

Relatório Final do Estágio Supervisionado Modelos de Microsserviços e Monolítico num Ambiente de Desenvolvimento

Gustavo B. Guimarães¹

¹Universidade Federal de Jataí (UFJ) - Jataí, GO - Brasil

{borges.gustavo}@discente.ufj.edu.br

Abstract. *This supervised internship final report was developed from the observation of a systems development team at the Federal University of Jataí. It aims to show what was proposed as a development activity during the internship, bringing a discussion on the theme of the Microservices architectural model, in contrast to the other opposite model, the Monolith, thus exploring its main points and differences. In order to draw a parallel with what was observed in the field during the internship, thus, concluding with a study on the model used by the development team, the motivations behind it and starting from the theoretical framework an analysis of the situation.*

Resumo. *Este relatório final de estágio supervisionado, foi desenvolvido a partir da observação de uma equipe de desenvolvimento de sistemas na Universidade Federal de Jataí. Ele tem o objetivo de mostrar o que foi proposto de atividade de desenvolvimento durante o estágio, trazendo uma discussão sobre o tema do modelo arquitetural de Microsserviços, em contraste com o outro modelo oposto o Monolítico, assim explorando seus principais pontos e diferenças. De forma a traçar um paralelo com o que foi observado em campo durante o estágio, assim, concluindo com um estudo relativo ao modelo utilizado pela equipe de desenvolvimentos, as motivações por trás disso e partido do referencial teórico uma análise da situação.*

1. Introdução

Os Microsserviços vem ganhando um grande destaque atualmente, sendo adotados por diversas empresas e grupos de desenvolvimento de software,[Lucio et al. 2017]. Isso se deriva de sua capacidade de modularização, reaproveitamento e produtividade, pontos que são prezados por toda equipe de desenvolvimento que buscam a criação de aplicações complexas, escaláveis, com boa performance de forma ágil.

Porém, a aplicação desse modelo não é nada simples, trazendo uma carga alta de complexidade e necessidade de adaptação muito forte por parte de toda a equipe, que caso aplicado de forma errada, pode prejudicar em muitos níveis o desenvolvimento da equipe. Sendo assim, uma análise do contexto da equipe de desenvolvimento e projetos a serem desenvolvidos é crucial.

Nesse sentido, este relatório aborda o estágio supervisionado de dentro de uma equipe de desenvolvimento, da qual foi analisado e traçado um paralelo com o referencial teórico relacionados ao tópico dos modelos arquiteturais de Microsserviço e Monolítico,

suas diferenças, vantagens e desvantagens, e sendo feita uma análise da equipe e seu desenvolvimento atual.

Sendo o objetivo deste relatório, trazer o material teórico estudado referente a atividade e tema definidos em conjunto com o professor orientador, a observação do local de trabalho e atuação da equipe, e uma análise conclusiva da atual situação como um todo, trazendo como ponto principal a discussão de modelo arquitetural de Microsserviços.

2. Arquitetura de Microsserviços e Monolítica

Os microsserviços são uma abordagem arquitetônica e organizacional da área de desenvolvimento de softwares(aplicações). E como o autor [da Conceição and Pinto 2021] bem coloca, os microsserviços são uma forma de divisão de uma aplicação em um conjunto de pequenos e leves serviços, os quais são completamente independentes e com um alto grau de acoplamento, de modo que, eles se comunicam trabalham dinamicamente em conjunto em prol da realização de uma tarefa.

O pilar central dessa arquitetura é a modularização, que proporciona um alto grau de autonomia e especialização para cada serviços atuante. Dessa forma, caso ocorra um erro em um componente, uma correção seria feita de maneira pontual em tal componente, diferentemente de uma aplicação de arquitetura monolítica no qual não só a correção seria no todo, mas também os efeitos do erro, como aponta [de Sousa Castro et al. 2021].

Nesse sentido o autor [da Conceição and Pinto 2021] levanta os principais pontos da boa utilização dos microsserviços no desenvolvimento de uma aplicação, mostrando mutias de suas vantagens:

- **Integração:** O desenvolvimento de microsserviços depende de equipes muito menores e independentes, facilitando e agilizando a atuação destas equipes, principalmente na tomada de decisões, fora em eventuais atualizações de códigos ou implementação de novos componentes.
- **Reutilização:** Como os serviços desenvolvidos são completamente independentes, eles podem ser facilmente reaproveitados em outros projetos, evitando o desperdício de recursos e acelerando imensamente o desenvolvimento num todo.
- **Flexibilidade:** No mesmo sentido de independência dos serviços, e o fato de eles se comunicarem via API, a possibilidade de utilização de diversas tecnologias e ferramentas para cada serviços independentemente, faz com que as equipes tenham uma liberdade e flexibilidade enorme durante o desenvolvimento.
- **Escalabilidade:** Cada componente é escalável de forma individual, e a eventual adição e alteração de componentes não impacta o resto do sistema, facilitando e proporcionando aos projetos desenvolvidos uma escalabilidade incrível.
- **Resiliência:** Como apontado previamente, qualquer tipo de eventual erro, tem um impacto muito menor por afetar os serviços individuais, além de que, o isolamento do problema e recuperação do funcionamento normal torna-se também algo muito mais fácil.
- **Performance:** O proveito de computação paralela, graças a modularidade e independência dos serviços, facilita a otimização de utilização dos núcleos de processamento.
- **Produtividade:** Levando em conta a independência e modularização, a atuação de equipes menores, divididas para o desenvolvimento paralelo de serviços individuais, facilita e acelera a atuação da equipe num todo.

