distribución de Gasolina por Barcos

Después de completar las distribuciones de gasolina por camión, se da prioridad a las exportaciones comprometidas de gasolina mediante barcos o barcazas. Se considerará barco si tiene un volumen superior a 50 MBBL, y barcaza si es menor o igual. Esto afectará la ocupación de los puertos y el tiempo de amarre. La lógica específica del funcionamiento de las ventas por barco está detallada en el anexo C.

3. Distribución de Gasolina por Tubería

La siguiente prioridad es la distribución a través de la tubería de gasolinas de acuerdo con su calendario de salidas. Es crucial evitar conflictos en el uso de las tuberías para asegurar un flujo constante y eficiente, ya que todas las gasolinas utilizan la misma tubería y no se pueden enviar simultáneamente dos tipos diferentes de gasolinas por la misma tubería. La lógica específica del funcionamiento de las ventas por tubería está detallada en el anexo C.

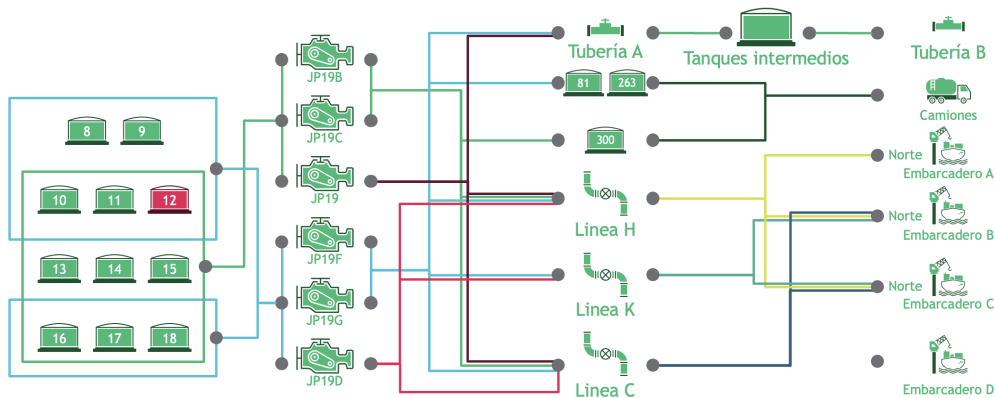


Figura 2: Diagrama funcionamiento Gasolinas

6.2. Destilados

La refinería comercializa cinco tipos de destilados: Diesel, Combustible de Turbina (Jet A), Kero Ultra Bajo en Azufre, Jet A1 y Def Stan.

Los últimos tres tipos de destilados se producen de manera ocasional. No se modelarán debido a su producción esporádica y se fabrican solo bajo pedido específico. Sin embargo, la lógica se diseñará para poder aceptar solicitudes de estos productos.

Los destilados se despachan según el siguiente orden de prioridad:

1. Distribución de destilados por barcos

En primer lugar, se priorizan las exportaciones comprometidas de destilados mediante barcos o barcazas. Se utiliza el mismo criterio de volumen para diferenciar entre barco y barcaza. La lógica específica del funcionamiento de las ventas por barco está detallada en el anexo C.

2. Distribución de destilados por tubería

La siguiente prioridad es la distribución a través de la tubería de destilados de acuerdo con su calendario de salidas. Hay una tubería dedicada para destilados, y solo un tipo de destilado puede ser despachado a la vez. La lógica específica del funcionamiento de las ventas por tubería está detallada en el anexo C.

3. Distribución de destilados por camión

El Diesel también se comercializa por camión. Los otros destilados no se venden de esta manera. Por lo tanto, solo habrá ventas de Diesel por camión. La lógica específica del funcionamiento de las ventas por camión está detallada en el anexo C.

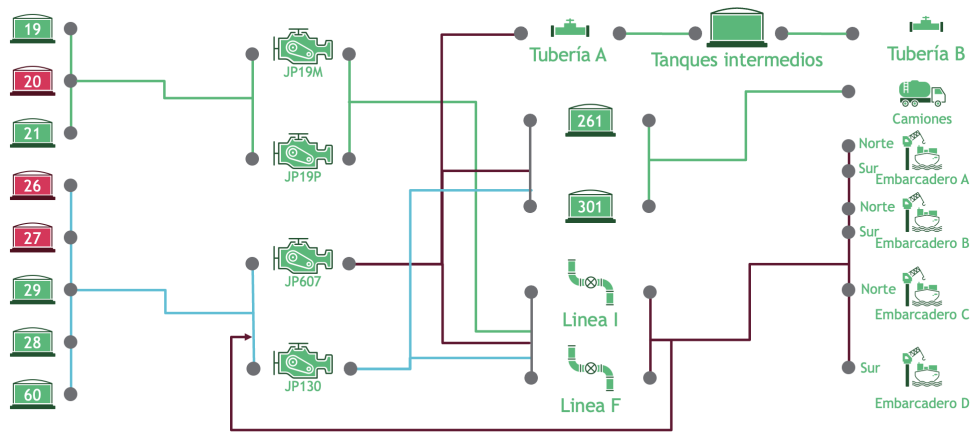


Figura 3: Diagrama funcionamiento Diesel

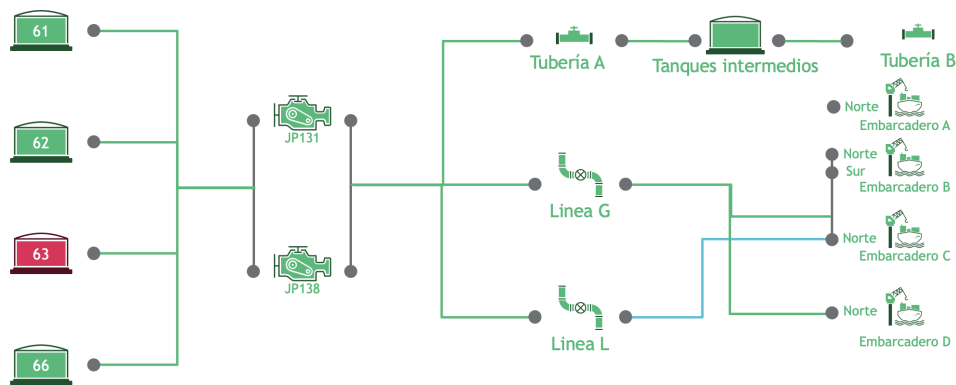


Figura 4: Diagrama funcionamiento Jet A

6.3. GNL

El Gas Natural Licuado (GNL) cuenta con una infraestructura propia, lo que proporciona una mayor flexibilidad en su planificación y distribución. A pesar de

esto, es fundamental optimizar los tiempos de entrega y gestionar eficientemente los recursos disponibles para alcanzar los objetivos diarios.

Se despachan 3 tipos de GNL:

6.3.1. Propano

El propano es recibido del suministro de propano y se almacena en los tanques finales designados para este tipo de gas. Estos tanques finales de propano no funcionan en modo flotante.

El propano se transfiere al tanque de entrega F31, que opera como un tanque flotante (para recepción y envío). Desde allí, el propano se transporta a través de la tubería designada como tubería C, con una capacidad máxima de 12 MBPD. Es esencial que el volumen en el tanque F31 se mantenga constante, ya que el flujo entrante debe ser igual al flujo saliente.

Una parte del propano, en lugar de ser despachado al tanque de entrega F31, puede ser vaporizado (quemado) si no cumple con las especificaciones o para ajustar el balance del combustible. Al calcular el flujo desde los tanques finales de propano, se debe tener en cuenta que este debe ser la suma del volumen que sale por la tubería C y el que se vaporizará (quemará) en el futuro. El volumen vaporizado se determinará como la diferencia entre la producción y el despacho.

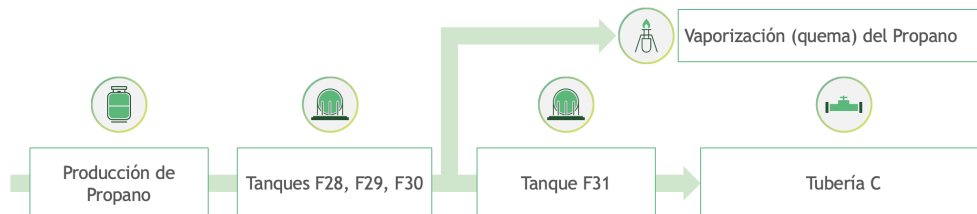


Figura 5: Diagrama funcionamiento Propano

6.3.2. C3 Mix

La producción de C3 Mix se obtiene del suministro de C3 Mix y se almacena en los tanques finales correspondientes, que operan en modo flotante.

Antes de llegar a estos tanques, se puede añadir propano, permitiendo una mezcla de hasta un 35 % de propano en el C3 Mix, dependiendo de las consideraciones económicas.

Desde los tanques finales de C3 Mix, el producto puede ser enviado a la tubería D y al Pozo 3. La tubería D es compartida con las ventas de nC4, teniendo el nC4 prioridad. Los tanques finales de C3 Mix funcionan de manera similar al tanque de entrega de propano F31, manteniendo un volumen constante ya que el flujo entrante debe ser igual al flujo saliente.

La tubería D tiene un límite de flujo de 30 MBPD, mientras que el Pozo 3 está conectado a través de la Línea A, con un límite de flujo de 19 MBPD. El Pozo 3 opera en modo flotante. El flujo del Pozo 3 se dirigirá a la tubería D cuando la recepción del producto de la producción sea insuficiente para satisfacer la demanda. Es posible enviar producto simultáneamente desde los tanques finales de C3 Mix y el Pozo 3.

Antes de despachar a la tubería D y al Pozo 3, se puede recibir un flujo de C3 Mix comprado a una empresa externa.

Para determinar la tasa a la que se debe introducir este volumen, se deben considerar los límites de flujo de la tubería D y del Pozo 3. Dependiendo del destino y la tasa de flujo desde los tanques finales de C3 Mix, se establecerá la tasa de flujo para mezclar el C3 Mix comprado con el C3 Mix producido por la refinería.

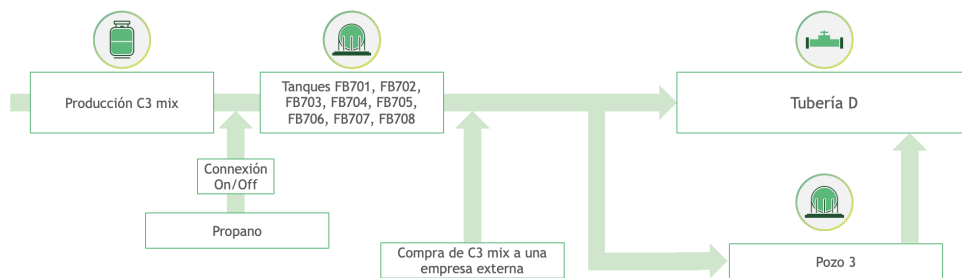


Figura 6: Diagrama funcionamiento C3 Mix

6.3.3. nC4

La producción de nC4 proviene del suministro de nC4 y se almacena en los tanques finales de nC4, los cuales operan en modo flotante. Desde estos tanques finales, la refinería destina una pequeña parte de nC4 para la mezcla de gasolina (aproximadamente 2 MBPD de una producción total de 17 MBPD de nC4).

