

SEMANA 2: ITERACIÓN 1 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-01 “Usar una arquitectura de microservicios para migrar de una arquitectura monolítica” mediante la elección de arquitectura de microservicios, correspondiente a la necesidad de transformación de la arquitectura monolítica (RF-000). Consecuentemente tal decisión engloba una serie de ventajas (cliente satisfecho, mayor modularidad, escalabilidad, versatilidad, agilidad y mantenimiento simple y barato) y desventajas (complejidad de la gestión, no uniformidad, dificultad en la realización de pruebas y alto consumo de memoria).

```
# Decisión de Diseño ADD-01: Usar una arquitectura de microservicios para migrar de una arquitectura monolítica

* Status: Proposed <!-- optional -->
* Deciders: Antonio Agudo Esperanza y Marcos Robles Rodriguez<!-- optional -->
* Date: 2020-11-17 <!-- optional -->

## Context and Problem Statement

Una tienda virtual está basada en una arquitectura monolítica y se desea migrar a una arquitectura de microservicios más flexible y escalable

## Decision Drivers <!-- optional -->

* El cliente nos especifica que se desea migrar a una arquitectura de microservicios.

## Considered Options

* Arquitectura de microservicios

## Decision Outcome

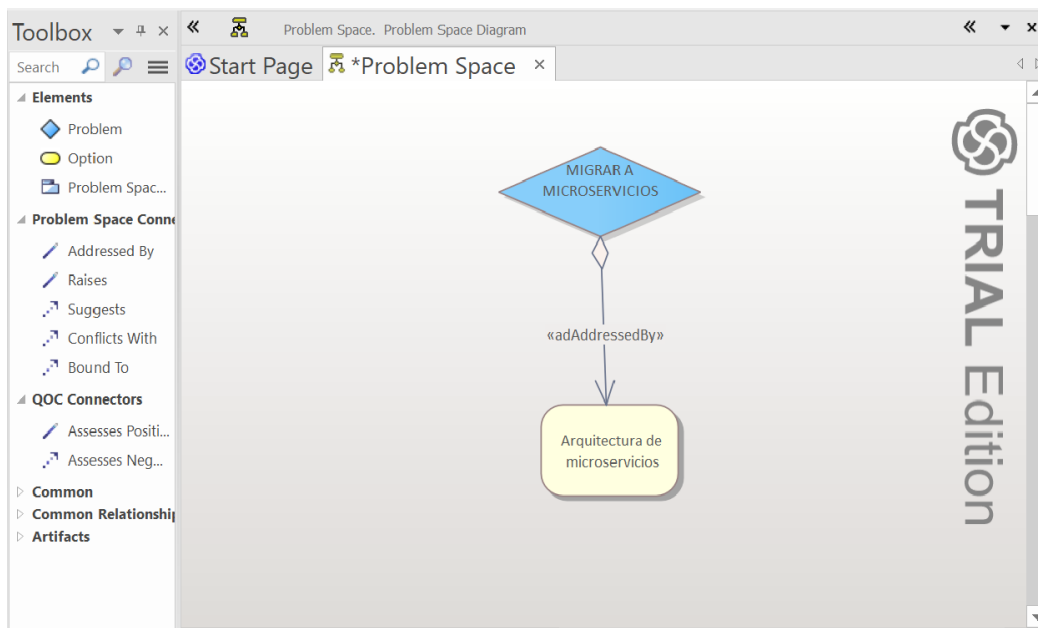
Chosen option: Arquitectura de microservicios porque es la que nuestro cliente desea y creemos que es la que mejor va a satisfacer sus necesidades.

### Positive Consequences <!-- optional -->

* El cliente queda satisfecho
* Modularidad
* Escalabilidad
* Versatilidad
* Mantenimiento simple y barato
* Agilidad

### Negative Consequences <!-- optional -->

* Complejidad de la gestión
* No uniformidad
* Dificultad en la realización de pruebas
* Alto consumo de memoria
```



Adicionalmente, en esta misma iteración, se ha considerado la decisión ADD-02 “Servicio de gestión de solicitudes y respuestas” tras observar el problema de los requisitos funcionales RF-001 (gestión de solicitudes y devolución de respuestas) .

