

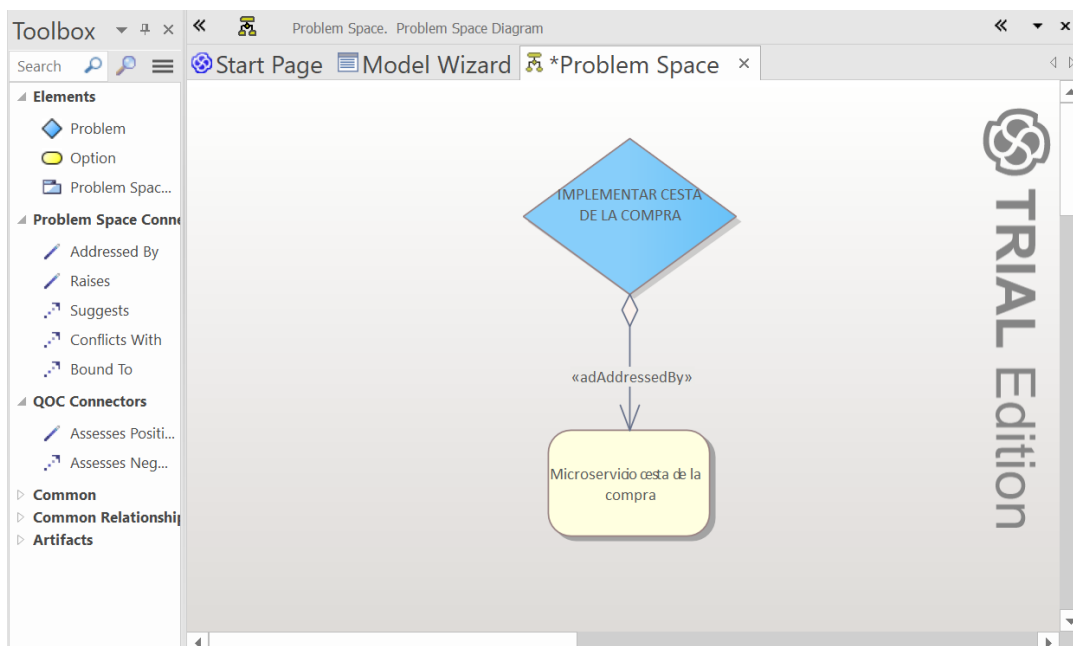
SEMANA 5: ITERACIÓN 4 DE LA ARQUITECTURA (ASJ)

Una vez obtenido todas las decisiones, los Juniors empezamos con la realización de diseño de los diagramas UML en base a las capturas de las decisiones de diseño aceptadas.

En primer lugar, se ha tomado la decisión ADD-13 “Implementación de la cesta como un microservicio y las localizaciones de terceros” mediante la implementación de un nuevo microservicio, correspondiente a la necesidad que se explica en el requisito RF-007 (el sistema debe implementar las funciones de la cesta). Consecuentemente tal decisión engloba una serie de ventajas (se mantiene la modularidad, reutilización de microservicios y escalabilidad) y desventajas (complejidad y probabilidad de exceso de mensajes en el bus de eventos).

```
# Decision de Diseño ADD-13: Implementación de la cesta como un microservicio y las localizaciones de terceros
* Status: proposed
* Deciders: Antonio Agudo Esperanza y Marcos Robles Rodríguez
* Date: 2020-12-08
## Context and Problem Statement
* Se pide implementar una cesta de la compra (como un microservicio) que usará una caché y la localización de microservicios de terceros en MongoDB
## Decision Drivers
* Se nos pide implementar la cesta.
* Se pide que la cesta use una caché
* Queremos que la arquitectura siga lo más modularizada posible
* Se supone que la localización de microservicios va asociada a la cesta.
* Se pide que la localización de microservicios de terceros tenga una base de datos en MongoDB
## Considered Options
* Usar una Caché e implementar en el microservicio la búsqueda de localizaciones de terceros de manera interna
* Crear un microservicio que se encargue de buscar localizaciones de terceros con MongoDB y otro microservicio con una caché interna que implemente las funcionalidades de la cesta
## Decision Outcome
* Chosen option: Implementaremos el nuevo microservicio por separado ya que esto nos permitirá seguir con la modularidad y además, en el caso de que otro microservicio necesite la localización de terceros ya no tendremos que volver a desarrollarlo
### Positive Consequences
* Se mantiene la modularidad.
* Reutilización de microservicios
* Escalabilidad
### Negative Consequences
* Complejidad a la hora de hacerlo.
* Probabilidad de exceso de mensajes en el bus de eventos.
## Pros and Cons of the Options
### Usar una Caché e implementar en el microservicio la búsqueda de localizaciones de terceros de manera interna
* Bueno, ya que el acceso es más rápido
* Bueno, ya que la creación va a ser más rápida
* Malo, ya que la localización no se puede reutilizar
* Malo, ya que no se respeta la modularidad

## Links <!-- optional -->
* [Requisito funcional 007] (https://github.com/Grupo3-DAS/Pr-ctical-Captura-y-Representaci-n-de-Decisiones-de-Dise-o-Equipo-3/blob/main/DAS-P1-Alba\_Sevillano\_Portilla-TAREA1.pdf)
```



Por último, se ha tomado la decisión ADD-14 “Comunicación de los servicios con una API REST”, tras detectar que la aplicación requiere un catálogo de microservicios. Esta decisión de diseño corresponde al requisito RF-002.1 (el sistema debe permitir que los microservicios independientes puedan comunicarse mediante el protocolo REST de forma asíncrona). Consecuentemente le siguen a tal decisión una serie de ventajas (velocidad, independencia de tecnologías/ lenguajes, flexibilidad, escalabilidad y eficiencia) y desventaja (mayor tiempo de desarrollo y no se almacena la sesión)

```
# Decisión de Diseño ADD-14: Comunicación de los servicios con una API REST
* Status: proposed
* Deciders: Antonio Agudo Esperanza y Marcos Robles Rodriguez
* Date: 2020-12-08

## Context and Problem Statement
* El sistema debe permitir que los microservicios independientes puedan comunicarse mediante el protocolo REST de forma asíncrona.
## Decision Drivers
* El cliente sugiere usar una API REST
## Considered Options
* API REST
## Decision Outcome
* Chosen option: API REST, ya que es una manera simple de permitir que se comuniquen servicios externos
### Positive Consequences
* Velocidad
* Independencia de tecnologías / lenguajes
* Flexibilidad
* Escalabilidad
* Eficiencia
### Negative Consequences
* Mayor tiempo de desarrollo al tener que hacer la API
* No se almacena la sesión
## Links <!-- optional -->

* [Requisito funcional 2.1] (https://github.com/Grupo3-DAS/Pr-ctical-Captura-y-Representaci-n-de-Decisiones-de-Dise-o-Equipo-3/blob/main/DAS-P1-Alba\_Sevillano\_Portilla-TAREA1.pdf)
<!-- example: Refined by [ADR-0005] (0005-example.md) -->
```