El resto del nC4 se envía a los Pozos 4 y 12 o a las barcazas. Es posible despachar simultáneamente a los tres destinos (mezcla, barcazas y pozos). Desde los Pozos 4 y 12, el flujo se canaliza hacia la tubería D. Ambos pozos operan en modo flotante.

Para el nC4, solo es necesario cumplir con los compromisos existentes, ya que puede almacenarse y utilizarse en la producción de gasolina durante el invierno.

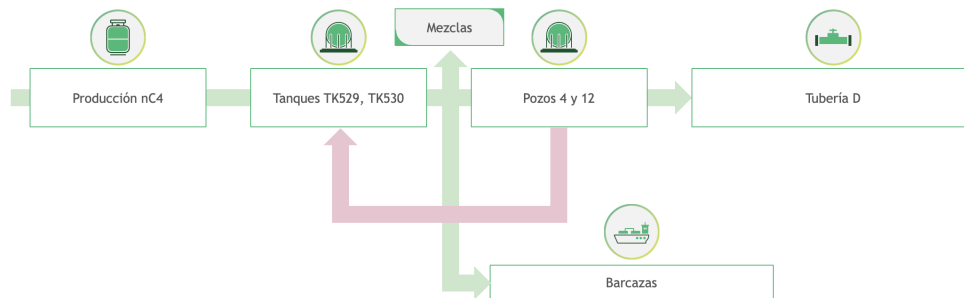


Figura 7: Diagrama funcionamiento nC4

7. Conclusiones

El implementar un gemelo digital en la refinería ha demostrado ser una solución tecnológica avanzada con numerosos beneficios. Este enfoque ha mejorado considerablemente la eficiencia operativa y la seguridad, permitiendo una gestión más precisa de los equipos y procesos. La capacidad de simular diferentes escenarios sin interrumpir las operaciones reales ha optimizado el rendimiento y facilitado la toma de decisiones informadas basadas en datos, adaptándose rápidamente a los cambios en el mercado y la producción.

En resumen, este avance tecnológico en la gestión de operaciones industriales destaca la importancia de la innovación para mantener la competitividad y la sostenibilidad a largo plazo.

Bibliografía

Sueldos BCG Glassdor : <https://www.glassdoor.es/Sueldo/Boston-Consulting-Group-Sueldos-E3879.htm>.

Consulta : Julio 08, 2024

Objetivos de Desarrollo Sostenible, ONU : <https://www.un.org/sustainabledevelopment/>.

Consulta : Julio 08, 2024

Vijini Mallawaarachichi: Introduction to Genetic Algorithms - Including Example Code <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>.

Consulta : Julio 08, 2022

L.F.L.Moro, M.Joly y J.M.Pinto: "Planning and scheduling for petroleum refineries using mathematical programming"

L.F.L.Moro, M.Joly y J.M.Pinto: "Planning and scheduling for petroleum refineries using mathematical programming".

Consulta: Julio 08, 2024

Apéndice A

GANTT del proyecto

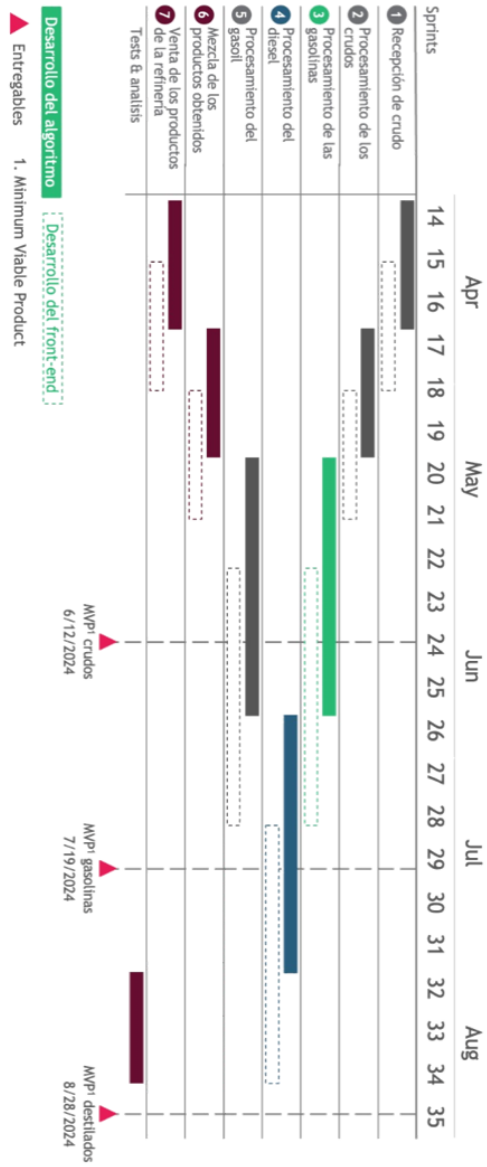


Figura A.1: GANTT del proyecto

Apéndice B

Lógica fundamental del algoritmo

B.1. Objetivo del Algoritmo

El objetivo del algoritmo de programación es simular un conjunto físico de componentes, interactuando entre sí a través de acciones desencadenadas por lógica interna y datos de entrada, con el fin de proporcionar escenarios factibles y optimizados que se extienden a lo largo de un período de tiempo.

Para ello, se simularán diferentes caminos en paralelo, divergentes en nuevas decisiones hacia nuevos caminos, priorizando qué caminos tomar y terminando aquellos caminos que enfrentan una situación imposible y por lo tanto se consideran inviables.

Esta sección presenta algunos de los componentes y comportamientos principales del algoritmo de programación. Para ilustrarlos, se usará un ejemplo de simulación de la cocina de un restaurante, donde el objetivo es simular la cocción y preparación de platos en una cocina, gestionando los inventarios de ingredientes, y tomando como entrada datos dinámicos sobre la entrada de inventario y los pedidos de salida.

B.2. Descripción de los objetos físicos

Esta sección describe las clases utilizadas para modelar objetos físicos, como Elementos, Sistema y Estados.

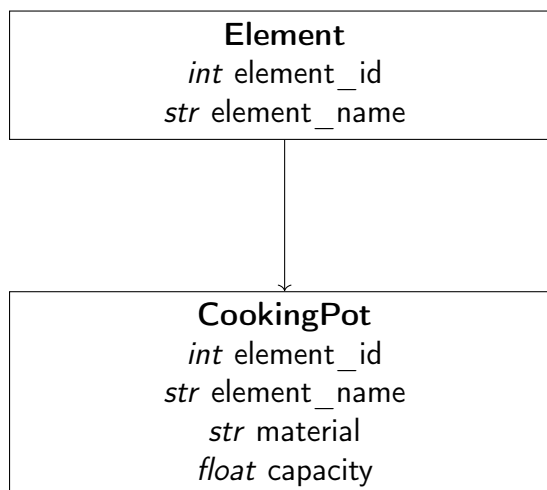
B.2.1. Las propiedades físicas estáticas: Elementos y Sistema

Las propiedades estáticas no cambian durante la simulación, por lo tanto, se almacenan en objetos separados que no se copian junto con el resto de las propiedades cambiantes y dinámicas, para ahorrar espacio en memoria.

B.2.2. Elemento

La información estática de un objeto físico se representa en una clase Elemento. Una clase Elemento no cambia durante la simulación, pero proporciona la información necesaria sobre las propiedades del objeto.

En el ejemplo de la cocina, una Olla será una subclase de Elemento y contendrá propiedades estáticas sobre una olla particular, como su nombre, material y capacidad.



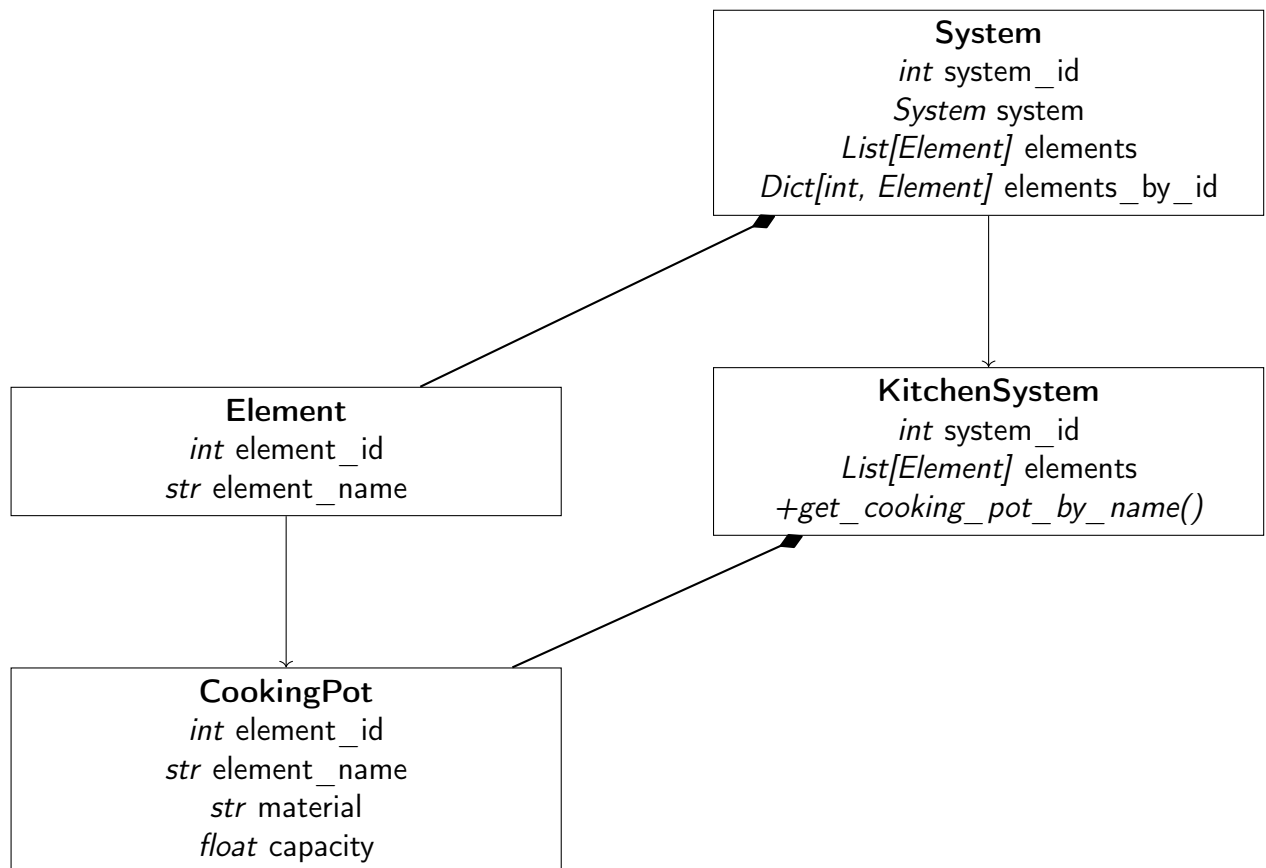
B.2.3. Sistema

Todos los elementos de una simulación se agrupan en un objeto de la clase Sistema.

