

## SEMANA 3: ITERACIÓN 2 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-04 “Identificación del usuario” mediante la elección de la clase logger, correspondiente a la necesidad que se explica en el requisito RF-003 (el sistema puede identificar y separar a los distintos usuarios del mismo). Consecuentemente tal decisión engloba una serie de ventajas (rápido tiempo de acceso, funcionamiento básico y ampliable) y desventajas (problemas de seguridad a menos que se añada encriptación adicional, la libre elección de contraseña del usuario, puede suponer un problema de seguridad si no se tramita bien).

```
# Decisión de Diseño ADD-04: Identificación del usuario.
* Status:proposed
* Deciders: Antonio Agudo y Marcos Robles
* Date: 2020/11/24
## Context and Problem Statement
* Dado que de se debe almacenar un registro de los clientes y que estos deben hacer login para acceder a los diferentes microservicios, se necesita dar esta posibilidad al sistema mediante un logger que contará con una BBDD.
## Decision Drivers
* Los usuarios tienen datos propios como nombre o correo
* Todos esos datos deben almacenarse en una BBDD
* El cliente debe mandar la solicitud al servidor con los datos
## Considered Options
* (como expansión de la opción elegida)Usar un OTP (one time password)
* (como expansión de la opción elegida)Usar verificación en 2 pasos
## Decision Outcome
* Chosen option: un logger con una BBDD diseñado como microservicio independiente, ya que consideramos que es una base extensible a la que se le puede añadir otras mejoras de seguridad como OTP o verificación en 2 pasos. A nivel funcional recibirá los datos y devolverá una cookie que se usará para la identidad en el resto de microservicios.

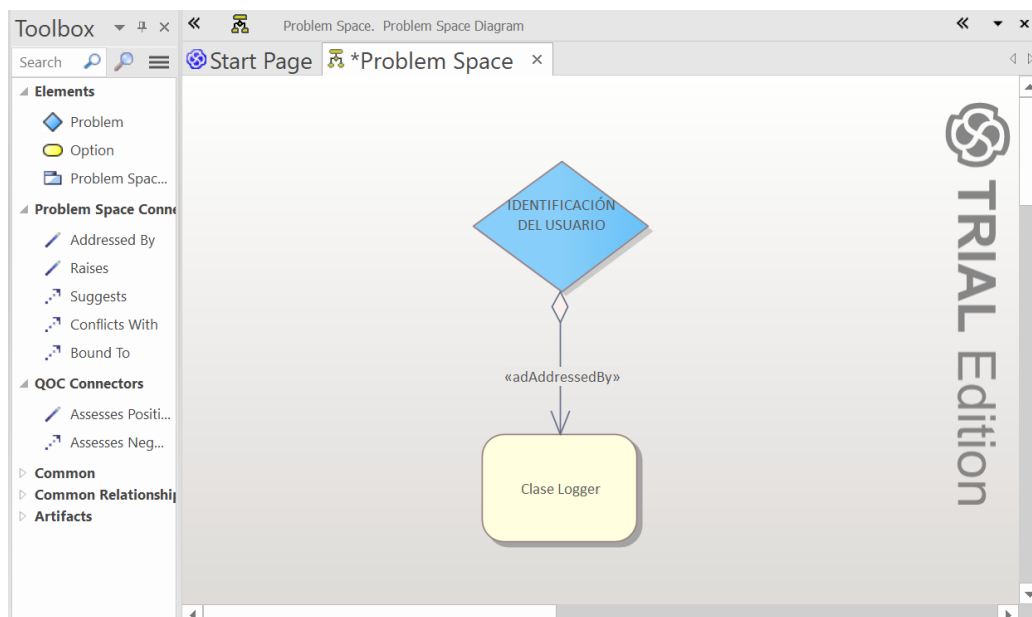
### Positive Consequences
* Rápido tiempo de acceso
* Funcionamiento básico
* Ampliable
### Negative Consequences
* Problemas de seguridad a menos que se añada encriptación adicional
* La libre elección de contraseña del usuario puede suponer un problema de seguridad si no se tramita bien.
