A development process of enterprise applications with microservices

F H Vera-Rivera¹,²,³
¹ Grupo de Investigación GIDIS, Universidad Francisco de Paula Santander, San José de Cúcuta, Colombia
² Grupo de Investigación GEDI, Universidad del Valle, Cali, Colombia
³ Foundation of Researches in Science and Technology of Materials, Colombia

E-mail: fredyhumbertovera@ufps.edu.co, fredy.vera@correounivalle.edu.co

Abstract: This paper presents a development process of enterprises applications with microservices. We identify the construction phases, practices, fundamentals, methods and tools that are being used in the development of microservices based enterprise applications. After identifying and proposing the development process, it is validated and tested in the construction of an application called Sinplafut. By using microservices the size of enterprise applications is reduced and easier to manage, deployment, scaling, test and replace using the resources offered by cloud computing. In this way the costs and efforts necessary to maintain the application in the cloud are considerably reduced.

1. Introduction

According to Lewis and Fowler: “The microservice architecture describes a particular way of designing software applications as suites of independently deployable services”. Microservices have these characteristics: 1- organization around business capability, 2- automated deployment, 3- intelligence in the endpoints, and 4- decentralized control of languages and data [1]. Microservices are a new trend in software architecture that is strongly influenced by distributed computing, the main idea is to divide an application into small services; each service must be as independent as possible from the others. Each one is executed in separate processes, in a single and isolated container, it is possible to assign more computational resources to the microservice that needs it most, unlike the monolithic applications that allocate resources to the entire application, which are sometimes resources that may not be used and fully exploited [2].

Microservices are highly modular distributed systems, reusable through an API exposed through the network. This implies that microservices inherit the advantages and disadvantages of distributed systems and web services. Yarygina and Bagge point out that microservices are an implementation approach of Service-Oriented Architecture (SOA) and SOA is a subclass of distributed system [3]. Also, Zimmerman, Pautasso, James Lewis and others indicate that microservices comprise an organic implementation approach for SOA (just as Scrum is one, but not the only way to practice agile development). Common features include business orientation, polyglot programming in multiple paradigms and languages, and design for failure; decentralization and automation are specifically emphasized in the microservices implementation approach [4,5].

Quality attributes are fundamental and propose challenges for today's enterprise applications. Availability, performance, automatic scaling, automated testing, continuous integration and
deployment, security, fault tolerance, are essential features that every business application must handle. Microservices reduce the complexity of handling these features.

This research work is given in advance of the doctoral thesis that is being developed at the “Universidad del Valle” entitled "Agile model of specification of the architecture of applications based on microservices". A development process of enterprise applications using microservices is proposed. With this process we want to identify the construction phases, practices, fundamentals, methods and tools that are being used in the development of enterprise applications. After identifying and proposing the development process, it is validated and tested in the construction of an application called Sinplafut.

This paper is organized as follows. In Section 2 the methods are presented. In Section 3, the results are discussed; this work presents a development process of enterprise applications with microservices. Finally, in Section 4 we summarize the conclusion of this work.

2. Methods
The methodology of this work is based on "Design Science Research" following the approaches proposed by Hevner, March, Park, & Ram for scientific research in the field of information systems and development of technological products. The paradigm of design science seeks to extend the limits of human and organizational capacities through the creation of new and innovative artefacts [6]. In the present investigation the artefact to be created is the agile development process to develop enterprise applications with microservices. The development process identifies phases, processes, tools and fundamental activities that are carried out to build applications that use microservices. The Figure 1 represents the research model.

![Research Model](image)

**Figure 1.** Research model.

This work begins with the theoretical and practical foundation in microservices, then a systematic literature review of the state of the art was done. With all this, the development process is proposed and defined; finally, the development process is validated and tested in the construction of a use case and a set of microservices.

3. Results and discussion
This section presents the results obtained in this research work, first we present the process of developing applications that use microservices and then we present Sinplafut a business web application based on microservices.

3.1. Development process of enterprise applications with microservices
In the process of developing applications based on microservices, the phases, processes, tools and activities fundamental in the construction of applications based on the architecture of microservices were identified. Once the current state of development based on microservices is characterized, it is intended to propose improvements to the process using agile practices and highlighting good practices. Figure 2 shows the features of the process. The process is divided into two fundamental parts, first the development of each microservice and second, the development of applications that use those microservices.

The phases for development each microservice are:
3.1.1. **Microservice development.** Each microservice is developed independently by a separated development team, including a different programming language and a different database.

3.1.2. **Continuous integration.** Each team synchronizes and integrates daily the microservice code that is being developed.

