**Inteligencia Artificial**

**Trabajo Práctico N° 1: Búsqueda**

Martín Destéfanis – tinchodestefanis@hotmail.com

Gastón Zianni – gaston\_zianni@hotmail.com

**Resumen**

En este trabajo se presenta la implementación de un agente virtual con el objetivo de dar respuesta al problema que surge cuando uno debe realizar la compra de productos y no sabe a qué negocio ir teniendo en cuenta el costo de traslado, el tiempo y el mismo costo del producto que se quiere adquirir. El agente otorga una solución basado en los requerimientos anteriormente descritos y, a su vez, tiene en cuenta que su entorno de movilidad estará acotado a un barrio determinado de la ciudad de Santa Fe.

**Introducción**

El entorno de acción del agente se encuentra delimitado por las calles Av. Gral. Paz, Regimiento 12 de infantería, Av. Almte. Brown y Pedro de Vega. A su vez, se indica que siempre el agente dará como solución un camino recomendado para la realización de la compra indicada comenzando en la esquina determinada por las calles Juan Castelli y Antonia Godoy. A modo de ilustración se muestra en la Figura 1.1 el entorno en dónde el agente operará.

Para poder presentar una solución el agente necesitará el conocimiento de los negocios que se encuentran y sus productos asociados con sus respectivos precios. También, deberá tener información sobre el costo necesario para el traslado, es decir, si por ejemplo se tratara de un medio de transporte automovilístico el agente deberá conocer el precio de la nafta.



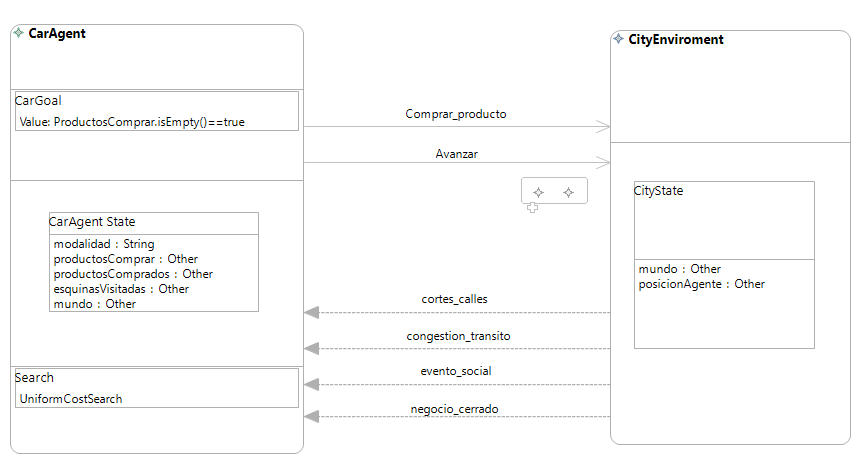
*Figura 1 – Entorno de acción del agente*