## Pros and Cons of the Options
## Usar un OTP (one time password)
https://www.evidian.com/pdf/wp-strongauth-es.pdf

* Bueno, ya que añade otra capa de seguridad
* Malo, ya que si el usuario accede mucho puede resultar molesto

## Usar verificación en 2 pasos
Por ejemplo, usar el sistema de verificación de google en 2 pasos

* Bueno, ya que añade otra capa de seguridad
* Malo, ya que si el usuario accede mucho puede resultar molesto
* Malo, ya que el usuario puede perder el acceso a la cuenta si pierde el dispositivo
* Malo, ya que puede resultar tedioso para el usuario hacer la verificación.

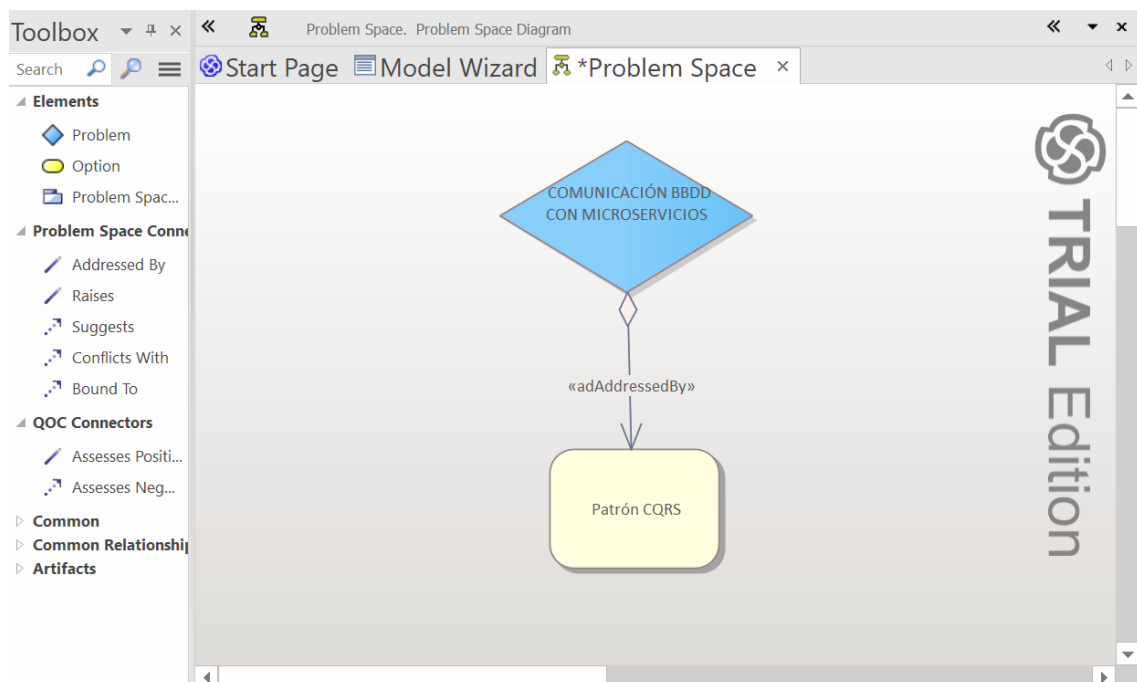
## Links <!-- optional -->
* [Definido por RF003](https://github.com/Grupo3-DAS/Pr-ctical-Captura-y-Representaci-n-de-Decisiones-de-Dise-o-Equipo-3/blob/main/DAS-P1-Alba_Sevillano_Portilla-TAREA1.pdf) <!-- example: Refined by [ADR-0005](0005-example.md) -->
```



Adicionalmente, en esta misma iteración, se ha considerado la decisión ADD-05 “Comunicación de las bases de datos con los microservicios” tras observar el problema que se explica en el requisito funcional RF-005 (el sistema permite acceder a los microservicios a diferentes bases de datos). Consecuentemente le siguen a tal decisión una serie de ventajas (separación de responsabilidades y evitar la congestión en una operación completa si esta está poco cargada) e inconvenientes (complejidad del código y problemas de coherencia).