3.1.3. **Automated tests.** Once the code was integrated, the repository executed a series of tests (unit, functionality, load, others) to guarantee the quality of the code implemented.

3.1.4. **Continuous deployment.** If the tests are satisfied, it is automatically deployed to the test or production server.

3.1.5. **Monitoring and updating.** Each microservice is monitored independently, so that the failures or needs of scaling or increase of computational resources can be identified. The update is done independently, being able to change a microservice without affecting its clients.

**Figure 2.** Main features of the development process of application with microservice [7].

The phases for development of microservices based application are:

3.1.6. **Identification and discovery.** Allow microservice customers to find and establish communication.

3.1.7. **Integration and composition.** Identified the microservices that meet the requirements of the application, we proceed to integrate and compose the enterprise application. In some cases, it is necessary to adapt some microservice to meet the requirements of the application or build a new one from scratch.
3.1.8. Deploy an enterprise application. Armed the application, it’s deployed and monitored using the DevOps practices of the same form as in the construction of any microservice. Therefore, the implemented application is fault tolerant, scalable, adaptable and distributed.

DevOps is an emerging paradigm that allows the development team to be integrated with the operations staff, hence its acronym Dev (Developers) - Ops (Operations). Their practices allow quick and frequent releases [8]. DevOps is a new term that emerges with the coalition of two new trends, agile infrastructure or agile operations, and collaboration between development and operations personnel throughout all stages of the development life cycle from coding to deployment to production [9].

The identification and discovery of services are a fundamental part in the development of applications that use the microservices that were created and deployed beforehand. This mechanism allows cataloguing, describing, specifying and organizing microservices so that they can be used. The problem with the discovery of services is to allow service consumers to locate service providers in real time to facilitate communication [10].

3.2. Sinplafut App, an enterprise application with microservices

Sinplafut is an information system to improve and speed up the planning of soccer training using information and communication technologies and modern sports training techniques. With Sinplafut you can manage your profile, each player's card, physiological information and injuries. Create and configure your macro-cycles, month-cycles, micro-cycles and training sessions, using pre-established methods and tests, managed and controlled by Sinplafut. Data records of competencies, analyses and controls [11].

From the user story or requirement, the microservices that should be implemented were identified. The Table 1 shows the relations between microservices and requirements.

| User story                                      | Microservice                          |
|------------------------------------------------|---------------------------------------|
| 1. Sport club management                       | EquipoApp                             |
| 2. Management of club teams                    |                                       |
| 3. Management of Team’s players                |                                       |
| 4. Management of coaching staff                |                                       |
| 5. Management of the training plan             |                                       |
| 6. Management of the month-cycles              |                                       |
| 7. Management of the micro-cycles              | PlanEntrenamientoApp                  |
| 8. Management of training sessions             |                                       |
| 9. Management of sports training methods       |                                       |
| 10. Generate the schedule of training sessions  |                                       |
| 11. Web site of Sinplafut, Front-End           | SinplafutApp                          |
| 12. Sports test management                     | TestDeportivosApp                     |
| 13. User management and security               | SeguridadApp                          |

Table 1. Microservices – requirements relations.

To analyses the dependencies between the microservices, you can analyze the existing relationships in the data model of each of them and from there determine how dependent they are. As much as possible, the aim is to reduce the number of dependencies. The Figure 3 shows the relations between “EquipoApp” and “PlanEntrenamientoApp”. Training plans are made for teams or can be for a specific player, hence the relationship of the two microservices.

This division is carried out following the approaches proposed by Evans in domain driven design. Domain-Driven design is an approach to complex software development in which: 1- It focuses on the main domain. 2- Models are explored in a creative collaboration of domain professionals and software professionals. 3- A ubiquitous language is spoken within an explicitly limited context [12].

The definition of the size or granularity of the microservice is a challenge of open research and in continuous evolution. Microservices are small, autonomous services that work together. They are
small and focused on doing something well done. They follow the “Simple Responsibility Principle” that says "Gather the things that change for the same reason, and separate those things that change for different reasons” [13].

![Diagram of microservices dependencies](image)

**Figure 3.** Dependences between microservices of Sinplafut. We can identify the dependencies between microservices from the entity relationship diagram.

The Figure 4 shows the dependences diagram of microservices of Sinplafut. In this diagram you can see how the application is divided into a set of microservices and its relationships and dependencies can be analyzed. A low coupling and high cohesion between each microservice must always be done.