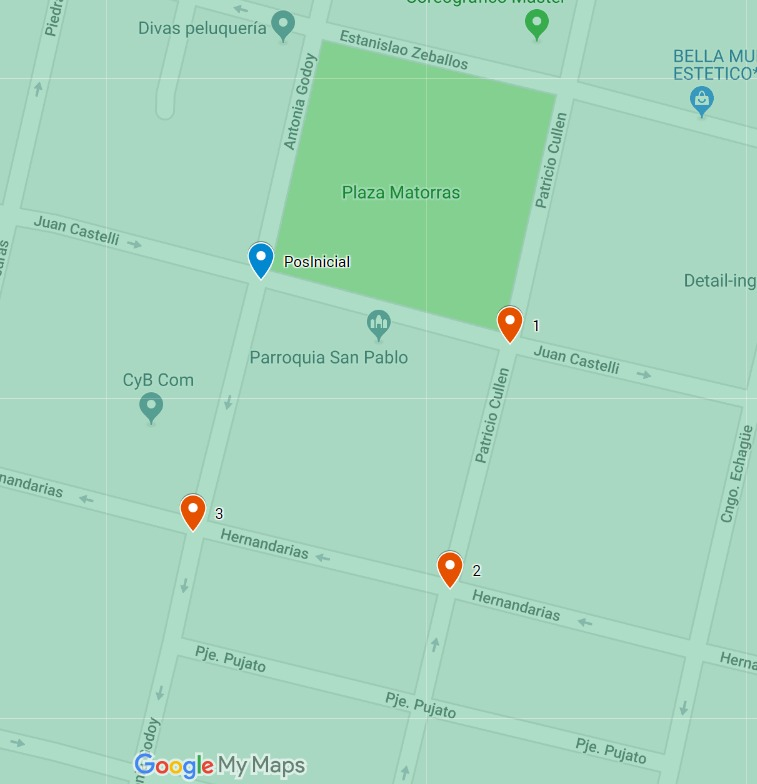
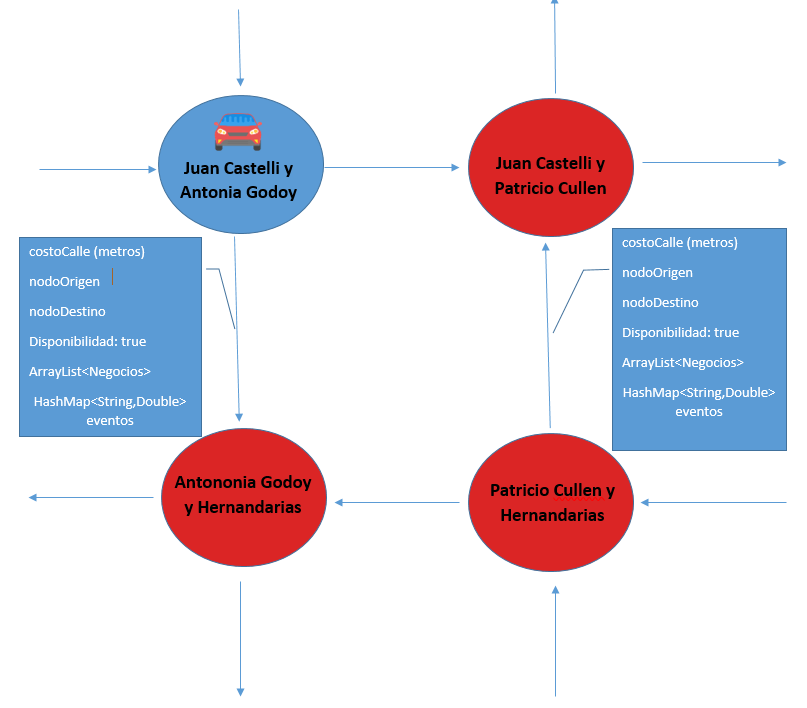
**Solución**

# Definición conceptual del agente

Para la descripción del agente, su estado, meta de prueba, acciones, percepciones y el ambiente en donde va a estar inmerso se utilizó el framework IDEMIA que permite realizar un modelo indicando todos los aspectos nombrados. Esto se ilustra en la Figura 2.

El modo de representar el ambiente elegido es una estructura de grafo dirigido en donde cada nodo representará una esquina en particular y cada arco, saliente, significará la existencia de una calle con su orientación correspondiente que conectará un nodo con otro. Para ser más específicos, se ilustra en la Figura 3 esta representación elegida con la salvedad que por una cuestión de simplicidad queda representado sólo una porción del entorno de acción del agente.



*Figura 2 - Representación del agente y el ambiente mediante IDEMIA*

*Figura 3 - Representación del ambiente a nivel de software con una estructura de grafo dirigido*

Para determinar una definición conceptual del agente se describen las siguientes características.

Totalmente observable

El agente posee información sobre el ambiente que en este caso es el mapa de la ciudad de Santa Fe delimitado por las calles Av. Gral. Paz, Pedro de Vega, Av. Almirante Brown, y Regimiento 12 de Infantería. También conoce los negocios disponibles, los productos y los precios asociados a los mismos, así como también los distintos eventos que pueden provocar un cambio en la decisión del agente.

Determinístico

Se sabe cómo va a quedar el ambiente luego de realizar una acción.

Secuencial

Un caso en dónde se puede visualizar esta característica es cuando el agente al comprar un producto, éste se elimina de la lista de productos disponibles y por ende no se tendrá en cuenta en la próxima decisión.

Estático

En cada toma de decisión el agente actúa en base al conocimiento obtenido en ese momento por lo que la introducción de un cambio recién tendrá efecto en la próxima toma de decisión.

Discreto

La determinación de las acciones de movilidad, la percepción de los negocios, precios y eventos forman un conjunto finito que caracteriza al medio como discreto.

Agente individual

Hay un único agente que interactúa en el medio.

## El agente y sus percepciones

Se define el estado inicial, estado final y prueba de meta del agente. La posición inicial identificará el nodo en el que se encuentra al inicio del problema. En nuestro caso será en el nodo que represente la intersección entre Juan Castelli y Antonia Godoy.

