Article **Journal Industrial Engineering**
December 2021, Vol.5 No.15 30-33

Microservices architecture as a viable option to support the organic growth of PYMEs

La arquitectura de microservicios como una opción viable para apoyar el crecimiento orgánico de las PYMEs

BENÍTEZ-QUECHA, Claribel†*, ALTAMIRANO-CABRERA, Marisol, LÓPEZ-GUZMÁN, Oscar Eduardo and MÉNDEZ-LÓPEZ, Minerva Donají

Tecnológico Nacional de México Campus Oaxaca, Mexico.

ID 1st Author: *Claribel, Benítez-Quecha* / **ORC ID:** 0000-0001-6516-5760, **CVU CONACYT ID:** 657582

ID 1st Co-author: *Marisol, Altamirano-Cabrera* / **ORC ID:** 0000-0001-5800-9655, **CVU CONACYT ID:** 657390

ID 2nd Co-author: *Oscar Eduardo, López-Guzmán* / **ORC ID:** 0000-0003-2427-4850

ID 3rd Co-author: *Minerva Donají, Méndez-López* / **ORC ID:** 0000-0001-5336-1772

DOI: 10.35429/JIE.2021.15.5.30.33 Received July 25, 2021; Accepted December 30, 2021

Abstract

Generally, computer developments for business activities are focused on systematizing data processing. Activities that as the company grows or changes, they also undergo changes.However, computer systems are not easy to modify when they are done through traditional methodologies. The microservices architecture is characterized by being a modular development, specifically it is divided into independent services that communicate with each other through APIs. These services run independently and autonomously, so if an PYME changes its needs, only new services are added that can interact with the existing ones. Therefore, the ability to respond to changes is increased. Objective: General: Implementation of a Computational System based on Microservices for the management of a PYME. Specific: Identification of horizontal and vertical scalability needs, Identification of technologies and services. Development and testing of services, Coupling and testing.Methodology used: SCRUM is designed for projects with a high level of uncertainty. In this project, it was not known at the outset which technologies should be implemented.Contribution: Serve as a guide in the identification of viable technologies to implement microservices, focused on scalability in PYMEs.

PYME, microservices, scalability

Resumen

Generalmente los desarrollos computacionales para actividades empresariales están enfocados a sistematizar procesamiento de datos. Actividades que conforme la empresa crece o se modifica, éstas también sufren cambios. Sin embargo, los sistemas computacionales no son fáciles de modificar cuando se realizan a través de metodologías tradicionales. La arquitectura de microservicios se caracteriza por ser un desarrollo modular, específicamente se divide en servicios independientes que se comunican entre sí mediante API. Estos servicios se ejecutan de forma independiente y autónoma, por lo que, si una PYME cambia sus necesidades, sólo se agregan nuevos servicios que pueden interactuar con los ya existentes. Por lo que se incrementa la capacidad de respuesta a los cambios. Objetivo: General: Implementación de un Sistema Computacional basado en Microservicios para la gestión de una PYME. Específicos: Identificación de necesidades de escalabilidad horizontal y vertical, Identificación de tecnologías y servicios. Desarrollo y pruebas de servicios, Acoplamiento y pruebas. Metodología usada: SCRUM está diseñado para proyectos de alto nivel de incertidumbre. En este proyecto, no se sabía de inicio qué tecnologías debían implementarse. Contribución: Servir de guía en la identificación de las tecnologías viables para implementar microservicios, enfocados a la escalabilidad en PYMEs.

PYME, microservicios, escalabilidad

Citation: BENÍTEZ-QUECHA, Claribel, ALTAMIRANO-CABRERA, Marisol, LÓPEZ-GUZMÁN, Oscar Eduardo and MÉNDEZ-LÓPEZ, Minerva Donají. Microservices architecture as a viable option to support the organic growth of PYMEs. Journal Industrial Engineering. 2021. 5-15:30-33.

* Author's Correspondence (E-mail: claribelbk@hotmail.com)
† Researcher contributing as first author.

Introduction

In recent years, when a scalability problem is addressed in software development, two main pillars have been erected, the first being the integration into cloud computing platforms and secondly the selection of clean architectures, such as hexagonal and based on microservices. Although microservices architectures are scalable in nature, it must be considered that said scalability responds to a solution that is growing in the solution and not necessarily the growth of the company that implements it. When a Computer System is developed based on the current needs of the company or in the best of cases, with some future considerations, it happens that the company modifies its operations and these may not be the ones considered in the future. This implies that the Computer System is no longer viable and must be modified, which results in the original developer being the one who modifies it or having exhaustive documentation for a new developer to update it, so that always ends up doing a new system. As a proposed solution, it is considered that the fact of segmenting the Computational System based on the services required, presents the possibility of carrying out the scaling required by the SME, over time. Since each service is independent and autonomous with respect to the others, and new services can be coupled as required without having to rewrite the entire code.