Si la simulación solo consiste en una cocina, una clase concreta de Sistema podría ser KitchenSystem, donde se incluyen todos los elementos simulados. Sin embargo, si la simulación incluyera todo un restaurante o una red de cocinas, de

manera que la cocina se convierta en un subgrupo de elementos dentro de la simulación, entonces la cocina no sería un Sistema, sino un Elemento, dentro del cual se incluirían los otros elementos de la cocina.

Asumiendo que la simulación solo incluye una cocina, las clases podrían ser:



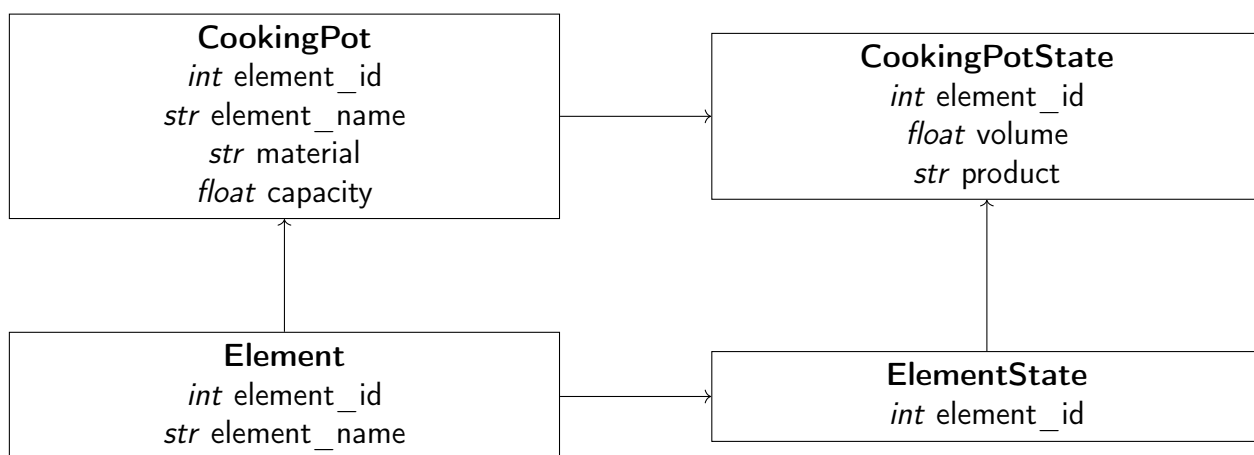
B.2.4. Las propiedades físicas dinámicas: Estados

Las propiedades físicas dinámicas representan las condiciones y propiedades cambiantes en los elementos operativos. A lo largo de la simulación, deben copiarse de iteración en iteración, para mantener los datos sobre cómo evoluciona el Sistema.

Estado del Elemento

Las propiedades cambiantes de un Elemento se almacenan en una clase separada llamada Estado del Elemento. Cada Estado del Elemento apunta a su Elemento correspondiente a través de su atributo `element`, para acceder a la información estática necesaria sobre un Estado del Elemento.

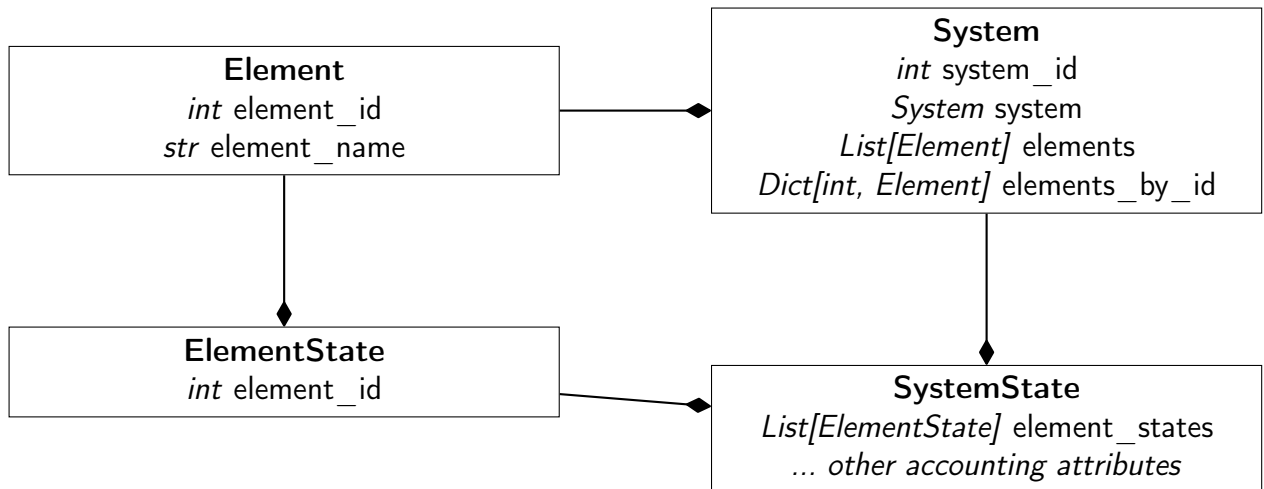
Por ejemplo, un Estado de Olla será una subclase de Estado del Elemento, contendrá propiedades dinámicas sobre una olla particular, como el volumen y el producto que contiene, y también apuntará a un objeto Olla para verificar propiedades estáticas como su velocidad.



Estado del Sistema

Los Estados de los Elementos se agrupan en objetos de la clase Estado del Sistema, que representan el estado cambiante de un grupo de elementos. El Estado del Sistema puede incluir otros atributos, como tareas pendientes, para realizar un seguimiento de las acciones que deben ejecutarse.

Para acceder fácilmente a los datos estáticos de los elementos en el Estado del Sistema, un atributo `system` apuntará al Sistema correspondiente por referencia. Este atributo no se copiará, para ahorrar memoria.



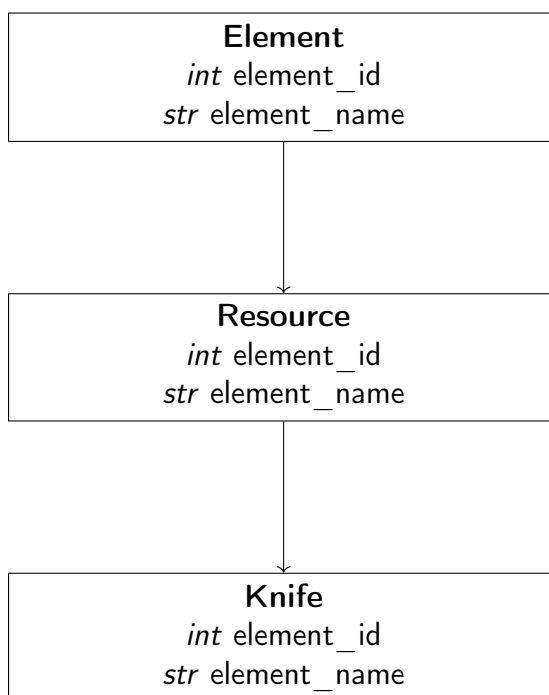
B.3. Otros componentes relevantes

Aquí hay algunos otros componentes auxiliares, estrechamente relacionados con los descritos anteriormente en esta sección, que serán relevantes al ejecutar el algoritmo:

B.3.1. Recurso

La clase Recurso es una subclase de Elemento y representa un elemento para el cual no se requiere un Estado del Elemento, ya que el elemento no tiene propiedades cambiantes. Por ejemplo, si necesitamos rastrear si el Recurso está disponible o no para su uso, esta información se puede almacenar en el objeto Estado del Sistema, en un atributo Gestor de Bloqueo, por ejemplo.

En una cocina, un ejemplo de Recurso podría ser un cuchillo, asumiendo que siempre permanece afilado.



B.3.2. Solución

No hay una clase Solución en el algoritmo. Sin embargo, el término solución se usa ampliamente en el código, por lo que es importante entender a qué se refiere.

Una solución apunta a una de las últimas instancias de Estado del Sistema actualmente ejecutadas por la simulación. Cuando un Estado del Sistema evoluciona a 2 o más Estados del Sistema después de ejecutar una acción, se tratarán como soluciones que deben priorizarse y seleccionarse.

Una vez que la simulación termina, los Estados del Sistema resultantes también se considerarán soluciones.

B.4. Los impulsores del cambio: Escenarios, Tareas y Eventos

La simulación evoluciona procesando diferentes eventos y tareas, que generan escenarios consistentes en secuencias de acciones que deben ejecutarse para cambiar el Estado del Sistema y completar las tareas asociadas a los eventos.

En cada iteración se procesa una tarea, lo que da lugar a uno o más escenarios. Por lo tanto, se crea una cadena de escenarios, cada uno apuntando al anterior.

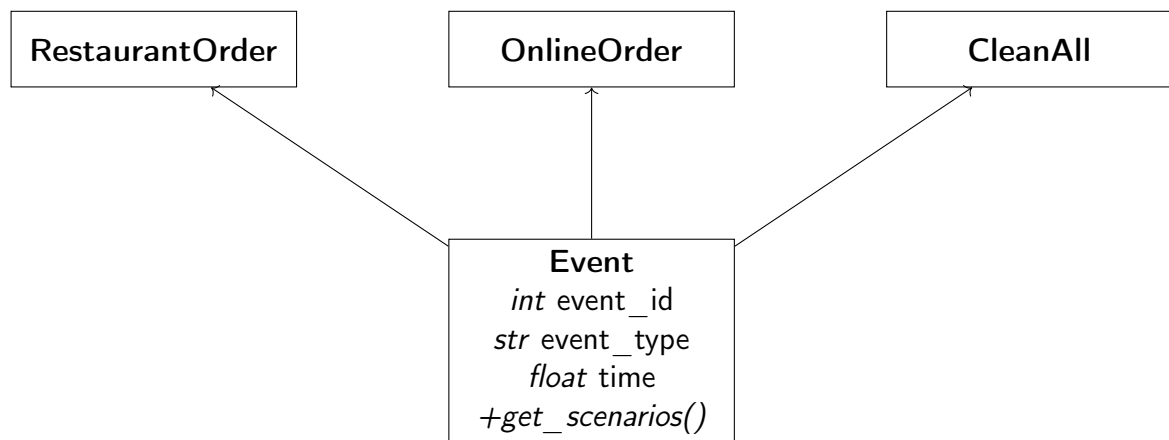
B.4.1. Qué hacer: Eventos y Tareas

Evento

Un Evento representa cualquier evento que ocurre durante el período de simulación. Ocurren en un momento particular y pueden tener un intervalo de tiempo asociado.

