

# DOCUMENTO DE RIESGOS

Arquitectos cognitivos:

- Jesús Sebastián Tocas Atarama
- Pablo Atahonero García de Blas

Nosotros, como arquitectos cognitivos, tuvimos una reunión con los Arquitectos Senior para debatir y cuestionar las primeras decisiones de diseño que se han tomado.

## **1º. Decisión de Diseño ADD-01: Usar una arquitectura de microservicios para migrar de una arquitectura monolítica**

Esta decisión arquitectónica no lleva requisito, es totalmente independiente.

Dentro de este aspecto, no hay nada que objetar, dado que este tipo de arquitectura es la que ha pedido el cliente, además de que hay que aportar lo más conveniente para él.

## **2º. Decisión de Diseño ADD-02: Servicio de gestión de solicitudes y respuestas.**

Esta decisión arquitectónica está relacionada con el requisito funcional:  
RF001 - Gestión de solicitudes y devolución de respuestas.

Con respecto a esta decisión, los Arquitectos Cognitivos consideramos que, para llevar a cabo esta tarea, se podrían usar el modelo Cliente-Servidor, o el modelo REST, ya que eran dos de las opciones que más se ajustaban a esta adaptación del servicio.

Pensamos en escoger el modelo REST por su fiabilidad, flexibilidad, o su requerimiento de menos recursos por parte del servidor.

Finalmente, la decisión ha sido que vamos a adoptar el modelo Cliente-Servidor ya que se ajusta con más precisión con el aspecto del intercambio de información.

A pesar de escoger esta opción, podrían presentarse alguna complicación ya sea por:

- La congestión de tráfico
- La necesidad de software o hardware específicos
- Su escasa robustez en relación con otros modelos

## **3º. Decisión de Diseño ADD-03: Se debe mostrar una interfaz de usuario**

Esta decisión arquitectónica está relacionada con el requisito funcional:  
RF006 - Mostrar una interfaz de usuario.

En relación con esta decisión de diseño, hemos llegado a un acuerdo con los Arquitectos Senior de que el modelo más adecuado para implementar una interfaz de usuario es el uso del modelo MVC, ya que podemos separar los datos de la vista que conformará la interfaz de usuario.

Aunque podrían surgir algunos inconvenientes para el desarrollo de esta tarea como:

- La separación de conceptos en diferentes capas agregaría más complejidad al sistema.
- El número de archivos a desarrollar y a mantener más tarde se incrementa de manera notable.