



Universitatea Tehnică a Moldovei

**Metode de implementare a microserviciilor
utilizând tehnologia .NET Core**

**Methods of implementing microservices using
.NET Core technology.**

Masterand:

Costiuc Irina

Conducător:


conf.univ., dr. Ciorbă Dumitru

Chișinău 2019

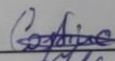
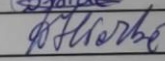
Ministerul Educației, Culturii și Cercetării al Republicii Moldova
Universitatea Tehnică a Moldovei
FACULTATEA Calculatoare, Informatică și Microelectronică
Departamentul Ingineria Software și Automatică

Admis la susținere

Șef de departament: conf. univ., dr. Dumitru Ciorbă


„12” decembrie 2018

**Metode de implementare a microserviciilor utilizând
tehnologia .NET Core**
Teză de master în
Tehnologii Informaționale

Masterand:  (Costiuc Irina)
Conducător:  (Ciorbă Dumitru)

Chișinău 2019

ADNOTARE

Dacă să analizăm istoria dezvoltării tehnologiilor începînd de la crearea primelor mașini de calcul și pînă în prezent, vom ajunge la concluzia că majoritatea arhitecturilor la baza cărora au fost create aceste tehnologii, erau soluții ce rezolvau dezavantajele sistemelor precedente. Ca de exemplu stilul monolit, a ajuns să fie popular datorită faptului că acesta a fost creat în perioada cînd creștea necesitatea de a utiliza tot mai mult mașinile de calcul. Stilul monolitic oferea stabilitate, integritate și securitate produsului, pentru că realizarea lui se producea pe o mașină de calcul. Însă pe lîngă faptul că creșteau în cantitate mașinile de calcul, se mărea și numărul de utilizatori a unui sistem, ce în final ruina procesul acestuia.

La fel și apariția microserviciilor, a fost un trend ce a devenit soluție optimă pentru arhitectura sistemelor complexe. În această lucrare au fost acumulate posibilitățile oferite de microservicii. S-a realizat o comparație a sistemelor bazate pe stilul monolit și pe microservicii. De asemenea a fost formulată concluzia, de ce nu trebuie de lăsat la o parte existența stilului monolit. Mai mult ca atît, s-au determinat avantajele și dezavantajele între aceste două stiluri, s-au stabilit motive după care trebuie să fim conduși pentru a realiza un sistem bazat pe microservicii.

De asemenea au fost accentuate modele de implementare a microserviciilor. Se regăsește descrierea celor mai reușite exemple de utilizare, dar și momente dezavantajoase a microserviciilor. S-au stabilit care sunt principiile de proiectare a microserviciilor, și de ce ele sunt importante în arhitectura sistemelor. De asemenea, au fost determinate relațiile între microservicii, pentru a avea o viziune corectă în structura logică a sistemului. În final, s-a explicat sistemul, dezvoltat pe baza principului bounded context și construit pe relații message bus. De asemenea, sunt descrise tehnologiile utilizate, în special s-a marcat importanța utilizării bazei de date MongoDB și principiul containzării utilizînd Docker.

ANNOTATION

If we look at the history of technology development from the creation of first computing machines to the present, we will come to the conclusion that most of the architectures on which these technologies were created were solutions that solved disadvantages of previous systems. For example, the monolithic style became popular because it was created in period when the need to use computing machines increased. This style offered product stability, integrity and security, because it was produced on only one computing machine. But besides the fact that the computing machines were growing in quantity, also was increased the number of users of a system, which eventually overloaded the process.

Similarity it was with microservices, it became a mainstream that now is the optimal solution for complex systems. In this thesis were accumulated all the possibilities that are offered by microservices. We'll find a comparison between systems based on monolith style and microservices. Also, here it was concluded, why should not monolithic style be left out. Moreover, there have been determined the advantages and disadvantages of these two styles, and were identified reasons that will help to create systems based on microservice.

Also, were underlined the microservices implementation models. There were described examples of microservices in use, as well as the disadvantages of it. Here was determined what are the principles of microservice pattern, and why it is important in the system architecture. Also was established the correct model of communication between microservices for good implementations of the logical structure in system. Finally, was developed the system based on bounded context and built on the message bus communication. There are described technologies used in system, especially the importance of using the MongoDB database and the principle of containerization by using Docker.

Cuprins

LISTA FIGURILOR8

INTRODUCERE	9
1. MICROSERVICII CA O NOUĂ TENDINȚĂ ÎN IT	10
1.1 Ideia generală despre microservicii	10
1.2 Aplicații monolite si problemele lor	10
1.3 Motive pentru implementare a microserviciilor	13
1.4 Avantaje si dezavantaje microserviciilor la nivelul dezvoltării	15
2. PRINCIPIILE DE PROIECTARE A MICROSERVICIILOR	21
2.1 Dimensiunea microserviciilor	21
2.2 Principiile DDD	22
2.3 Conteinizare și orchestrare	24
2.4 Posibilitățile microserviciilor în cadrul bazelor de date	25
2.5 Integrarea continuă	26
2.6 Tipuri de conexiuni între microservicii	26
3. ARHITECTURA PROGRAMULUI	32
3.1 Nucleul microserviciului	32
3.2 Web API	35
3.3 Open API Swager	35
3.4 Smart Proxy API Gateway	37
3.5 Conteinizarea utilizînd Docker	38
3.6 MongoDB	39
CONCLUZIE	41
BIBLIOGRAFIE	42
ANEXA 1	44
ANEXA 2	46
ANEXA 3	47
ANEXA 4	49
ANEXA 5	51