In this work, each of the development phases and the technology used are exposed, according to tests based on response times, security and coupling capacity.

Theoretical framework

We can define electronic commerce that is carried out through the internet as the exchange of commercial information using this data transmission network as a medium in order to acquire goods or services. Microservices architecture is an approach to developing software solutions by dividing it into different modules or low-coupling services, each of them executing a process independent of the rest. Cloud computing is a scheme for the provision of computing resources through a distributed computing system that are dynamically supplied to the consumer.

A continuous integration and deployment chain refers to the process of automating operational tasks when building and deploying software. CI (Continuous Integration) covers the operations of building applications from source code while CD (Continuous Deployment) involves those operations referring to the start-up of the built software.

Development

11 microservices were developed grouped into 2 modules (Providers and clients) using the Java Spring Boot framework that consist of a secure session handler (Provider / Client security), an event handler for each module (Events Provider / Client), administrators of business information (Business Management), of the client (Clients management), and of products and services (Products / Services management), a manager of the purchase and sale process (Purchase / Sale) a point of consumption of product information and services (Products / services) and finally a microservice for registering customer, supplier and business information (Onboarding). The selection of Spring Boot is based on the robustness of the framework and the ease with which unit and integration tests can be automated, a key step in the development of services that must maintain a low coupling. For information management, MongoDB was integrated in order to store and retrieve the information of the actors. The main use of a non-relational database responds to specific needs of the solution, such as the ability to consult coordinate information in real time, and to perform searches based on parameters. The nature of a non-relational database helps to improve the experience in these specific processes that are the main part of the solution. The session management was implemented using Redis, this database engine not only allows a more advanced administration of the sessions, but also acts as a general information dashboard in which all the microservices can obtain information in common, becoming a channel more efficient communication than an event queue for certain actions.

ISSN 2523-0344

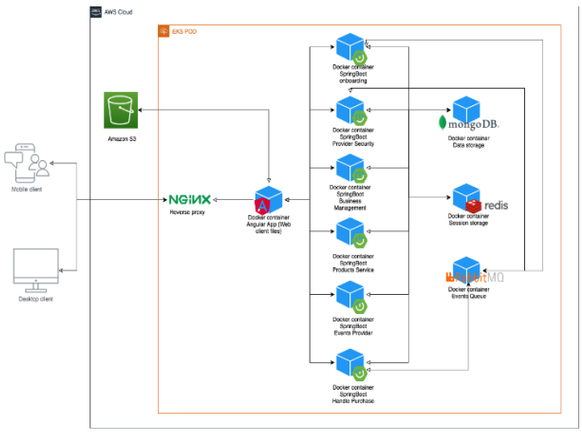

Figure 1 Microservices architecture for providers module
Source: Own elaboration

RabbitMQ was also implemented to manage event queues in communication between microservices, and with the outside world. When an architecture of this style is designed, it is necessary to implement different communication channels where it is possible to report or consult events that are of common interest for one or more microservices. 2 Angular Web clients were also developed for consumption from the end devices, both modules were designed with easy integration with microservices in mind, allowing to enable and disable functionalities in a simple way.

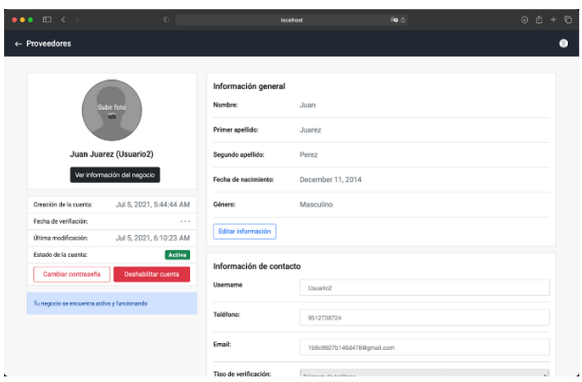

Figure 2 Suppliers module, profile editing functionality
Source: Own elaboration

Angular was selected as the frontend development framework because it allows more robust and less casual development. Although the decision of frontend technologies are not usually influenced by the architecture of the solution, they are influenced by the size of the project, and when an architecture is designed to improve scalability, it is useful to have technologies that allow this scalability in a organic, especially when it comes to SMEs, since they benefit from being able to invest their resources in maintaining an infrastructure that corresponds to their volume of sales or traffic.

As more traffic is generated, the cost of maintenance will increase as well. For the deployment of all the elements, it was necessary to use docker containers and in this way facilitate the maintenance of the microservices, as well as Kubernetes to improve monitoring and ensure their response.