![Diagram of microservices](image)

**Figure 4.** Microservices dependences diagram of Sinplafut. A microservice is an independent application.

Finally, Figure 5 shows the implementation architecture of each microservice. In this case, they were implemented in Django and Rest framework. Each microservice was implemented, tested and deployed using the devops practices and a deployment pipeline in bitbucked [14].

By using microservices the size of applications is reduced, divided into smaller and easier to manage, deployment, scaling, test and replace using the resources offered by cloud computing. In this way the costs and efforts necessary to maintain the application in the cloud are considerably reduced. The architectural style of microservices changes the way applications are built, tested, deployed and maintained. Microservices facilitate the migration of applications to infrastructure in the cloud, allowing automatic scaling, load balancing and fault tolerance. By using microservices you can deploy
a large application as a set of small applications (microservices) that can be developed, implemented, expanded, managed and monitored independently. Agility, cost reduction and granular scalability bring some challenges and the complexity of managing distributed systems [15].

Figure 5. The implementation architecture of each microservice.

4. Conclusions
This research has gathered from different resources with novelties and current information technologies that have a very good projection in the future, such as service orientation and specifically the microservices. Currently in our environment a small number of investigations have been carried out in this area, it is very important for the University and system engineering schooling, to have the knowledge in these technologies in order to transmit the experience that arises as a result of this research to students, future engineers and the community in general. The result of this research may also be used for the benefit of the University, to improve the characteristics of enterprise applications that support its academic, administrative and extended processes.

The trends, good practices and the way to develop microservices and the applications that use them were characterized and identified, so that in the short time a transfer of knowledge can be made to the companies that develop software in the region.

It is noteworthy that the architecture of microservices is starting to be used in other areas and obtaining important benefits such as the Internet of Things, Autonomous vehicles, telecommunications, Systems of health and others. In many of the reviewed articles these benefits are evidenced especially in topics such as scaling, tolerance to failures and deployment at the level of devices and sensors in the cloud using containers and microservices.

References
[1] Lewis J and Fowler M 2014 Microservices Consulted on: https://martinfowler.com/articles/microservices.html
[2] Thönes J 2015 Microservices IEEE Software 32(1) 113 – 116
[3] Yarygina T and Bagge A H 2018 Overcoming security challenges in microservice architectures Proc. IEEE Symp. on Service-Oriented System Engineering–SOSE (Bamberg, Germany: IEEE) pp 11–20
[4] Pautasso C, Zimmermann O, Amundsen M, Lewis J, Josuttis N 2017 Microservices in practice, part 1: Reality check and service design IEEE Software 34(1) 91–98
[5] Zimmermann O 2017 Microservices tenets: Agile approach to service development and deployment. Comput. Sci. - Res. Dev. 32(3–4) 301–310
[6] Hevner A R, March S T, Park J, Ram S 2004 Design science in information systems research MIS Quarterly 28(1) 75–105
[7] Vera-Rivera F. H 2018 Método de automatización del despliegue continuo en la nube para la
implementación de microservicios Proc. XXI Conf. Iberoamericana de Ingeniería del Software CIBSE (Bogotá, Colombia: Universidad de los Andes)

[8] Wettinger J, Breitenbucher U, Leymann F, 2014 Standards-based DevOps automation and integration using TOSCA Proc. 2014 IEEE/ACM 7th Int. Conf. Util. Cloud Computing (UCC) (Washington: IEEE) pp 59–68

[9] Mueller E, Wickett J, Gaekwad K, Karayanev P 2017 What Is DevOps? The agile admin Consulted on: https://theagileadmin.com/what-is-devops/

[10] Stubbs J, Moreira W, Dooley R 2015 Distributed systems of microservices using Docker and Serfnode 7th International Workshop on Science Gateways (Budapest, Hungary: IEEE) pp 34–39

[11] Vera-Rivera F H and Vera-Rivera J L 2018 Sinplafut©-sistema de información para la planificación del entrenamiento en el fútbol Consulted on: http://sinplafut.com:8030/

[12] Evans E 2004 Domain-driven design (Boston: Addison Wesley) pp 20–22

[13] Newman S 2015 Building microservices (California: O’Reilly Media, Inc.) pp 16–18

[14] Atlassian Inc. Bitbucket 2017 Bitbucket pipelines-continuous delivery|bitbucket Consulted on: https://bitbucket.org/product/features/pipelines

[15] Villamizar M, Garcés O, Castro H, Verano M, Salamanca L, Gil S, 2015 Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud evaluando el patrón de arquitectura monolítica y de micro servicios para desplegar aplicaciones en la nube 10th. Conf. Colombiana de Computación (Colombia: Sociedad Colombiana de Computación and IEEE) pp 583–590