Consecuentemente le siguen a tal decisión una serie de ventajas (mucho control sobre el tráfico, centralización de control, fácil mantenimiento, segura, poco coste de recursos para el cliente y puede funcionar sin cliente) e inconvenientes (congestión del tráfico, poca robustez en caso de fallo comparado con P2P, el software y hardware del servidor darán el coste y si el servidor falla todos los clientes fallarán).

```
# Decisión de Diseño AD-02: Servicio de gestión de solicitudes y respuestas

* Status: proposed
* Deciders: Antonio Agudo y Marcos Robles
* Date: 2020-11-17

## Context and Problem Statement

Comunicación con el cliente por solicitudes y respuestas

## Decision Drivers <!-- optional -->

* El sistema debe comunicarse con los clientes mediante solicitudes y respuestas, por lo que se sugiere que actúen como cliente servidor

## Considered Options

* Cliente-servidor
* Peer to Peer

## Decision Outcome

Chosen option: Cliente-servidor, ya que es la idea principal que se adapta a todas las necesidades cuando se piensa en intercambio de datos.

### Positive Consequences <!-- optional -->

* Mucho Control sobre el tráfico
* Centralización del control
* Fácil mantenimiento
* Segura
* Poco coste de recursos para el cliente
* Puede funcionar sin clientes

### Negative Consequences <!-- optional -->

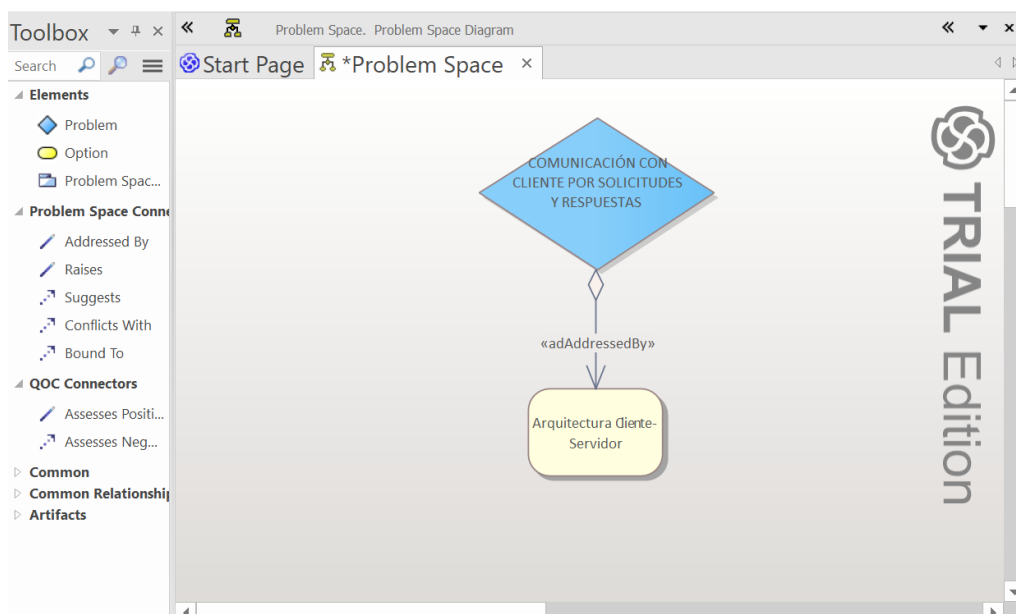
* Congestión del tráfico
* Poca robustez en caso de fallo comparado con P2P
* El software y el hardware del servidor aumentará el costo
* El servidor contiene la aplicación, si este falla todos los clientes fallarán.

## Pros and Cons of the Options <!-- optional -->

### Peer to Peer

Red distribuida de datos

* Bueno, porque, al igual que los microservicios, tiene una gran escalabilidad
* Bueno, porque, descongestiona el tráfico con el servidor
* Bueno, porque aporta mayor robustez
* Bueno, porque tiene una mayor velocidad
* Malo, porque requiere que haya un número mínimo de usuarios para trabajar
* Malo, porque no permite un control centralizado
* Malo, porque la presencia de firewalls o proxies puede dificultar que el usuario use el sistema
* Malo, porque cada microservicio una base de datos puede resultar ineficiente en memoria para el cliente, y puede haber redundancia, o puede darse que se hayan desconectado de la red todos los clientes con ese dato y no esté disponible
```



Por último, se ha tomado la decisión ADD-03 “Interfaz de Usuario”, tras detectar que la aplicación requería de una interfaz para la interacción del cliente con el servicio. Esta decisión de diseño corresponde a los requisitos RF-003 (identificación de usuarios) y RF-006 (interfaz de usuario). Consecuentemente le siguen a tal decisión una serie de ventajas (modularización, información siempre actualizada y las modificaciones en las vistas no afectan al modelo del dominio) y desventaja (mayor dedicación en el desarrollo)

```
# Decisión de Diseño ADD-03: Se debe mostrar una interfaz de usuario
* Status:Proposed
* Deciders:Antonio Agudo y Marcos Robles
* Date:2020/11/18

## Context and Problem Statement
Se usará el patrón MVC para la comunicación con el cliente de los microservicios

## Decision Drivers
* Como se debe mostrar una interfaz al usuario, y esta debe tener los datos del microservicio, hemos decidido utilizar el MVC para que esté ordenado.

## Considered Options
* Patrón modelo-vista-controlador

## Decision Outcome
Chosen option: Hemos elegido el MVC ya que creemos que es una buena manera de tener la parte del diseño y el código de la interfaz bien ordenado.

## Positive Consequences
* Modularización
* Información actualizada siempre
* Las modificaciones en las vistas no afectan al modelo del dominio

## Negative Consequences
* Mayor dedicación en el desarrollo
```