El estado final estará determinado por el nodo en que se encuentre el agente y que su lista de productos a comprar esté vacía. Por ende, la prueba de meta será exitosa en el caso en que la lista de productos a comprar que posee el agente se encuentre vacía.

* Estado inicial: (modalidad, productosComprar, productosComprados, mundo, esquinasVisitadas)
* Estado final: (\_, { } ,\_,\_,\_)
* Prueba de meta: IF (productosComprar.isEmpty()==true ) THEN éxito

El agente recibirá como objetivo una lista de productos a comprar. Luego el algoritmo de búsqueda seleccionado determinará en cada iteración su acción a realizar, ya sea avanzar o comprar un producto. El producto se compra luego de haber llegado a la esquina próxima en la que se encuentra el negocio.

Para poder realizar éstas acciones se definen el conjunto de percepciones y operadores que poseerá el agente.

## Percepciones

* Poder reconocer qué calles se encuentran cortadas. Esta afecta a todos los tipos de modalidades (Bici, automóvil).
* Poder reconocer si la calle seleccionada para realizar el recorrido se encuentra con una congestión de tráfico lo que afectará a las restricciones de tiempo y costo de nafta en el caso de que se trate de una movilidad mediante un automóvil.
* Poder reconocer si hay un evento social. Este afectará únicamente a la modalidad “Bicicleta”

Acciones

* Avanzar: permitirá al agente moverse de un nodo a otro. Se define, a continuación, el método utilizando pseudocódigo.

IF (posiciónActual.getEnlace().get(i).isDisponible)

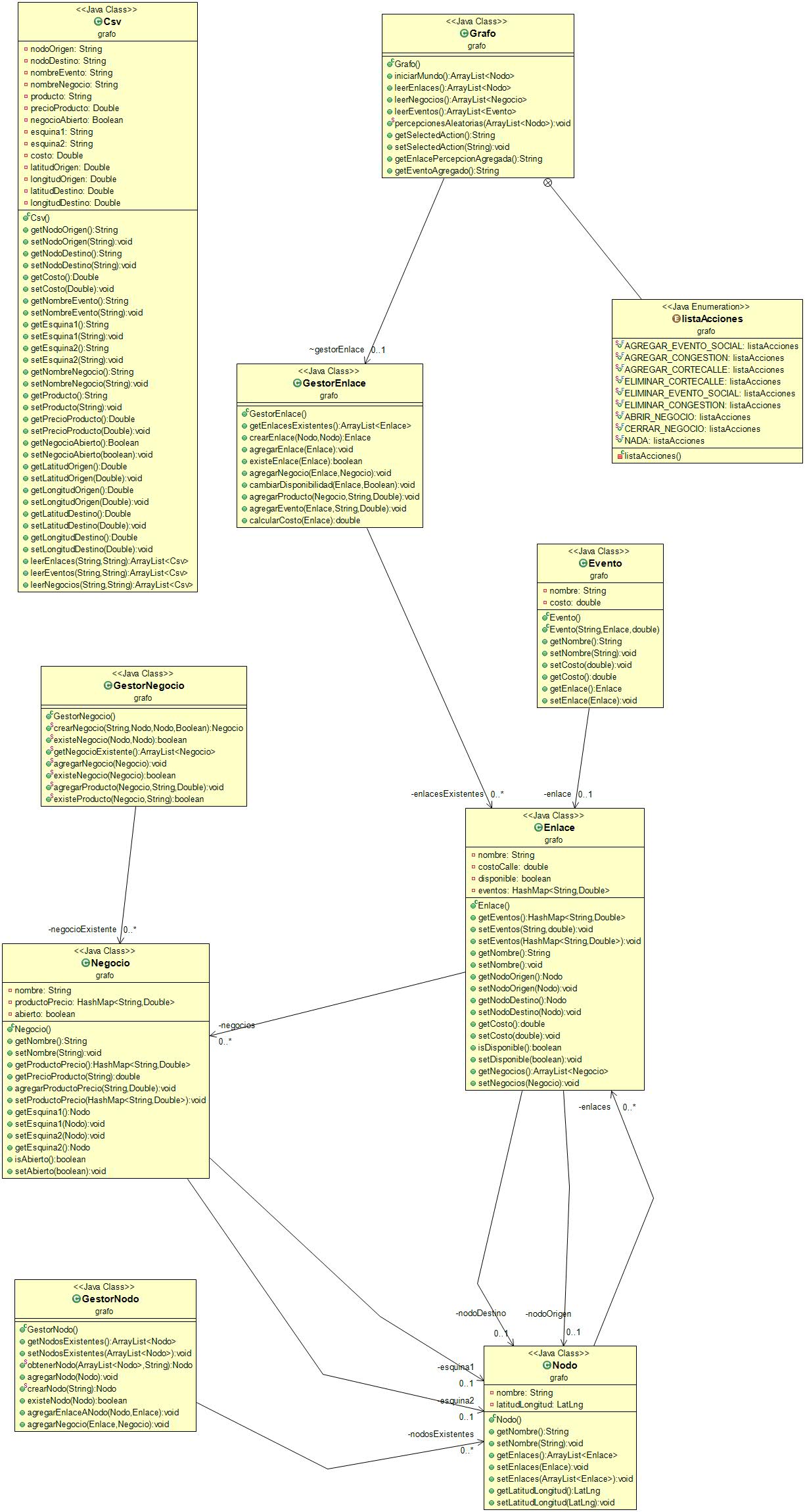
THEN posiciónActual = posiciónActual.getEnlace().get(i).getNodoDestino

