

Informe desafío N.º 2

Diego Andrés Pedrozo Figueroa

Facultad de Ingeniería, Universidad de Antioquia

Informática II

Aníbal José Guerra Soler

17 de octubre de 2024

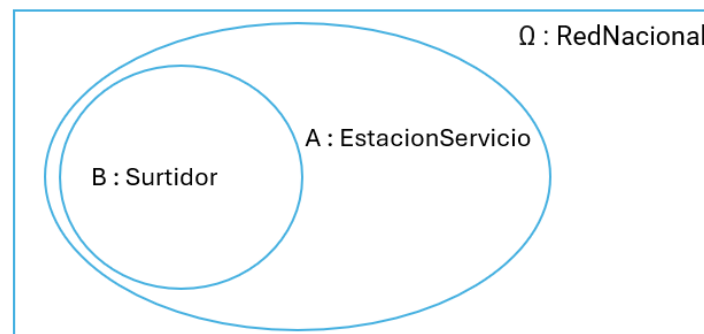
Introducción

En el mundo de la programación existen diferentes paradigmas o estilos de programación, los cuales pueden ser clasificados según la metodología propuesta a la hora de desarrollar una solución mediante un programa de software para un problema dado. En la vida cotidiana podemos evidenciar interacciones entre entidades, si queremos analizar el comportamiento de dichas interacciones el método o paradigma de programación más adecuado sería la programación orientada a objetos. Siendo así, es de interés modelar el comportamiento de una red nacional de estaciones de servicio, las cuales, a su vez, están conformadas por una serie de surtidores. Es importante tener en cuenta la interacción que podría tener un cliente u otro tipo de interacciones para gestionar la red nacional.

Análisis del problema

La situación descrita en el problema muestra una estrecha relación entre varias entidades que posteriormente fueron clasificadas como clases: clase *RedNacional*, clase *EstacionServicio* y clase *Surtidor*. Fácilmente es posible relacionar estas tres entidades a manera de notación de conjuntos, lo cual permite identificar el orden en que las clases y sus respectivos métodos debían ser definidos y posteriormente implementados teniendo en cuenta la forma en que el compilador analiza el código, usando *declaraciones anticipadas* para eliminar dependencias entre clases. El siguiente diagrama de Venn permite hacer una aproximación al diagrama de clases que se verá más adelante.

$$B : \text{Surtidor} \subset A : \text{EstacionServicio} \subset \Omega : \text{RedNacional}$$



Pese a que se pudo evidenciar una estrecha relación entre las clases, fue necesario tener mucho cuidado para no crear dependencias entre las clases, o situaciones donde la

manipulación y la creación de relaciones entre clases hicieran de la búsqueda de la solución un camino más dispendioso.

La mayoría de los atributos de las clases fueron proporcionados en el planteamiento del problema. Surgió entonces la necesidad de recopilar esta información y realizar un proceso de selección de la información que resultaba más importante o que de verdad influyera en la solución. Un buen desarrollo de la solución debería empezar desde la identificación de los atributos, para posteriormente relacionarlos y poder identificar los posibles métodos que tendría cada clase. Algunos de estos atributos fueron determinados de forma automática, otros de forma manual o directa según la información ingresada por el usuario. A continuación, entre otras cosas, hablaremos sobre las consideraciones tenidas a la hora de determinar cuándo un atributo era posible generarlo / controlarlo de forma automática o no.

Consideraciones para la alternativa de solución propuesta

“Separando la paja del trigo...” Ciertamente el problema estaba cargado de datos que por supuesto están presentes en el mundo real, pero algunos de estos fueron ignorados a la hora de desarrollar la solución, pues no fueron considerados relevantes para realizar los cálculos necesarios para dar solución al problema. Por ejemplo, un tipo de información que fue descartada corresponde al gerente de una estación de servicio.

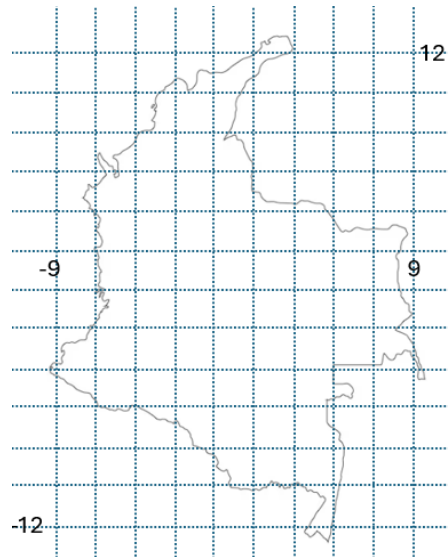
Gran cantidad de datos → Análisis → Extracción de la información relevante

¿Atributos automáticos?... En la ruta de solución propuesta se definió que la gestión y el comportamiento de algunos atributos de algunas clases debían realizarse de forma automática, es decir, el usuario no interfiere directamente en estos procesos, sino más bien el programa actúa de acuerdo a las acciones del usuario.