```
# Decisión de diseño ADD-05: Comunicación de las BBDD de los microservicios.
* Status: proposed
* Deciders: Antonio Agudo Esperanza y Marcos Robles Rodríguez
* Date: 2020-11-24
## Context and Problem Statement
* Se debe comunicar las distintas BBDD con sus respectivos microservicios.
## Decision Drivers
* Se busca dividir los comandos y queries
* El cliente sugiere que usemos CQRS
## Considered Options
* Acceder a las bases de datos mediante un solo canal por microservicio y sin dividir comandos y queries.
## Decision Outcome
* Chosen option: "Comunicación de las BBDD de los microservicios mediante el patrón CQRS, ya que opinamos que dividir las operaciones en dos subgrupos puede ayudar a que haya menos congestión, como por ejemplo, si hay muchas queries y pocos comandos, que los comandos no tengan que esperar a que se descongestionen las queries.
## Positive Consequences
* Separación de responsabilidades
* Evitar la congestión en una operación concreta si esta está poco cargada.
## Negative Consequences
* Complejidad del código
* Coherencia: Si separamos la lectura de la escritura puede darse el caso de que al leer la información de la base de datos, puede estar desactualizada.
## Pros and Cons of the Options
### Mediante acceso directo
* Bueno, ya que es mucho más simple.
* Bueno, ya que siempre se va a leer información actualizada.
* Mala, ya que hay mucha más congestión.

## Links <!-- optional -->
* [Requisito RF005] (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)
* [Información de] (https://martinfowler.com/bliki/CQRS.html) <!-- example: Refined by [ADR-0005] (0005-example.md) -->
```



Por último, se ha tomado la decisión ADD-06 “Usar base de datos para mostrar un catálogo microservicios”, tras detectar que la aplicación requiere un catálogo de microservicios. Esta decisión de diseño corresponde al requisito RF-004 (sistema debe proporcionar un catálogo de microservicios). Consecuentemente le siguen a tal decisión una serie de ventajas (fácil añadir nuevos microservicios, fácil de implementar y eficiente memoria) y desventaja (puede resultar demasiado simple si se desea tener una descripción o campos detallados)

```
# Decisión de Diseño ADD-06: Usar BBDD para mostrar un catálogo de microservicios

* Status: proposed
* Deciders: Antonio Agudo Esperanza y Marcos Robles Rodríguez
* Date: 2020-11-25

Esta decisión se hizo de manera paralela a la ADD-0007

## Context and Problem Statement

* Se requiere la funcionalidad de que haya un catálogo de microservicios.

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

* Se quiere que simplemente sea un catálogo simple. Compuesto, por ejemplo, por una id, el nombre, una descripción corta y su URL.

## Considered Options

* AD007- Patrón composite

## Decision Outcome

* Chosen option: Al final hemos seleccionado esta decisión ya que cumple con el requisito, y la otra opción hacía que los microservicios fueran dependientes del catálogo, por lo que provocaría dependencias innecesarias.

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

* Es fácil añadir nuevos microservicios ya que solo es crear una nueva entrada en la base de datos.
* Es fácil de implementar.
* Es eficiente en memoria.

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

* Puede resultar demasiado simple si se desea tener una descripción o campos detallados.

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

### AD007- Patrón composite

* Buena, agrupación de todos los microservicios en un único catalogo global
* Mala, parentesco con una arquitectura monolítica
* Mala, hace los microservicios dependientes de otro

## Links

* [Requisito RF004] (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)
```