* ComprarProducto: le permitirá al agente poder realizar la adquisición de un producto en un negocio. Cumpliendo así que el producto se elimine de la lista de los productos a comprar y se lo almacene en la lista de productos comprados. Nuevamente se define el método mediante pseudocódigo.

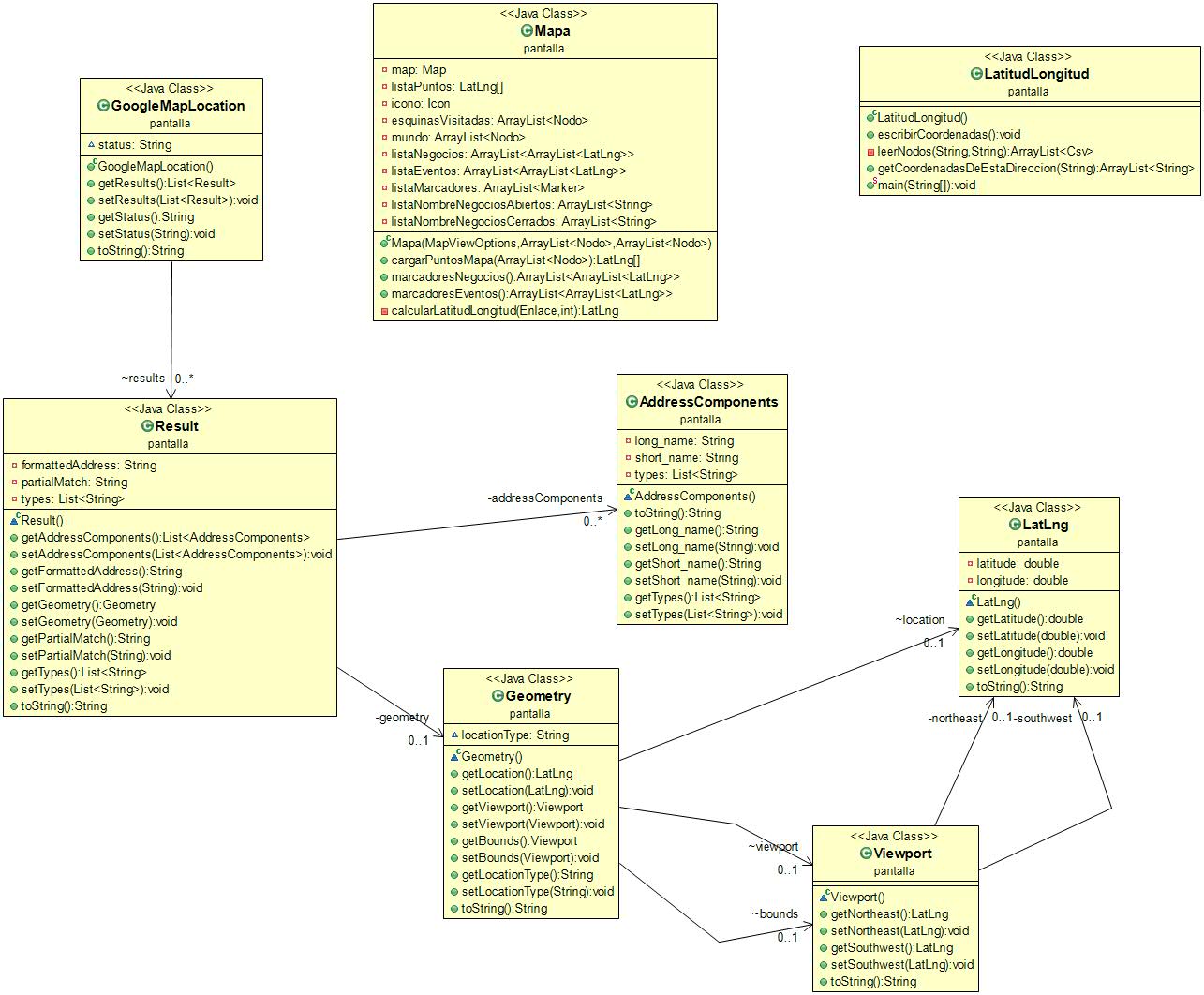
IF (negocio == Enlace.getNegocio().get(i) AND Enlace.getNegocio(i).isAbierto())