Los eventos se clasifican en diferentes tipos, dependiendo del tipo de evento que ocurra. En nuestro ejemplo de la cocina, algunos tipos de eventos pueden ser PedidoRestaurante, PedidoOnline o LimpiarTodo.

Muchos eventos se generan al comienzo de la simulación utilizando datos de entrada, pero también se pueden generar nuevos eventos durante la simulación.



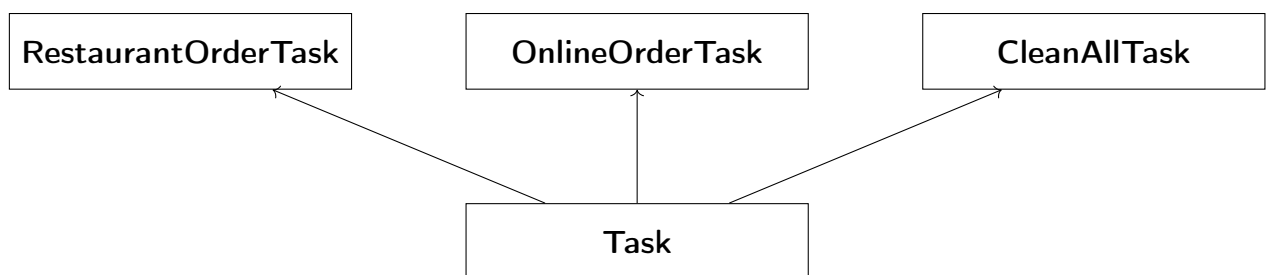
Tarea

Una Tarea es una unidad de trabajo que debe procesarse para cambiar el Estado del Sistema. Las tareas están asociadas a eventos específicos.

Se ejecutan secuencialmente en cada solución, actuando como un desencadenante para la generación de escenarios. La secuencia de tareas se determina mediante

algún mecanismo de priorización basado en los atributos de las tareas, como el tipo de tarea o su tiempo.

Ejemplos de tareas de cocina podrían ser `TareaPedidoRestaurante`, `TareaPedidoOnline` o `TareaLimpiarTodo`. Por ejemplo, una `TareaPedidoRestaurante` hará un seguimiento del tiempo en que se recibe el pedido y cada uno de los platos que se deben preparar para completar el evento del pedido. Una `TareaLimpiarTodo`, por otro lado, solo puede especificar el tiempo al final del día en el que se debe limpiar la cocina antes de cerrar, para asegurar que esto suceda.



B.5. Aplicación de esta lógica fundamental al caso de la refinería

Esta sección proporciona explicaciones de los elementos particulares que participan en la modelización del sistema de refinería.

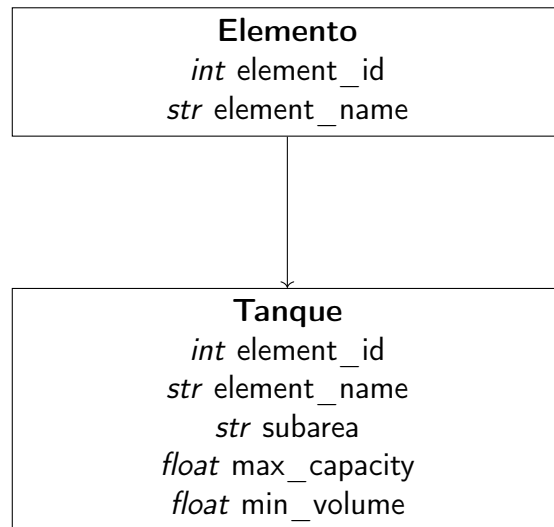
B.5.1. Elementos

Hay algunos elementos que son esenciales para el sistema de refinería. Estos elementos son los que se utilizan para modelar los objetos físicos que forman parte del sistema de refinería.

Tanques

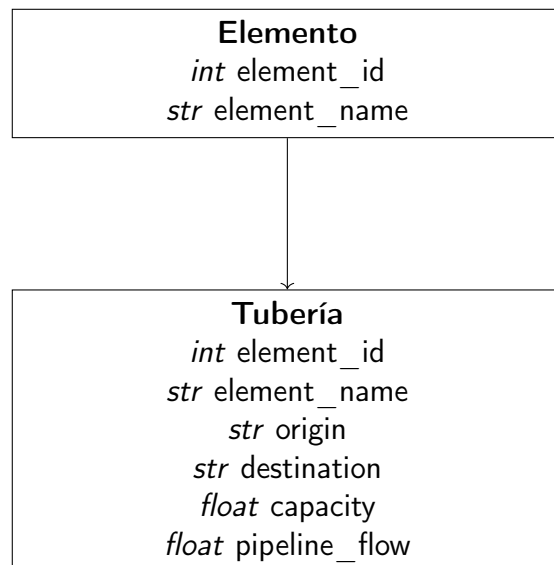
Los tanques son los activos más básicos de una refinería. Son contenedores en los que se almacena un determinado producto. Un Tanque será una subclase de `Elemento` y contendrá propiedades estáticas sobre un tanque particular, como su

nombre, área o capacidad máxima.



Tuberías

Una Tubería será una subclase de Elemento y contendrá propiedades estáticas sobre una tubería particular, como su nombre, capacidad, origen, destino o flujo de la tubería.



ADVERTENCIA

Una tubería puede o no tener un objeto PipelineState asociado, dependiendo

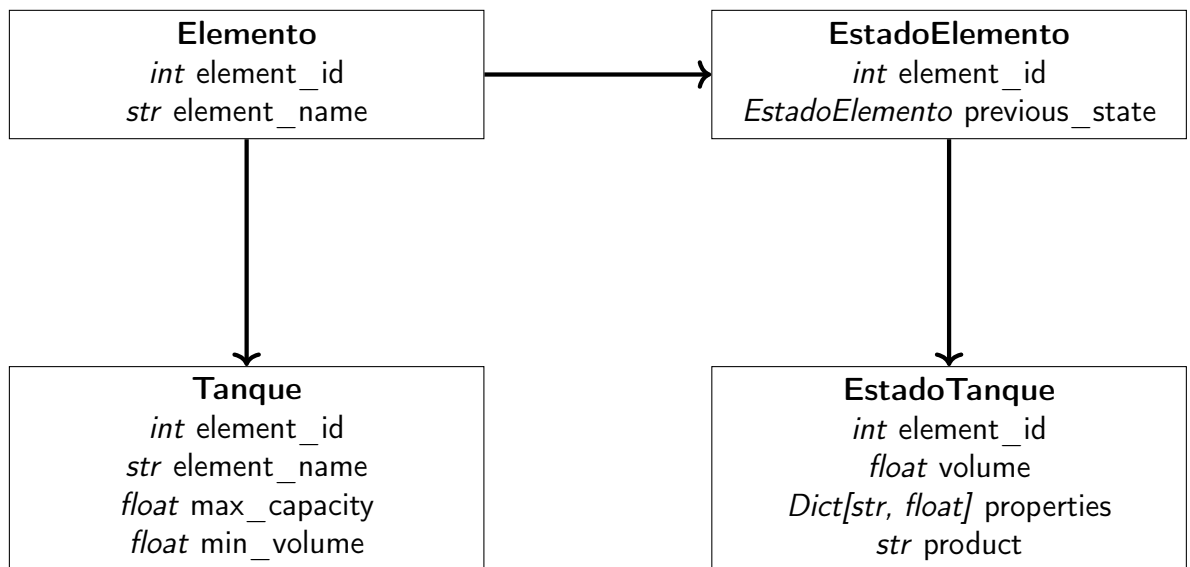
de si se modelan o no las propiedades dinámicas de la tubería (como su composición). En caso de que no se modelen las propiedades dinámicas, la tubería será en realidad un objeto `Recurso`.

B.5.2. Estados de los Elementos

Los `ElementStates` se utilizan para modelar el estado de un elemento.

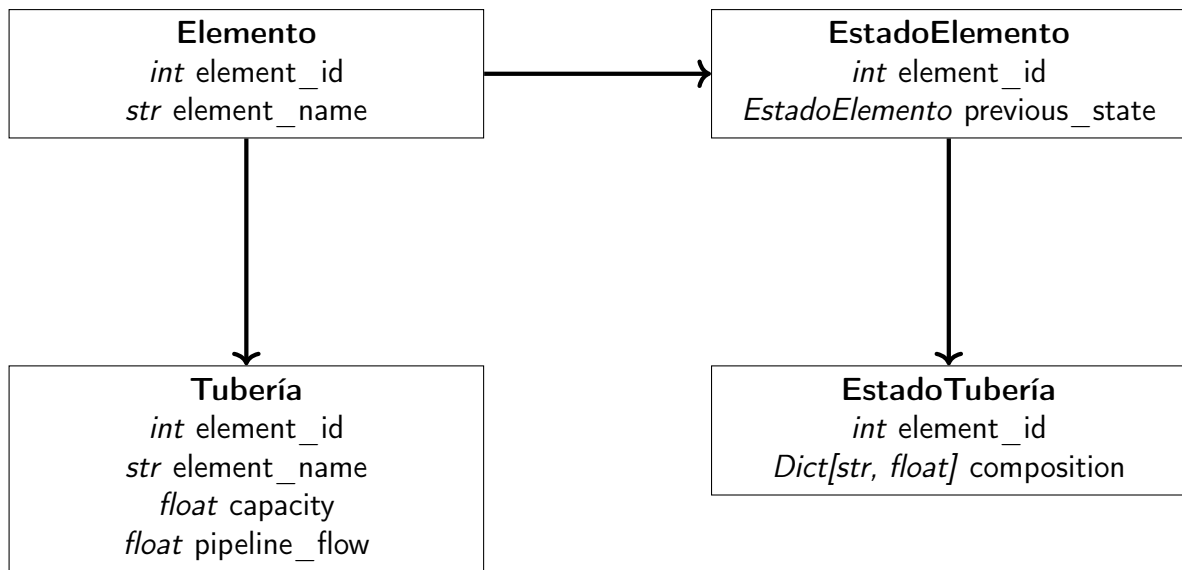
Estado del Tanque

La clase `TankState` contiene propiedades dinámicas sobre un tanque particular, como el volumen actual, y también apuntará a un objeto `Tanque`, que es una subclase de `Elemento`, para verificar propiedades estáticas como su capacidad máxima.