Es el caso de la gestión de los códigos identificadores, tanto de los surtidores como de las estaciones de servicio, estos fueron determinados por en un rango de [1000, 9999] para los surtidores, y en el rango [100, 999] para las estaciones de servicio, suponiendo que en una red nacional no se presentan más de 999 estaciones de servicio ni 9999 surtidores, aunque claramente estos valores pueden ajustarse más a un caso específico de la realidad.

Otro atributo gestionado automáticamente fue el de la región, que estaba determinado por unas coordenadas geográficas y teniendo en cuenta que el enunciado del

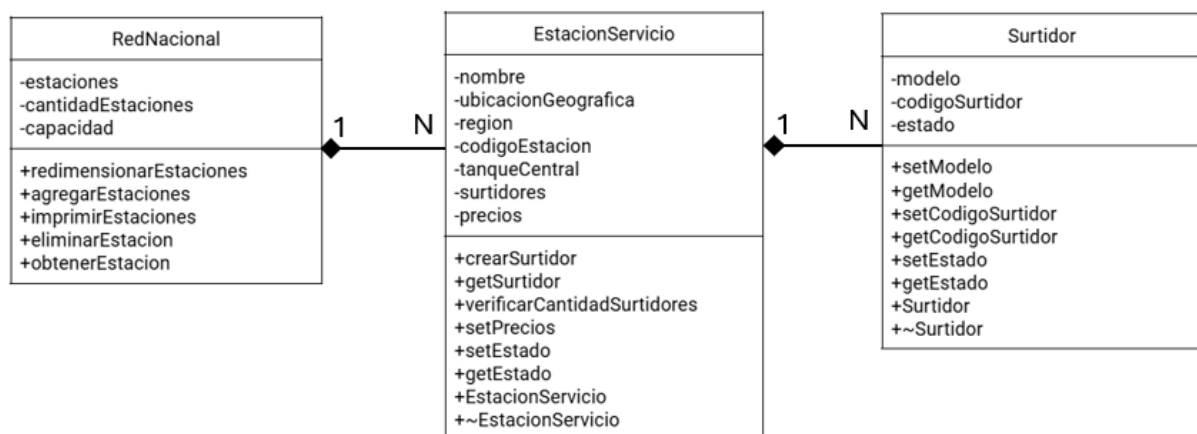
problema localizaba la red nacional de estaciones en Colombia, entonces se establecieron valores aproximados para la latitud y longitud geográficas que pudiera tener la estación. Así pues, se realizó un mapeo para determinar la región a la cual pudiera pertenecer una región creada, y teniendo esto en cuenta poder determinar más fácilmente el precio del combustible para una determinada estación de servicio.



Latitud = [-12, 12] | Longitud : [-9, 9]

¿Por qué todas las clases están definidas en un mismo archivo?... En un principio, por cuestiones de estética y organización, pensé en definir las clases en archivos separados *header* y *source* separados, sin embargo esto podría dificultar el llamado de funciones en caso de que no se tengan tres clases sino cien por ejemplo en un mismo proyecto... no sería lo más recomendable.

Diagrama de clases



Problemas de desarrollo

Ciertamente la complejidad analítica que conllevaba la solución del problema planteado no era muy alta, sin embargo la cantidad de tareas a realizar se hacía cada vez más y más grande a la hora de: 1) definir cada clase, pensando en sus atributos, la modificación y obtención de los mismos y 2) al relacionar cada una de las tres clases creadas según las diferentes funcionalidades y comportamientos que podrían presentarse en la vida real.

Evolución de la solución

Ciertamente un buen análisis del problema antes de lanzarse a resolverlo es el mejor cimiento para las sólidas bases de una buena solución. Es por esto que se procedió a dedicar un buen tiempo (aproximadamente día y medio) al desglose de la información, la identificación de atributos y el posterior diseño de métodos, con el fin de poder ver el cuadro completo y no tener frecuentes tropiezos y replanteamientos de posibles soluciones durante el desarrollo de la solución debido a la falta de un buen análisis.

Consideraciones a tener en cuenta en la implementación

- El problema exigía varias cosas, cuando se finaliza la simulación la información de las estaciones de servicio o los surtidores creados se pierde, no se implementa un sistema de recuperación de información para reconstruir la red.