THEN sacar producto de listaProductos y agregar en listaProductosComprados

## Diseño de la solución



*Figura 4 – Diagrama de Clases del grafo utilizando en nuestra solución*



*Figura 5 – Diagrama de Clases de la interfaz ultilizada*

## Implementación de la solución

Para analizar y comparar diferentes estrategias se partió de un mismo escenario con el agente en Juan Castelli y Antonia Godoy.

Los supermercados con sus productos:

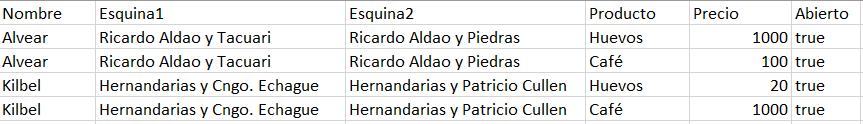
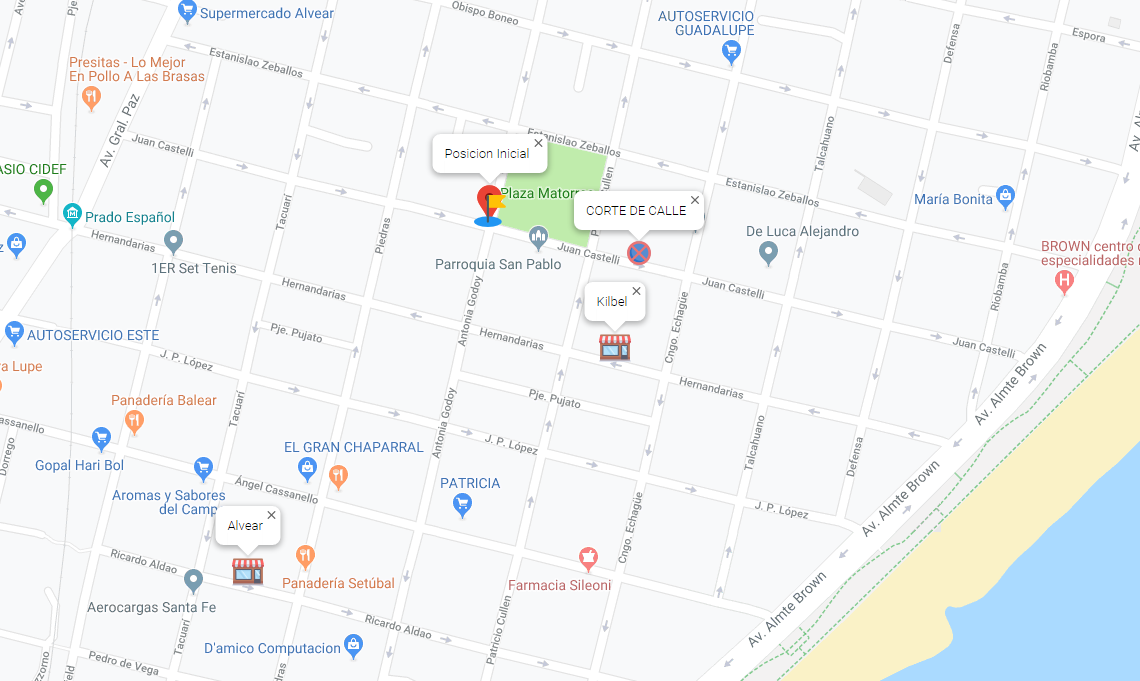


Figura 6 – Datos de los negocios cargados

Los eventos son generados por iteración, pero además hay uno pre cargado que es:



Figura 8 – Mapa visual del estado del agente al inicio

Figura 7 – Datos de los eventos cargados

Hay 3 modalidades de uso del agente:

