**Microservices Conceptual Questions**

1. Around 5 Microservices (not including API Gateway or Discovery)
   1. File System developed to handle the docs of 3 Apps
   2. Rest Api to solve the data transfer between Med Labs and Facilities
   3. App module to manage the prescriptions of patients
   4. Management module for patients information
   5. Management module of patients therapies

Few more microservices were developed in my but those are the ones I worked from scratch.

1. I always have thinked that independence and modularity is really important. I think in microservice as a well design machine to work with independently parts. One piece of the machine depends on the other one, up to certain point, but the absence of that dependent part, will not break down the entire machine.

Efficient way to work and a real divide and conquer architecture. Each team work independently, make their own changes, updates and push to production without break down other work. Better performance and stability for the systems.

1. Not going in any logic order, but the discovery of new microservices (using Eureka is my exp) is essential part in the architecture and design. The resiliency patterns to handle fault tolerance in general and the Api gateway for the right distribution and balance of the requests.
2. DevOps Tools: Jenkins, Docker, Kubernetes and Terraform
3. Choosing Spring Boot I create an Api to process and transfer of HL7 files between laboratories and and facilities. An HL7 is a standardized protocol for exchanging data between apps used for various health cares. Even when there is an standard, existe bunch of changes in the files that need to be acquired and processed. My solution, provide a better way to process all the differents files and successfully saved the information into the database. Thousands of HL7 file are interchanging between a Facility daily, just to order test of urine and blood and more than a half were lost or missing before my solution. The system took a few months(almost 2 after 1st version) to gain the maturity and rise around 98% of daily file processing without lost.
4. With more than 5 apps already develop, deployed and currently been used for many users, I continue creating all my new projects using start.spring.io. Avoid the necessity of package the app and copy the .war inside a JBoss to be deployed, is one things that I’m more than ok with. The new way of handling the dependencies using the BOM is a life changing and not going back, the auto configuration using annotation and how to handle files for the system are the things, from the top of my head, that make me really comfortable and like to continue work with Spring Boot.