B.5.3. Estado de la Tubería

La clase `PipelineState` contiene propiedades dinámicas sobre una tubería particular, como la composición actual, y también apuntará a un objeto `Tubería`, que es una subclase de `Elemento`, para verificar propiedades estáticas como su capacidad.

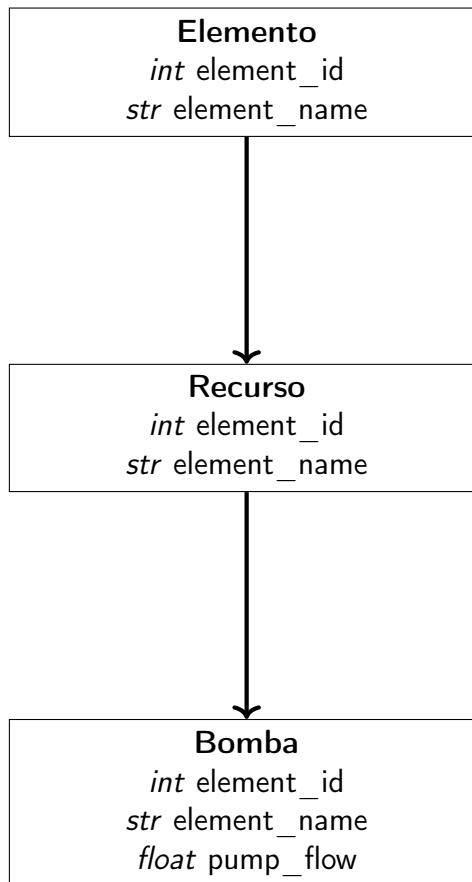


B.5.4. Recursos

Hay muchos recursos que se utilizan en el sistema de refinería. Estos recursos son elementos que no requieren un `ElementState`, y su uso principal es limitar algunas operaciones en la refinería por su disponibilidad.

Bomba

Una Bomba será una subclase de `Recurso` y contendrá propiedades estáticas sobre una bomba particular, como su nombre o flujo de la bomba. No tiene un `ElementState` asociado porque no tiene propiedades dinámicas. La disponibilidad de la bomba se puede rastrear fuera del objeto bomba, por ejemplo, en un atributo `Locking Manager` en el `SystemState`.

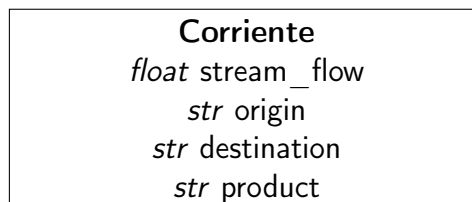


B.5.5. Otras Clases

Hay algunos otros objetos que se utilizan en el sistema de refinería.

Corriente

Un objeto **Corriente** se utiliza para modelar el flujo de un producto de un elemento a otro. Contiene el flujo de la corriente, los elementos de origen y destino, y el producto que se está transfiriendo, entre otros.



B.5.6. Logística de Tanques

Visión general

Esta sección describe la logística y dinámicas operativas del sistema de gestión de tanques. Discutiremos cómo los tanques almacenan y regulan volúmenes, manejan múltiples escenarios de llenado y vaciado, y rastrean estos cambios a través de los estados de tanque (TankState) para mantener una funcionalidad óptima. La siguiente explicación proporciona una comprensión clara de las complejidades involucradas en la gestión de volúmenes de tanques en tiempo real.

Componentes

Existen varios componentes clave en el sistema de gestión de tanques:

- **Tanques:** Almacenan el volumen mínimo y la capacidad máxima de los tanques.
- **TankState:** Contiene el volumen actual del tanque en un momento determinado.
- **Streams:** Mecanismos a través de los cuales los tanques se llenan (aumentando el volumen) o se vacían (disminuyendo el volumen). Los Streams operan dentro del marco temporal de un escenario.
- **Escenarios:** Periodos definidos con tiempo de inicio y fin durante los cuales los Streams afectan a los tanques.

Dinámicas

Los tanques pueden estar sujetos a múltiples escenarios concurrentes donde pueden estar siendo llenados y vaciados simultáneamente. Esto introduce complejidad en la determinación del volumen real de un tanque en cualquier momento dado.

Cálculo de Volumen

Para calcular el volumen disponible de un tanque, se invoca un método que tiene en cuenta todos los escenarios concurrentes que afectan al tanque (`get_concurrent_scenarios_for_element`). El volumen final del tanque después

de aplicar todos los escenarios se almacena en el TankState. Sin embargo, el volumen en cada instante dentro de los escenarios puede calcularse invirtiendo los Streams. Este cálculo detallado permite rastrear la evolución del volumen del tanque a lo largo de la duración de los escenarios concurrentes (`compute_tank_real_volume`).

Volumen más Restringido

El volumen disponible real de un tanque en cualquier momento está determinado por el volumen más restrictivo derivado de los escenarios concurrentes. Esto significa que se toma como verdadero volumen del tanque el volumen más pequeño o más limitante dentro del rango de volúmenes calculados durante estos escenarios (`get_most_restrictive_volume`). De manera análoga, se calcula la capacidad más restrictiva de la misma forma, solo que los criterios para determinarla es el opuesto.

Variables de Tiempo

También se utilizan dos variables para rastrear la evolución del volumen del tanque:

- **Última Hora de Llenado:** El momento más reciente en que el tanque fue llenado.
- **Última Hora de Vaciado:** El momento más reciente en que el tanque fue vaciado.

Estas marcas de tiempo son muy útiles para determinar el estado actual del tanque (si está siendo llenado o vaciado) comparando el tiempo actual con estas marcas. Tienen dos usos principales:

- En caso de que un tanque no esté siendo llenado, no es necesario calcular el volumen más restrictivo al vaciar el tanque, y viceversa. Esto simplifica los cálculos al calcular el volumen disponible, ya que el volumen se da directamente por el volumen actual en el TankState.
- Pueden usarse para seleccionar el mejor tanque para realizar una operación, ya que indican si un tanque está siendo llenado o vaciado actualmente, y proporcionan información sobre cuál fue el último tanque en ser llenado o vaciado.

Apéndice C

Lógica de ventas según medio de transporte

C.1. Ventas por camión

Se utilizan camiones para despachar gasolina A, gasolina D y ULSD.

Hay tanques de despacho de camiones, identificados por subárea `DELIVERY_TRUCK`, desde donde los productos se envían a los camiones, modelados como elementos de tipo `Truck`. Hay un `Truck` para cada uno de los 3 productos despachados. Estos tanques de despacho de camiones operan en modo flotante, lo que les permite ser rellenos a la vez que se realiza un despacho a camiones. Para realizar un seguimiento del volumen enviado a cada elemento de tipo `Truck` crearemos un `ElementState` llamado `TruckState` en el que se almacenará el volumen enviado durante ese día al camión y así poder saber cual es el volumen restante a enviar en cada momento del día.

Los despachos siguen un horario diario de despacho de camiones. Los 3 productos pueden despacharse simultáneamente a sus respectivos camiones. El evento `TruckDispatchEvent` maneja estos despachos, ejecutando 3 tareas del tipo `TruckDispatchTask`.

Todos los tanques de despacho de camiones disponibles para el producto despachado se utilizarán en paralelo, vaciándose a la vez para rellenar los camiones. Todos los tanques se vaciarán a la misma velocidad, lo que significa que puede haber múltiples flujos del mismo producto despachados simultáneamente al mismo camión.

La lógica actual verifica el volumen diario de despacho para cada producto e intenta mantener un flujo continuo desde los tanques de despacho de camiones hacia el camión. Al comienzo de cada día, calculamos una tasa de flujo continua dividiendo el volumen total a despachar ese día por las 24 horas disponibles. También necesitamos dividir este flujo entre todos los tanques de despacho de camiones para ese producto, ya que todos se vaciarán simultáneamente y a la misma velocidad.

Los tanques de despacho de camiones deben ser llenados antes de que estén casi vacíos. Al crear el evento `TruckDispatchEvent`, debemos anticipar cuándo los tanques estarán casi vacíos y detener el evento poco antes de este punto. Para determinar cuándo los tanques están casi vacíos, se considerarán dos criterios particulares:

- El tiempo cuatro horas antes de que se detecte que el tanque está casi vacío al final del escenario generado actual.
- El tiempo en que el tanque alcance el 10% de su capacidad.

El menor de estos dos tiempos se tomará como la hora de inicio de la `RefillTruckTankTask`, teniendo en cuenta que el intervalo resultante entre la hora actual y la hora de finalización sea positivo.

El `TruckDispatchEvent.set_up_state` activará otro evento llamado `RefillTruckTankEvent`, utilizando una tarea del tipo `RefillTruckTankTask` que almacena el tanque a rellenar. Se activará un `RefillTruckTankEvent` por cada tanque que necesite ser rellenado para ese producto. Los tanques de camiones se rellenarán tan pronto como las bombas y las tuberías estén disponibles. Sin embargo, habrá algunas restricciones, ya que estos recursos se comparten no solo con otros tanques de despacho de camiones, sino también con los despachos de barcos y el relleno de los tanques de delivery de tuberías.

Después de activar el evento de recarga, continuaremos despachando a los camiones al mismo ritmo que antes. Dado que nuestros tanques intermedios, operan en modo flotante, podemos rellenarlos mientras los despachamos simultáneamente.

Es importante tener en cuenta que el escenario de despacho finalizará cuando se espere que el tanque esté vacío. Esto significa que el escenario se genera sin considerar una posible recarga en el medio, lo que podría hacer que el despacho termine más tarde. Este enfoque conservador permite verificar si el tanque está realmente vacío en el momento previsto. El comportamiento esperado es que si el

tanque tiene volumen en el tiempo estimado de vacío, el evento de despacho se extenderá hasta el final del día o hasta el próximo tiempo estimado de casi vacío (usando los criterios explicados anteriormente), lo que ocurra primero.

El `TruckDispatchEvent` utiliza un `TruckDispatchScenario` que hereda de `CitgoScenario` pero incluye una nueva variable, `critical_hour_refill`. Esta variable proporciona información sobre cuándo el tanque está casi vacío (usando los criterios explicados anteriormente). En el `TruckDispatchEvent.set_up_state`, esto activa el `RefillTruckTankEvent` utilizando una tarea del tipo `RefillTruckTankTask`.