* En Automóvil donde el avanzar tiene un costo establecido a partir de los metros de una cuadra y el precio de combustible. La acción ‘ComprarProducto’ tiene un costo equivalente al producto comprado. Los eventos que afectan a este agente son los cortes de calle o congestión. En esta modalidad del agente se optimiza precio.
* En Bicicleta donde el avanzar tiene un costo de 1 minuto (esto es debido a que números menores a 1 no responde muy bien o demora demasiado en costo uniforme). La acción ‘ComprarProducto’ tiene un costo equivalente a 5 minutos. Los eventos que afectan a este agente son los cortes de calle o eventos sociales. En este agente se optimiza tiempo.

Aclaración: por convención los eventos sociales solo afectar a la modalidad en bicicleta con la idea de diferenciar aún más las modalidades entre sí y, por ende, las soluciones.

Para simplificar el informe vamos a resolver con las distintas estrategias solo la modalidad Automóvil.

## Estrategias no informadas

## Costo Uniforme

## Suponiendo que un auto hace 11[km/litro] entonces por metro consume 0.0000909091 [litro/m]

Asumo que la nafta está $50 el litro, entonces: el costo por metro de nafta sería: $0.0045454545

Al ser un costo tan pequeño el costo de avanzar, demora muchísimo costo uniforme en llegar a la solución. Por lo que contamos con poner un precio por metro de 0.45454545 (es ilógico este precio, pero se lo compensó con los precios de los productos para obtener una solución adecuada).

Como podemos ver al tener un corte de calle en la dirección que se tendría que dirigir recalcula e igual considera que tiene que comprar huevos en el Kilbel y café en el Alvear.

Recordamos que los eventos sociales no afectan a esta modalidad, si las congestiones.

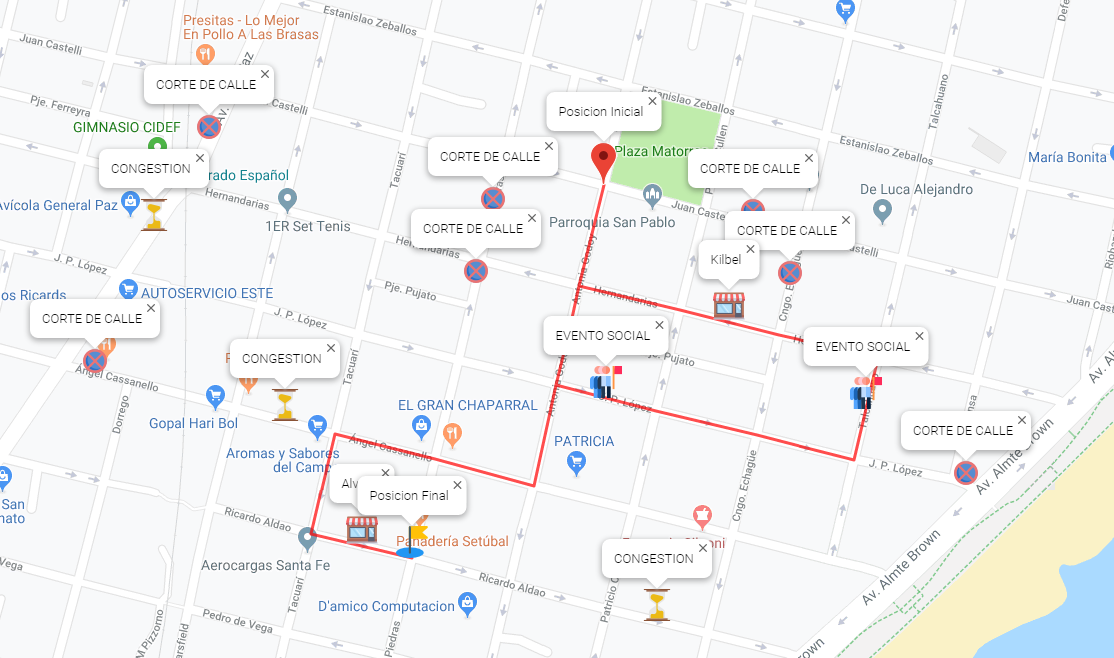


Figura 9 – Mapa visual del estado del agente utilizando estrategia de costo uniforme

## Amplitud

Al implementar esta técnica podemos observar que el agente decide comprar todos los productos en el Alvear.