A arquitetura de Microsserviços é contrastada por outra arquitetura mais simplista chamada de Monolítica. Que contrariamente ao que vimos do modelo Microsserviços, é desenvolvido numa massa de funcionalidades única, sendo um modelo unificado composto por apenas uma peça, sendo seus componentes interconectados e interdependentes de modo que o sistema é único e projetado para ser independente. Assim, qualquer alteração feita em algum componente impacta todo o sistema, como aponta [da Conceição and Pinto 2021].

Nesse sentido, toda a proposta de uma aplicação monolítica é sua atuação inteiriça, independente e única. Logo, todos os benefícios de modularização e autonomia das partes independentes do modelo de Microsserviços é perdido se optado por seguir o modelo Monolítico, porém como destaca [Garcia and Pagani], ele trás algumas vantagens que devem ser levadas em conta:

- **Simplicidade de Arquitetura:** Como já foi comentado, ele é um modelo mais simples, sem muitas camadas ou nuances, tornando seu desenvolvimento muito mais tranquilo e simples.
- **Fluxo de Publicação Simples:** Por se tratar de um sistema inteiriço, ele vai ser alterado, compilado e publicado de forma única e inteira, sem ter que levar em considerações múltiplos módulos.
- **Tecnologia:** Um ponto contrastante coma alta flexibilidade do Microsserviço. Já que o projeto num todo deve ser trabalho pela equipe com uma mesma tecnologia e ferramentas, criando uma cordialidade e coesão maior, claro a troco da flexibilidade.

Consequentemente, ambos modelos tem seus pontos positivos e atrativos, porém além disso é necessário levar em conta a aplicabilidade, consequências e as desvantagens de aplicação deles. E dessa forma, o [Garcia and Pagani] mostra alguns lados negativos de ambos os modelos, sendo os do Microsserviço:

- **Estrutura e Cultura:** Deve ser levado em conta a cultura da empresa para aplicação do modelo de Microsserviços, já que é necessário uma série de fatores estruturais, configurações e adaptações.
- **Segurança:** Em contraponto à boa resiliência, é de extrema importância ter segurança em toda a comunicação entre os módulos, que dependem de diversos pontos de comunicação para conseguir a atuação conjunta e acoplamento deles. Algo que não ocorre da mesma forma em aplicações monolíticas.
- **Tecnologia:** Da mesma forma que a flexibilidade e independência possibilitar a utilização de tecnologias diferentes é algo bom, pode ser algo desastroso se não for muito bem documentado, e a equipe num todo ter conhecimento multidisciplinar, conseguindo trabalhar de forma coesa com diversas tecnologias e ferramentas num mesmo projeto.
- **Complexidade:** Uma das principais dificuldades e características do Microsserviços, é justamente a carga de complexidade que esse modelo trás para o desenvolvimento em todas suas etapas, por se tratar de múltiplas camadas, diversos serviços e possivelmente bancos de dados, de modo que, é preciso um monitoramento, realização de testes e suporte em tudo isso, o que exige bastante dos profissionais envolvidos.

Assim vemos que muitas vantagens do modelo de Microserviços podem ser desvantagens caso não seja corretamente aplicado, além de, precisar de uma capacitação maior dos profissionais envolvidos em diversas etapas, com um crescimento significativo na complexidade das aplicações a serem desenvolvidas.

E alguns problemas chave do modelo Monolítico são:

- **Build:** Por se tratar de um sistema único, qualquer tipo de alteração requer uma compilação total da aplicação, o que piora de acordo com o tamanho da aplicação.
- **Ponto de Falha:** Como já foi mencionado, qualquer tipo de erro ou falha pode facilmente comprometer o sistema como um todo, sendo assim, qualquer tipo de problema pode ser muito crítico.
- **Acultramento:** Se tratando de novos integrantes, para conseguir uma adequação ao sistema e todo seu código, pode ser algo dificultoso, novamente, piorando mais conforme o tamanho do sistema.

É possível notar, que quanto maior o sistema mais evidente os problemas se tornam, fazendo com o que antes era fácil e simples, difícil e complexo. O que compromete o desenvolvimento de grandes aplicações com diversas funcionalidades.