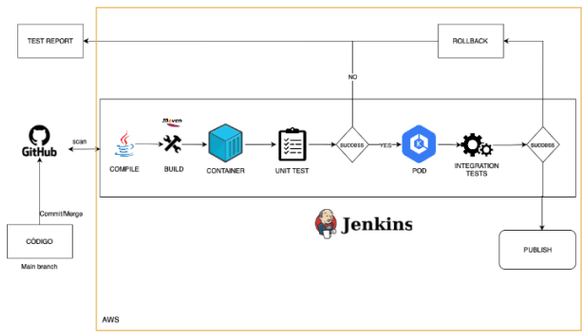

Figure 3 Continuous integration and deployment scheme (CI / CD)
Source: Own elaboration

Finally, by designing a continuous integration and deployment model using the Jenkins tool, it was possible to automate all the tasks corresponding to the packaging, dockerization and deployment of the microservices on the AWS cloud, this integration and deployment process also contributes to the scalability of the solution, as it allows making changes more efficiently.

Choosing a cloud computing service provider like AWS is perhaps the most obvious step an SMB can take to deploy a software project. These types of providers offer payment schemes based on traffic, that is, the use that the solution is having, in this way the company will not have to make a large investment in hosting until this solution has high traffic.

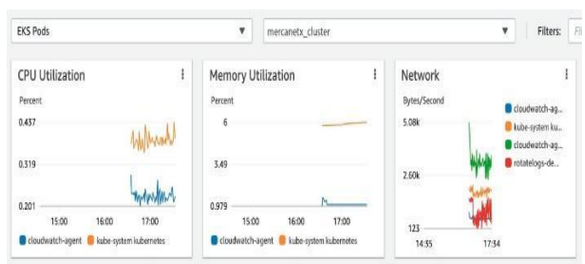

Figure 4 Kubernetes pod monitoring on AWS
Source: Generated with the Amazon Cloudwatch tool

ISSN 2523-0344

All of the above was proposed as an initial development, since within the general design of the solution, 2 additional modules (administrators and dispatchers) are contemplated whose implementation would require 7 microservices and 2 additional Web clients.

Results

In addition to all the advantages that a non-monolithic infrastructure brings with it, such as the low level of coupling between the functionalities, for the SME, having a development where the architecture is completely modularized also represents a possibility of implementing small parts of the solution and freeing up functionalities progressively as your infrastructure as a company and economic condition can allow it.

Gratitude

We thank the Tecnológico Nacional de México Campus Oaxaca for its support in carrying out this work.

Conclusions

Current technologies respond satisfactorily in computational development through architectures based on microservices. With regard to database managers, programming languages and container platforms. However, not all of them do, it is necessary to carry out tests of response times, security in the transactions and coupling. Factors such as resources, time and personnel influenced the design procedure of the following architecture, as well as the selection of the technologies to be used within it. For the development of MercanetXpress (Registration in process), a solution for an SME dedicated to connecting product and service providers with customers and distributors, vertical and horizontal scalability needs were identified for both the solution (platform) and the company, as that part of his business plan consisted of implementing the solution in certain geographical locations and replicating it following the organic growth of the company.

References

Alfaro Azofeifqa, Cindy. Alfaro Chamberlain, José I. Monge González, Ricardo. ICTs in Central American SMEs: Impact of the Adoption of Information and Communication Technologies on the performance of companies. Ed. Tecnológica de Costa Rica. 2005.

Arias, Á. (2015). Cloud Computing: 2nd Edition. IT Campus Academy. Berenguel Gómez, José Luis. Development of distributed web applications. Ed. Editions Paraninfo. 2016. Married, Sergio. How to take your backend in Node to the next level? Introduction to NestJS. Retrieved from https://www.paradigmadigital.com/dev/introduccion-nestjs / on May 7, 2021.

Dadoo, Maya. The need to implement technology in an SME. Retrieved from https://expansion.mx/opinion/2018/10/03/opinion-la-necesidad-de-implementar-tecnologia-en-una-pyme, on February 10, 2021.

Llopis, Joan. Lighten the Monolith: Miniservices and Microservices. Retrieved from https://clouddistrict.com/aligerar-monolitominiservicios-microservicios/ on May 11, 2021.

López, D., & Maya, E. (2017). Software Architecture based on Microservices for Web Application Development.

Ortega Candel, José Manuel. Technologies for microservices-based architectures: Patterns and solutions for applications deployed in containers. Ed. José Manuel Ortega. June 30, 2020.

Ruiz, A. D. L. Á. R. (2015). Analysis and perspectives of electronic commerce in Mexico. Profiles of Social Sciences, 3 (5).

ISSN 2523-0344