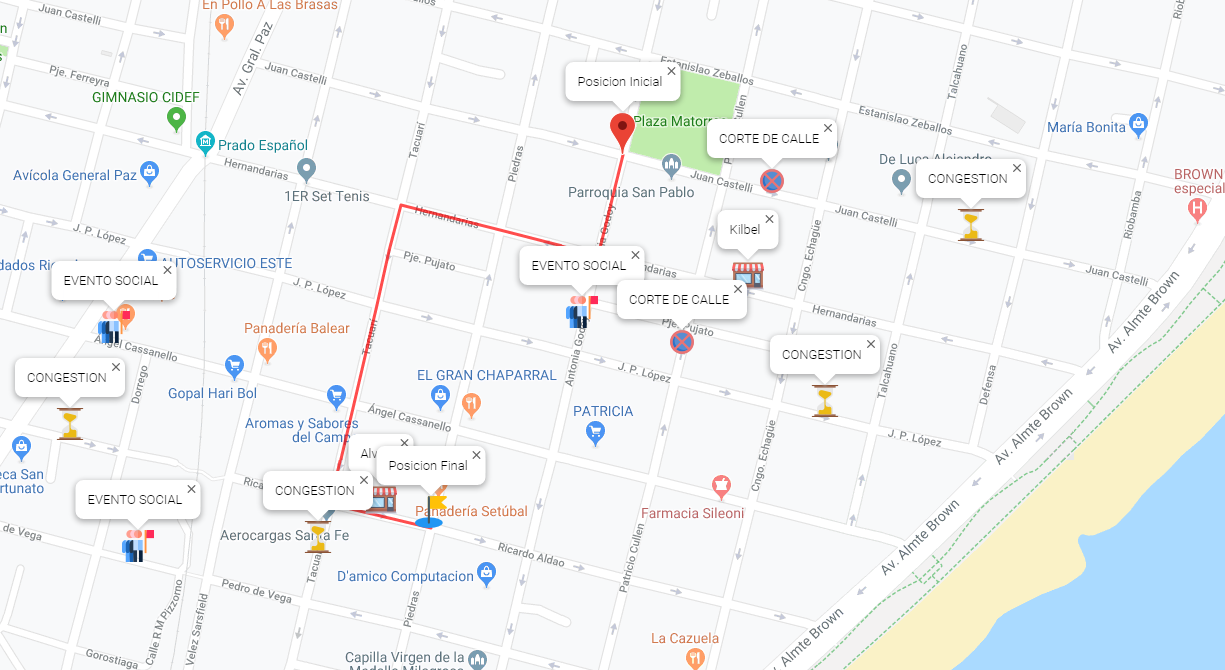


Figura 10 – Mapa visual del estado del agente utilizando estrategia en amplitud

## Profundidad

Al intentar correr esta técnica en el escenario propuesto nos encontramos con problemas de memoria (o eso es al menos lo que nos devuelve como resultado). Teniendo en cuenta que esta estrategia el orden de los operadores, probamos alterar el orden de la creación de los operadores y el resultado es el mismo.

**¿El orden en el que se definen los operadores afecta la solución del problema?**

La estrategia de costo uniforme buscará expandir el nodo que tenga el menor costo asociado, de tal modo que por más que se cambien los operadores siempre se expandirá el mismo nodo.

En la estrategia búsqueda en amplitud, siempre se expandirán todos los nodos hijos que resultaron de una expansión anterior del nodo padre, entonces por más que se modifique el orden de los operadores se seguirán expandiendo todos los nodos que se generen.

Si se habría utilizado una búsqueda en profundidad, distinta seria el resultado ya que para este caso se expanden siempre los nodos que están más a la izquierda en el árbol. Entonces en esta estrategia el orden de los operadores puede cambiar la solución encontrada.

## Estrategias informadas

## A\*

Esta fue la estrategia elegida ya que nos permite hallar la solución óptima a diferencia de Avara.

La heurística que definimos es la cantidad de productos que le faltan comprar al agente:  **agState.getproductosComprar().size();**

Como podemos ver esta no es una heurística buena ya que estaríamos sumando costos de precios de productos y nafta con tamaño de una lista. No corresponden las unidades.

Fue la única heurística que se nos ocurrió ya que no hay forma de saber cuál va a ser el negocio destino debido a que esa es la solución del algoritmo de búsqueda.

