

TAREA 1: SEMANA 1: ANÁLISIS DE REQUISITOS (15/11/2020)

REQUISITOS FUNCIONALES:

IDENTIFICADOR (RF-XXX)	NOMBRE	DESCRIPCIÓN
RF001	Gestión de solicitudes y devolución de respuestas	El sistema constará de un servicio de gestión de solicitudes y devoluciones de respuestas.
RF001.1	Procesamiento de pedidos por parte de la app	El sistema tendrá un servicio de procesamiento para procesar los pedidos dentro de la aplicación
RF001.2	Consulta de actualizaciones	El sistema dispondrá de un Gateway con diferentes APIs para la consulta de actualizaciones desde las aplicaciones cliente.
RF001.3	Comunicación de clientes con microservicios	El sistema podrá establecer una comunicación entre los clientes y los microservicios (MVC)
RF002	Canal de mensajería	La lógica de la aplicación deberá contener un canal de mensajería, basado principalmente en agentes de mensajes.
RF002.1	Comunicar los servicios con una API REST y de forma asíncrona	El sistema debe permitir que los microservicios independientes puedan comunicarse mediante el protocolo REST de forma asíncrona.
RF002.1.1	Uso de contenedores de microservicios para orquestar contenedores	El sistema posibilita el uso de estos contenedores dentro de un único host o utilizar un cluster para orquestar contenedores, donde cada uno de ellos se ejecuta en un host diferente
RF002.2	Propagación de actualizaciones entre microservicios	La actualización se realizará gracias a un bus de eventos para propagar las actualizaciones entre microservicios para la integración con aplicaciones externas
RF003	Identificación de usuarios	El sistema puede identificar y separar a los distintos usuarios de este.
RF004	Catálogo de microservicios	El sistema debe proporcionar un catálogo de microservicios
RF005	Acceder a base de datos	El sistema permite acceder a los microservicios a diferentes bases de datos.
RF006	Mostrar una interfaz de usuario	El sistema tendrá la posibilidad de enseñar una interfaz de usuario con la que el cliente pueda interactuar con el servicio.
RF007	Funciones de la cesta	El sistema debe de implementar las funciones de la cesta
RF008	Comunicación entre las BBDD	Las BBDD deben ser capaces de comunicarse entre ellas cuando sea necesario

REQUISITOS DE DOMINIO:

IDENTIFICADOR (RD-XXX)	NOMBRE	DESCRIPCIÓN
RD001	Formato de las respuestas a solicitudes	Las respuestas actuales son en HTML y JSON
RD002	Tipo de la base de datos	Las bases de datos pueden ser SQL o NoSQL
RD003	Funcionalidad de la cesta	La funcionalidad de la cesta usará una caché y la localización de microservicios de terceros en MongoDB
RD004	Comunicación de bases de datos	Las bases de datos se comunican mediante CQRS
RD005	Forma de comunicación de los clientes	Los clientes móviles y de páginas Web se comunican con los puntos de conexión de puerta de enlace de API única.
RD006	Vía de comunicación entre cliente y microservicios	Comunicación de cliente a microservicio de HTTP a través de un Gateway que contiene diversas APIs con el fin de poder consultar las actualizaciones desde las aplicaciones cliente.
RD007	Implementación del bus de eventos	El bus de eventos se puede implementar con cualquier tecnología de infraestructura de agente de mensajería como RabbitMQ, o bien mediante Service Bus de nivel superior (nivel de abstracción) como Azure Service Bus, NServiceBus, MassTransit o Brighter

ANOTACIONES:

- Que la aplicación funcione en varios dispositivos al principio lo consideramos como funcional, pero al final hemos decidido considerarlo como no funcional debido a que hemos optado por entenderlo como atributo de calidad (adaptability).
- Además, también habría que añadir en los requisitos de dominio los datos de cada microservicio, pero como se nos dice que podemos elegirlos nosotros, hemos decidido elegirlos en un puto más avanzado del desarrollo cuando tengamos una idea básica de la arquitectura

TABLA DE TIEMPOS:

Week	Iteration	Time in ADD (ASS)	Reflection Time (ASS-ASC)	Time refined ADD (ASS)	Desing DD Time (ASJ)
1	1	40	31	20	25