El evento `RefillTruckTankEvent` busca tanques finales disponibles y conectados que puedan rellenar su tanque de despacho de camiones con el mismo producto que tiene. Si encuentra un tanque final del cual tomar producto, rellenará al flujo máximo permitido por las bombas y líneas en la conexión. Si no encuentra ningún tanque, esperará tanques utilizando `wait_until_tanks_are_available`. Esto puede suceder si todos los tanques finales con el producto requerido están vacíos o bloqueados, si no tienen conexiones con el tanque a rellenar, o si las bombas y líneas requeridas para estas conexiones están bloqueadas por otros flujos.

A continuación se puede observar los resultados de la lógica de ventas por camiones, podemos ver como los tanque sintermedios rellenan los camiones y como estos tanques son rellenos desde los tanques finales.

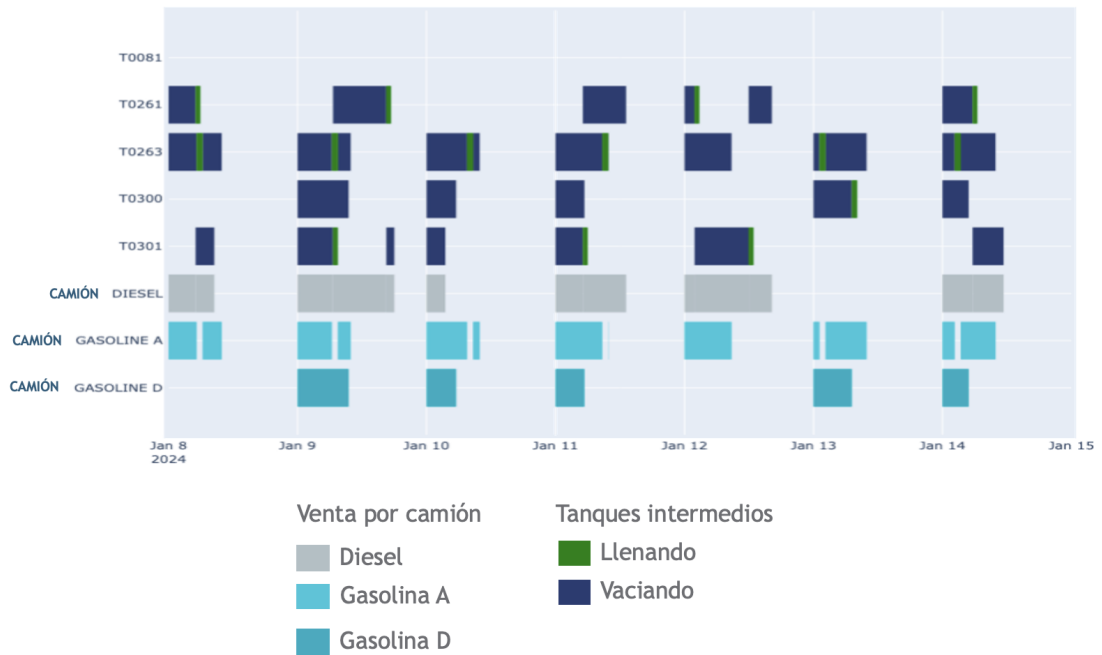


Figura C.1: Resultados funcionamiento ventas de camiones

C.2. Ventas por barco

El puerto marítimo permite despachar diferentes tipos de gasolinas, destilados y GNL (C Reformate y Naftas) en barcos y barcazas.

El puerto tiene la siguiente estructura de muelles y atracaderos:

- **Muelle A**
 - Norte
 - Sur
 - Slip
- **Muelle B**
 - Norte
 - Sur
- **Muelle C:** usualmente utiliza todos los atracaderos simultáneamente.

- Norte
 - Sur
 - Slip
- **Muelle D**
- Norte
 - Sur
 - Slip

Un conjunto de líneas y bombas permiten ciertas conexiones de productos a los muelles.

El tiempo de amarre se modelará como el tiempo requerido para que los barcos ingresen al puerto y comiencen a recibir o despachar productos. El tiempo de amarre para barcos será de 12 horas, y para barcasas será de 10 horas. Otro tiempo de desamarre de 6 horas se modelará para ambos tipos de embarcaciones.

La herramienta de **Planificación Diaria** establecerá un programa de barcos que prescribe un atracadero predeterminado para cada barco y asigna volúmenes diarios para cada embarcación. Si la asignación inicial no es factible, la herramienta de **Planificación Horaria** ajustará tanto los volúmenes como los atracaderos en consecuencia.

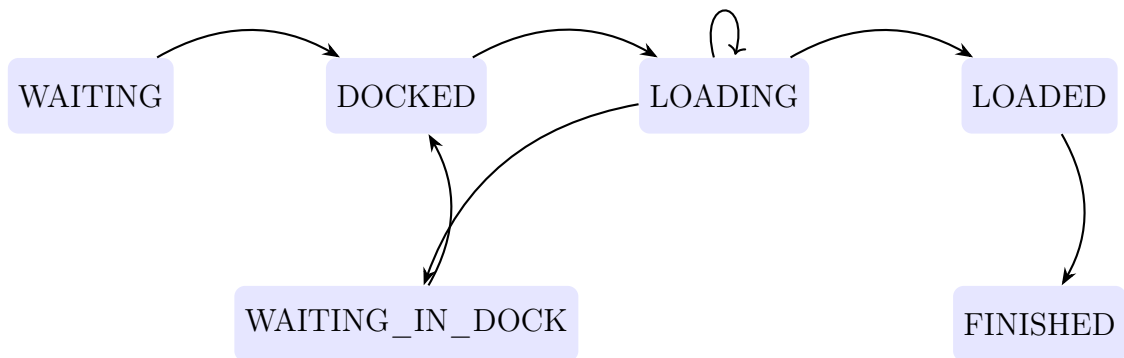
En situaciones donde no se asigna un atracadero, el programador seleccionará el atracadero disponible más apropiado. Se espera que las instancias donde el programa permita servir a los barcos en menos días de los planeados sean raras, ya que la fase de planificación enfrentará menos restricciones; sin embargo, si es factible, el programador debe priorizar la oportunidad de servir a los barcos lo más rápido posible.

Lógica de Despacho de Barcos

El evento `ShipDispatchEvent` maneja la lógica para despachar productos desde los tanques finales a los barcos, así como las operaciones portuarias para la transición entre estados de los barcos, realizar el atraque, etc.

La lógica es una máquina de estados que, basada en el estado actual del barco, registrado en las instancias de `ShipState`, toma una decisión u otra:

- Si el estado del barco está en espera, encontrar un atracadero, reservar volumen en los tanques, establecer el estado a atraque y aplicar el tiempo de atraque. Para más información, consulte la reserva de volúmenes y la gestión de muelles y atracaderos.
- Si el barco está atracado, comenzar a cargar el barco buscando tanques finales para despachar el producto. Esto se puede hacer en paralelo.
- Si el barco está cargando, continuar cargando el barco hasta completarlo, donde se establece como cargado.
- Si el barco está cargado, iniciar el proceso de desatraque y establecerlo como finalizado.



- **WAITING** → **DOCKED**: Buscar muelle, atracadero y tanques y comenzar atraque.
- **DOCKED** → **LOADING**: Buscar tanque y despachar.
- **LOADING** → **LOADING**: Más volumen pendiente.
- **LOADING** → **LOADED**: Carga completada.
- **LOADING** → **WAITING_IN_DOCK**: Capacidad máxima del muelle alcanzada.
- **WAITING_IN_DOCK** → **DOCKED**: Buscar muelle y atracadero y comenzar atraque.
- **LOADED** → **FINISHED**: Desatraque.

Reserva de Volumen en Tanques

Antes de comenzar su proceso de atraque, un barco debe garantizar que tiene suficiente volumen listo en los tanques para finalizar su carga sin interrupciones debido a inventarios faltantes.

El método `get_reserved_tanks` verifica todos los tanques disponibles e intenta encontrar diferentes combinaciones de tanques que, juntos, puedan cumplir con los volúmenes completos del barco.

Si no hay tanques disponibles, debido a estar bloqueados o vacíos, el barco esperará para verificar nuevamente. Si hay tanques disponibles pero no son suficientes para completar los volúmenes del barco, se bloquearán para que no sean consumidos por otros eventos, y el barco esperará a que nuevos tanques estén disponibles.

Cuando un barco encuentra suficientes volúmenes en los tanques para comenzar su atraque, reservará el volumen requerido en estos tanques, en lugar de bloquearlos completamente, para permitir que los eventos de camiones y oleoductos consuman producto de estos tanques, si hay un excedente. Sin embargo, no permitirá que otros barcos consuman de estos tanques, ya que cada tanque puede tener volumen reservado solo para un barco.

Una vez que los tanques están reservados y el atraque comienza, el `ShipState` hará un seguimiento de los tanques pendientes y los volúmenes reservados, y los consumirá durante el proceso de carga, hasta que el despacho del barco esté completo y no queden volúmenes pendientes.

Carga Paralela de Productos

Los métodos `ShipDispatchEvent.group_products_by_volume` y `ShipDispatchEvent.get_dispatch_combinations` optimizan cómo distribuir los flujos de productos en paralelo para maximizar el rendimiento. Cada barco tiene un flujo máximo de entrada que puede recibir de conexiones paralelas; también tiene un atributo que especifica el número máximo de conexiones paralelas.

El flujo máximo de entrada que un barco puede recibir de bombas y líneas paralelas es generalmente mayor que la capacidad individual de la bomba para despachar un solo producto desde un solo tanque. Entonces, para minimizar el tiempo que toma cargar un barco, el algoritmo intenta maximizar la paralelización de productos, agrupándolos primero en pilas de productos equilibradas y luego

ajustando la distribución del flujo de cada una de estas pilas, de manera que las pilas con mayor volumen tengan un mayor flujo que las pilas más pequeñas, de modo que ambas terminen su procesamiento al mismo tiempo.

Gestión de Muelles y Atracaderos

Cada atracadero puede recibir de varias líneas simultáneamente si tiene múltiples conexiones. Cada línea solo puede usar una bomba a la vez. Las limitaciones sobre qué producto puede transportarse al muelle están determinadas por la línea, y el flujo máximo está dictado por la bomba utilizada para transportar el producto.

Desde todos los tanques finales, el producto puede ser bombeado al muelle y atracadero correspondiente utilizando una línea conectada con ese muelle y atracadero, siempre que el producto sea compatible con la línea, y el flujo esté limitado a la capacidad de la bomba.

