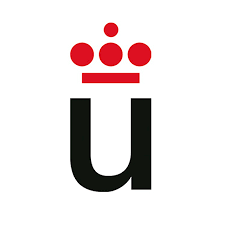
**PRÁCTICA 1**

**Diseño y Arquitecturas del Software**

****

**Correo del responsable:** jd.ortiz.2019@alumnos.urjc

Miembros del equipo:

Andrea Patricia Acuña Padrón

Carlos Fuentes Díaz

Jhostin Davis Ortiz Moreno

Sergio Octavio Mancebo

Mani Víctor Patel Bentz

Juan Luis Rico Rus

# Índice

[**Índice**](#_tm5oudf6zm7m) **2**

[**Roles de los integrantes**](#_n5qdxqmaphan) **3**

[**Tabla de requisitos**](#_ymdcoi7sc2nn) **4**

# Roles de los integrantes

* ASS
  + Jhostin Davis Ortiz Moreno
  + Sergio Octavio Mancebo
* ASJ
  + Mani Víctor Patel Bentz
  + Juan Luis Rico Rus
* ASC
  + Carlos Fuentes Díaz
  + Andrea Patricia Acuña Padrón

Para esta decisión nos hemos basado en las facilidades de cada uno.

De esta forma los arquitectos senior son buenos en la toma de decisiones. Los arquitectos cognitivos son los mejores en debatir decisiones y en pensar en posibles problemas que pueden conllevar realizar una decisión. Por último, los arquitectos junior son los que se desarrollan mejor en UML.

# Tabla de requisitos

| **Identificador** | **Nombre del requisito** | **Descripción del requisito** |
| --- | --- | --- |
| **RF 1** | Sistema escalable | El sistema debe ser tanto flexible como escalable para adaptarse fácilmente a nuevas funcionalidades. |
| **RF 2** | Reingeniería | Aquellos microservicios que soporten el cambio de arquitectura deberán mantenerse. |
| **RF 3** | Sistema de pedidos | Se debe contar con un sistema de pedidos que gestione los pedidos de los usuarios. |
| **RF 4** | Sistema de mensajería | Se debe contar con un sistema de mensajería para la comunicación. |
| **RF 5** | Sistema de devoluciones | Se debe contar con un sistema que permita realizar devoluciones de las compras previas. |
| **RF 6** | Sistema de preferencias | Se deben gestionar las preferencias de compra para una personalización más concreta. |
| **RF 7** | Sistemas de pago | Se dispone de un microservicio que permita conectar a sistemas de pago. |
| **RF 8** | Sistema de seguridad compras | Todas las compras se deben realizar de forma segura. Para ello, se dispondrá de un sistema de seguridad que garantice la seguridad en las compras. |
| **RF 9** | Restricción intentos de compra | Se deben restringir los intentos de compra a los usuarios para evitar la saturación de los microservicios correspondientes. |
| **RF 10** | Interfaz responsive. | El sistema debe contener una interfaz adaptable a distintas plataformas (Tableta y Móvil). |
| **RF 11** | Lógica de acceso a la base de datos | Permite acceder a las bases de datos. |
| **RF 12** | Sistema de acceso a MongoDB | Permite acceder a los datos de la base de datos de MongoDB y obtener las localizaciones de los microservicios. |
| **RF 13** | Bus de eventos lógicos. | El sistema debe gestionar la coherencia entre las bases de datos de los distintos microservicios. |
| **RF 14** | Bases de datos SQL y NoSQL | Se usan para modelar y acceder a los datos de compra y de preferencia respectivamente. |
| **RF 15** | Lenguajes de programación Java y .NET | Se usa Java aplicado a la orientación a objetos para la lógica del sistema y .NET para gestionar el bus lógico de datos de todos los microservicios. |
| **RF 16** | Protocolo REST | Los microservicios que soporten la reingeniería y migración deben usar el protocolo REST para conectar varios sistemas basados en el protocolo HTTP. |
| **RF 17** | Comunicaciones vía HTTP | Todas las comunicaciones se realizan vía HTTP, evitando así errores con la transferencia de datos. Para esto, los clientes se comunican con los microservicios mediante un componente gateway. |
| **RF 18** | Componente Gateway | Debe monitorizar el estado de los microservicios. |
| **RF 19** | NServiceBus | Se debe usar un bus dedicado de eventos lógicos como NServiceBus para que exista coherencia entre las distintas bases de datos de los distintos microservicios. De esta forma, la arquitectura NserviceBus permite el intercambio de mensajes entre servicios de manera desacoplada proporcionando, además, fiabilidad y manejo de errores. |
| **RF 20** | AMQP para sistema de mensajería. | Se debe usar el protocolo AMQP para el intercambio de datos e información entre los diferentes sistemas. |

Se han recogido los requisitos como funcionales puesto que todas las funcionalidades se desarrollan en el sistema. Sin embargo, se podrían considerar como requisitos de dominio aquellos que indagan más en el hardware o el uso de estándares y software (e.g. AMQP o el HTTP).