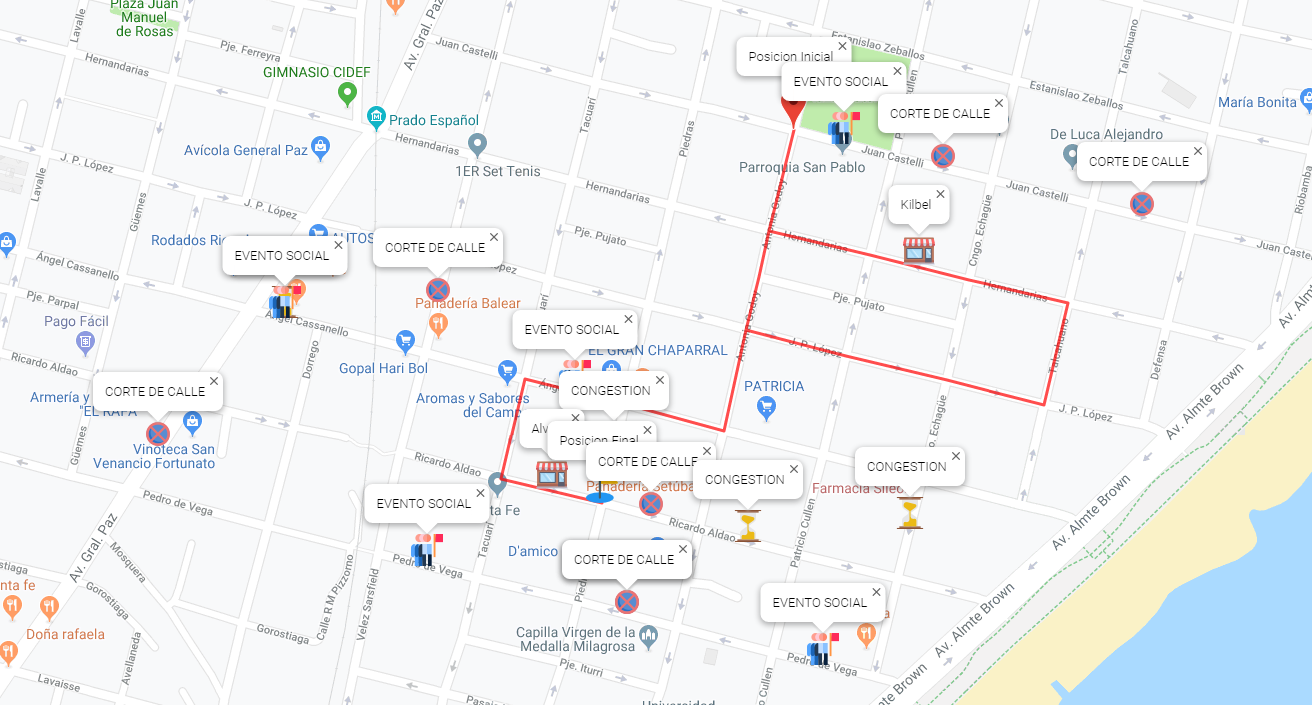


Figura 11 – Mapa visual del estado del agente utilizando estrategia A\*

*Nota sobre los arboles de búsqueda:* se adjuntan los arboles de búsqueda obtenidos de las soluciones mostradas anteriormente. En cada carpeta están todos los arboles generados en cada estrategia, pero representado en imágenes únicamente los últimos tres con su solución.

## Conclusiones

Al trabajar con una heurística cuyos valores radican en la cantidad de productos que le faltan comprar al agente se vuelve insignificante a la hora de evaluarla en una estrategia de A\*. Es decir, la estrategia A\* termina siendo casi equivalente a la estrategia de costo uniforme. Se puede ver las soluciones idénticas en las figuras 9 y 11.

A la hora de definir los costos en los operadores nos encontramos con el problema de que al trabajar con números pequeños (por ejemplo, 0.00454545) se llegaba al error de *memory heap* otorgado por Eclipse. Por lo que se optó, dado ese ejemplo, en trabajar con 0,454545 y de esa forma se puede llegar a una solución. Esto implica tener un valor por cuadra de $45 de nafta en la modalidad Automóvil lo que generó que debiéramos aumentar los costos de los productos para llegar a una solución lógica.

Una de las principales ventajas que encontramos al utilizar grafos fue que la carga de datos de los nodos fue sumamente engorrosa ya hubo que cargar cada esquina del mapa.

Otro problema que tuvimos es que con el mapa completo no funciona ya que se queda sin memoria. Por lo tanto, tuvimos que recortar el mapa a la mitad.

## Referencias

<https://www.teamdev.com/jxmaps#getting-help>

<https://sites.google.com/site/gson/gson-user-guide>

<https://www.objectaid.com/home>