No hay restricciones horarias para usar el puerto, todos los muelles operan 24/7.

Hay 2 tamaños de barcos operando en el puerto. Cada barco tiene su tipo especificado en el programa de entrada:

- Barcazas (≤ 50 MBBL) que ocupan 1 atracadero. Tienen limitaciones en la capacidad máxima y productos permitidos, basadas en las conexiones.
- Barcos (>50 MMBL) que ocupan un muelle completo, con todos sus atracaderos.

Una barcaza solo puede ocupar un atracadero durante su carga, mientras que un barco puede cambiar entre los diferentes atracaderos en el muelle al que está conectado. La lógica para cada uno de ellos gestionando el atracadero es la siguiente:

Barcazas:

- Antes de comenzar el atraque, encontrar un atracadero adecuado en el muelle que no esté bloqueado y sea compatible con todos los productos en el programa.
- Si no se encuentran atracaderos disponibles en el muelle, verificar si hay algún otro muelle con atracaderos adecuados para conectar y, de ser así, cambiar de muelle.

- Bloquear el atracadero durante toda la operación y almacenar el atracadero seleccionado en el objeto `ShipState`.

Barcos:

- Antes de comenzar el atraque, verificar que todos los atracaderos en un muelle estén desbloqueados y disponibles para el barco.
- Si el muelle especificado en el programa no está disponible, verificar otros muelles que estén disponibles.
- Durante el proceso de carga, considerar los atracaderos que sean compatibles con cualquiera de los productos pendientes en el programa.
- Se pueden usar diferentes atracaderos simultáneamente durante las cargas paralelas de productos.
- Solo mientras el proceso de carga esté en curso, en cada iteración, el objeto `ShipState` almacenará qué atracaderos se están utilizando para cada producto, y los priorizará al continuar el despacho de esos productos, para evitar cambios constantes entre atracaderos.
- Al cargar el barco, verificar que se cumpla la capacidad máxima del muelle que se está utilizando, y cuando el barco esté cargado a la capacidad máxima del muelle, establecerlo en un estado `WAITING_IN_DOCK`, después del cual debe buscar otro muelle.

A continuación se pueden observar dos imágenes que muestran el funcionamiento de la lógica de envíos por barcos. Se puede ver cómo los tiempos de amarre son diferentes para los barcos y las barcasas. En algunos barcos, se carga el producto uno a uno, ya que son barcos de un solo producto, mientras que en otros se carga lo que llamamos "Mixto", que se refiere a varios productos a la vez. También se puede ver como una vez ha terminado el proceso de amarre los barcos comienzan directamente la carga, ya que los volúmenes de producto han sido reservados con anterioridad.

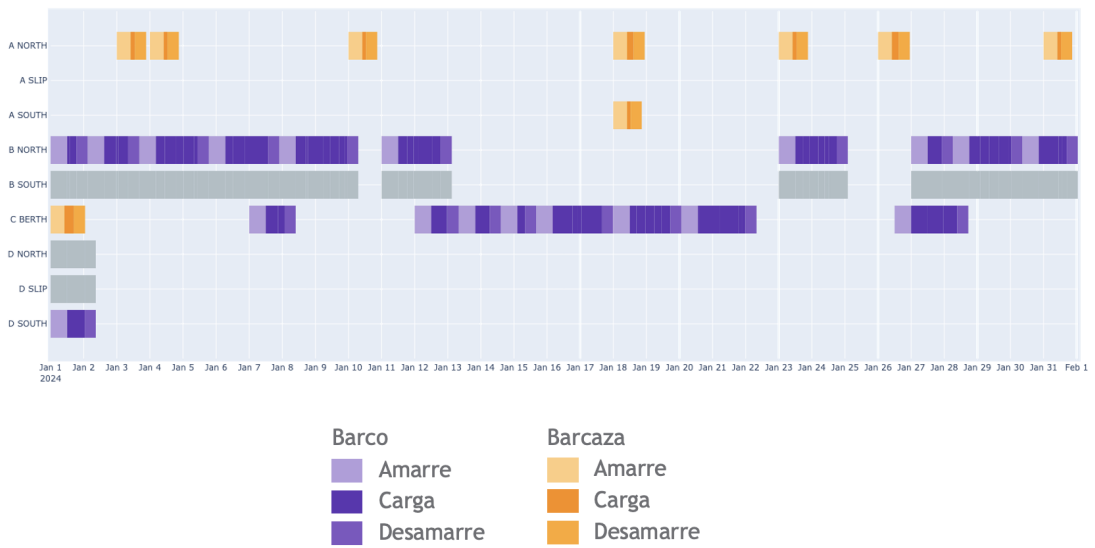


Figura C.2: Resultados funcionamiento ventas de barcos

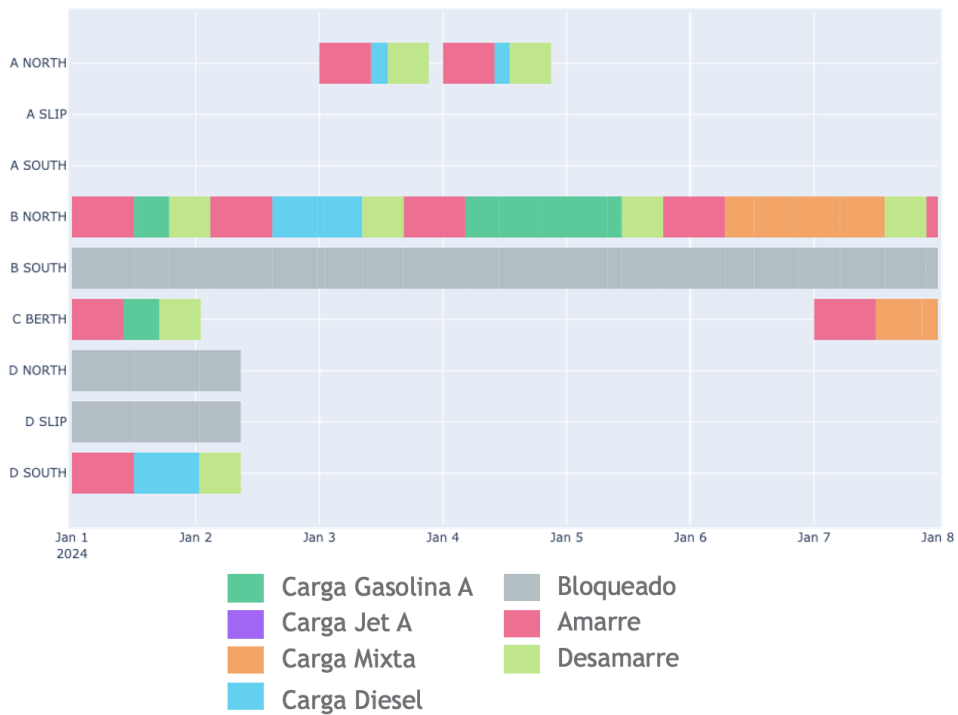


Figura C.3: Resultados funcionamiento ventas de barcos

C.3. Ventas por tubería

Hay dos tuberías que se utilizan para las ventas de gasolinas y destilados. Las llamaremos tubería A, que conecta los tanques finales con tanques intermedios de almacenamiento, y tubería B, que es la utilizada para realizar las ventas.

C.3.1. Tubería B

Se utiliza para vender gasolinas y destilados. Recibiremos un programa de ventas con un plan diario detallado, que modelará los despachos desde los tanques de intermedios de almacenamiento hacia la tubería B, vaciando los tanques. Hay dos tubería de tipo B, una para gasolinas y otra para destilados.

C.3.2. Tubería A

La tubería A conecta los tanques finales con los tanques intermedios destinados a la venta por tubería.

El objetivo principal de los envíos por esta tubería es rellenar los tanques intermedios con suficiente volumen antes de cada venta programada por la tubería B, para garantizar que la venta pueda cumplirse, en las fechas especificadas por el programa de despacho dado por la herramienta de **Planificación diaria**, sin interrupciones.

Los valores máximos por ciclo (periodo de 5 días) y producto son los siguientes:

- **Jet A:** 251 MBBL/ciclo
- **ULSD:** sin límite
- **Gasolina grado D:** 160 MBBL/ciclo
- **Gasolina grado M:** 75 MBBL/ciclo
- **Gasolina grado A:** sin límite

Estos valores están establecidos por la capacidad máxima de los tanques intermedios de cada producto. La gasolina grado A y el ULSD no tienen límite porque pueden funcionar como tanques flotantes (enviando y recibiendo producto a la

vez), por lo que podemos bombear volumen mientras la tubería B está despachando.

Hay 3 tuberías de tipo A: una para gasolinas, otra para Kerosenos y otra para Diesel. Solo se puede enviar un producto a través de cada tubería simultáneamente. El volumen máximo por día es de 240 MBBL para cada tubería. Esto nos impone unas restricciones de flujo.

La lógica para los envíos a través de las tuberías de tipo A está implementada en el evento *RefillIntermediateTankEvent*.

Ciclos de envío a los tanques intermedios: tiempos de inicio y fin

Los tiempos de inicio y fin de los envíos a los tanques intermedios están determinados por el programa creado por la herramienta de **Planificación diaria**. Al final de cada ciclo de venta por la tubería B, cuando los tanques se han vaciado (total o parcialmente), el envío a los tanques intermedios correspondiente al siguiente ciclo de ventas comienza a rellenar los tanques. El objetivo de los tanques intermedios es terminar su relleno antes de que comience el siguiente ciclo de ventas. En el caso del ULSD y la gasolina A, se permite seguir rellenando los tanques intermedios mientras el siguiente ciclo de ventas ya está en curso, ya que son tanques flotantes, recibiendo y enviando al mismo tiempo.

Por ejemplo, consideremos el siguiente programa de ventas de gasolina:

FECHA	PRODUCTO	VOLUMEN	CYCLE_ID
2024-04-01	Gasolina A	10	1
2024-04-02	Gasolina A	10	1
2024-04-03	Gasolina A	10	1
2024-04-04	Gasolina M	10	1
2024-04-05	Gasolina M	10	1
2024-04-06	Gasolina A	10	2
2024-04-07	Gasolina A	10	2
2024-04-08	Gasolina A	10	2
2024-04-09	Gasolina M	10	2
2024-04-10	Gasolina M	10	2

Esto resultaría en la siguiente tarea de relleno de los tanques intermedios de gasolina M, justo entre los ciclos de despachos de gasolina M en el programa de

ventas:

FECHA_INICIO	FECHA_FIN	PRODUCTO	CYCLE_ID
2024-04-06	2024-04-08	Gasolina M	2

Y también resultaría en la siguiente tarea de relleno de los tanques intermedios de Gasolina A, donde se permite operar en tanques flotantes, terminando al final de cada ciclo de ventas.

FECHA_INICIO	FECHA_FIN	PRODUCTO	CYCLE_ID
2024-04-04	2024-04-08	Gasolina A	2

Ciclos de envío a los tanques intermedios: volúmenes

Los volúmenes de cada ciclo de envío a los tanques intermedios deben garantizar que se cumpla y despache el volumen total programado del ciclo de ventas. La herramienta de **Planificación diaria** proporcionará un conjunto optimizado de volúmenes por día y ciclo, considerando cuándo los volúmenes pueden incrementarse más allá del mínimo requerido para cumplir con el ciclo de ventas, para maximizar el rendimiento.

La herramienta de **Planificación horaria** agregará los volúmenes para cada producto enviado y ciclo y los enviará lo antes posible, asegurando que se cumpla el volumen mínimo requerido. Es importante recibir de la herramienta de **Planificación diaria** el *cycle_id* al que pertenecen estos volúmenes, así como las fechas de inicio de cada despacho desde los tanques finales hasta los tanques intermedios de almacenamiento.

Es posible tener volumen restante en los tanques intermedios al comienzo del ciclo. Este volumen debe considerarse al calcular el volumen a enviar a los tanques en el próximo ciclo. Esta decisión la toma la herramienta de **Planificación diaria**.

También es posible que la herramienta de **Planificación horaria** no logre enviar todo el volumen durante el ciclo, pero si los tanques intermedios tienen inventario excedente, y es suficiente para completar la venta, la simulación continuará sin interrupción. En este caso, la herramienta de **Planificación horaria** agregará el volumen faltante al próximo ciclo, por lo que el volumen total enviado a los tanques intermedios seguirá siendo el mismo. La simulación solo se detendrá en caso de que una venta no pueda cumplirse según lo programado debido a la falta de producto en los tanques intermedios de almacenamiento.

Objetos físicos a tener en cuenta

Hay 3 tuberías de tipo A: una para gasolina, otra para Kerosenos y otra para Diesel. Estas tuberías envían producto a cinco tanques intermedios, uno para cada producto: Gasolina A, Gasolina M, Gasolina D, Jet A y ULSD.

Los tanques de Gasolina A y ULSD pueden operar como tanques flotantes (envío y recepción simultáneos).

Generación de tareas *RefillIntermediateTankTask*

Cada tarea de tipo *RefillIntermediateTankTask* está asociada con el evento *RefillIntermediateTankEvent*. Representa un producto que se enviará a los tanques intermedios, con sus tiempos de inicio y fin representando la duración del ciclo virtual y el volumen restante por despachar a los tanques durante ese ciclo.

Las tareas de tipo *RefillIntermediateTankTask* se generan de 2 maneras:

- Al comienzo de la simulación, se crea una tarea para cada ciclo definido, para cada producto, según lo definido por el programa de ventasy la entrada dela herramienta de **Planificación diaria**.
- Al final de cada iteración procesando una tarea, en caso de que aún haya volumen pendiente por rellenar, se creará una nueva tarea con el volumen pendiente.

Restricciones a tener en cuenta

Específicamente, los tanques intermedios de ULSD y Gasolina A operan como tanques flotantes. Por lo tanto, no tenemos que ser tan restrictivos con el volumen enviado a ellos durante cada ciclo virtual. Esta flexibilidad se aplica a estos dos casos porque podemos hacer ajustes mientras enviamos productos a través de la tubería B para su venta. Debemos bloquear los tanques intermedios de Gasolina D, Gasolina M y Jet A para no poder vender producto desde ellos a la vez que los rellenamos de producto. Cuando necesitamos realizar una venta, el tanque debe estar disponible; el rellenado de dicho tanque debe haberse completarse de antemano.

Debido a que los eventos de relleno de camiones y los eventos de despacho de camiones tienen prioridad, antes de procesar cada tarea de *RefillIntermediateTankTask*, debe asegurarse de que no se necesiten tanques finales para rellenar un tanque de entrega de camiones. Por lo tanto, las prioridades de las tareas y el

intervalo de cada escenario se ajustarán en consecuencia.

Restringiremos el intervalo del escenario de *RefillIntermediateTankEvent* basado en el inicio de una *TruckDispatchTask* o una *RefillTruckTankTask*. Esto garantiza la prioridad para rellenar los tanques de los camiones, evitando que se bloqueen por un escenario de *RefillIntermediateTankEvent* en curso cuando sea necesario.

Ejemplo:

Hora	Evento
00:00 - 10:00	TruckDispatchTask A
00:00 - 10:00	RefillIntermediateTankTask A
10:00 - 24:00	TruckDispatchTask B
20:00 - 22:00	RefillTruckTankTask B
10:00 - 20:00	RefillIntermediateTankTask B

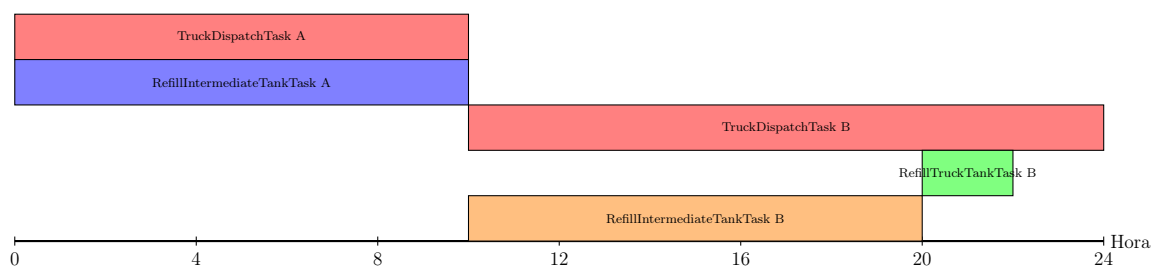
Cuadro C.1: Línea de tiempo de eventos

Secuencia de eventos:

- **TruckDispatchTask A:** Comienza a las 00:00 y termina a las 10:00.
- **RefillIntermediateTankTask A:** Comienza a las 00:00 y termina a las 10:00, ya que comienza una tarea de despacho de camiones (*TruckDispatchTask B*).
- **TruckDispatchTask B:** Comienza a las 10:00 y termina a las 24:00. Se activa una tarea de despacho de camiones a las 20:00 debido a que los tanques de los camiones están cerca de vaciarse.
- **RefillTruckTankTask B:** Comienza a las 20:00 y termina a las 22:00.
- **RefillIntermediateTankTask B:** Comienza a las 10:00 y termina a las 10:00, ya que comienza una tarea de despacho de camiones (*RefillTruckTankTask B*).

Representación Diagrama

Figura C.4: Línea de tiempo de eventos



A continuación se muestra una imagen del funcionamiento por tubería. Podemos observar cómo el producto llega desde la Tubería A, llena los tanques intermedios y sale por la Tubería B. Se puede ver que los únicos tanques que pueden flotar (llenarse y vaciarse a la vez) son los de Gasolina A y Diesel, tal como lo restringe la lógica. Además, se observa cómo las gasolinas se envían de manera secuencial por la Tubería B.

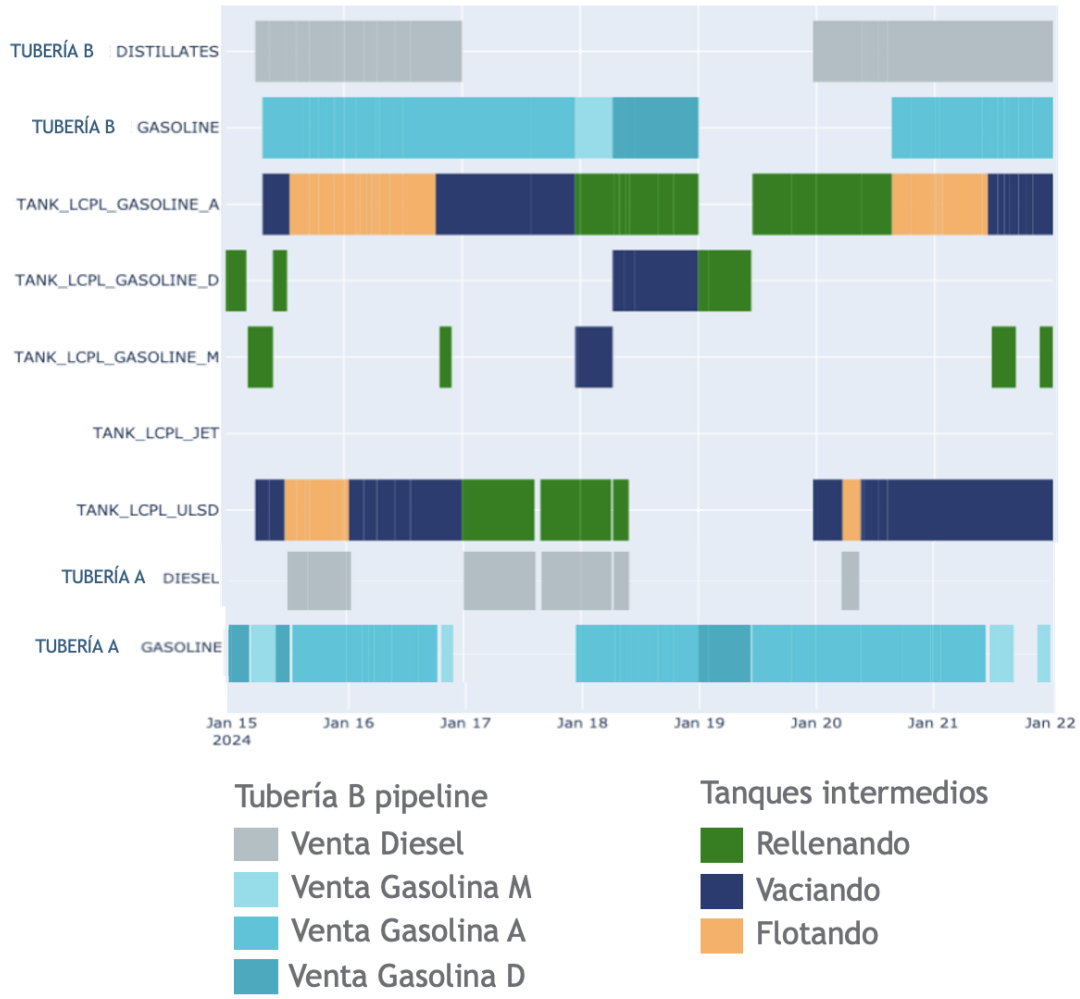


Figura C.5: Resultados funcionamiento ventas por tubería