3. Observação

O estágio foi realizado dentro da própria Universidade Federal de Jataí, dentro de um departamento de desenvolvimento da SETI, o qual foi recentemente formado e contratado alguns terceirizados para dar início as atividades do setor no âmbito de desenvolvimento de sistemas. Sendo que, a ideia principal para esse setor seria num futuro próximo, a universidade por meio dele teria uma maior independência com seus sistemas, principalmente o SIGAA, do qual precisa de manutenção paga da universidade de Goiânia. Então, essa nova equipe está sendo treinada para a criação e desenvolvimento de novos sistemas, mas também para a autonomia e manutenção relativos a outros sistemas mais antigos.

Durante a atuação em campo no estágio, a equipe total estava dividida em duas equipes menores, sendo que cada uma estava trabalhando em um projeto diferente. Nesse sentido, será retratado mais sobre um dos projetos de uma sub-equipe, do qual se tratava em desenvolver uma aplicação que, recebendo um arquivo CSV, faria validações e formatação de acordo com a documentação, exportando de volta esse arquivo no formato TXT para o usuário. Esse "conversor CSV" deveria ser futuramente utilizado por alguém para realizar conversões de arquivos que seriam alimentados num banco de dados.

O desenvolvimento desse projeto se deu por 3 atuantes terceirizados e um concursado mais experiente que os estava auxiliando com esse desenvolvimento. A linguagem utilizada foi o Java, com o Amazon Correto 11, posteriormente utilizando do Java Spring para criação da API nos padrões REST, que ligaria o back-end com o front-end, que foram feitos separados, sendo que o front-end foi feito em TypeScript utilizando React.

O projeto se deu início com todos trabalhando no back-end, até a conclusão, então a construção da API que alimentaria o front-end, entrão finalmente o front-end. Durante todo o processo foram estudados diversos cursos, video aulas e materiais para auxiliar no aprendizado e entendimento. E se tratando de uma nova equipe com atuantes inexperientes na área, o desenvolvimento seguiu devagar juntamente com estudos para justamente integrar os novos contratados.

Por se tratar de um primeiro projeto, e falta de conhecimento e experiência, não tiveram muitas documentações ou padrões sendo utilizados, além de algumas reuniões e estudos específicos, todos estavam atuando sem muitas referências teóricas ou qualquer tipo de arquétipo.

Dito isso, e partindo do referencial teórico levantado relacionados aos modelos de arquitetura, feita uma análise desse projeto e da atuação de desenvolvimento num geral da equipe, o modelo sendo utilizado claramente é o Monolítico. Já que se tratavam de aplicações inteiriças, com módulos completamente interconectados e interdependentes, sendo que durante o desenvolvimento foi possível presenciar momentos nos quais houveram erros que quebraram o sistema num todo, ou, durante alguma alteração sempre foi necessário compilar o sistema por inteiro, entre outras características descritas no que foi apresentado como material de estudo dos modelos de Microserviços e Monolíticos.

4. Conclusão

Fica claro com a análise feita do modelo de Microserviços e do modelo Monolítico, que por mais que o modelo de Microserviços é um modelo mais novo, sendo amplamente adotado, com seus merecidos méritos, o modelo Monolítico não seja necessariamente ruim ou pior, já que foi bem destacado as vantagens e desvantagens de ambos os modelos. E existem sim muitos cenários que a adoção do modelo Monolítico é bem mais vantajoso do que o de Microserviços.

Com isso em mente e feita a análise do ambiente no qual o estágio foi realizado, dando destaque ao projeto do conversor CSV que está sendo desenvolvido pela equipe, é evidente que a equipe de desenvolvimento da SETI, que foi recém formada, ainda está num período de adaptação e evolução inicial. Como retratado acima, os primeiros projetos somente agora estão encaminhando para a fase final, e a equipe ainda tem bastante tempo dedicado a estudo e capacitação dos membros, buscando num momento inicial não apenas resultados, mas a consolidação dessa equipe para que num futuro próximo ela esteja preparada para assumir projetos muito maiores que contam com uma expertise e domínio que ainda falta em algumas formas para a equipe no atual momento.

Dessa forma, feita uma análise do desenvolvimento atual da equipe, o modelo adotado é claramente o Monolítico, o que baseado na pesquisa teórica realizada, realmente condiz ser a melhor opção no momento, com uma equipe ainda crescendo, projetos iniciais não muito extensos, e o desenvolvimento da própria equipe ainda em progresso, buscar a utilização de um modelo complexo como o de Microserviços seria um fator de complicação desnecessária neste atual momento.

Entretanto, isso é algo que obviamente deve ser buscado no futuro, e um preparo da equipe para isso deve ser iniciado o quanto antes possível. Porque além de novas tecnologias, para a produção e desenvolvimento futuro que a equipe busca, uma melhor estruturação, coordenação, documentação e padronização são extremamente necessários, e a adoção do modelo de Microserviços é indispensável para qualquer equipe de desenvolvimentos séria atualmente. Principalmente se tratando de projetos grandes como alguns como os que a equipe da SETI espera estar desenvolvendo num futuro próximo.

Referências

- da Conceição, M. T. and Pinto, G. S. (2021). Arquitetura de microsserviços. *Revista Interface Tecnológica*, 18(2):53–64.
- de Sousa Castro, T. F., de Melo Vale, F. G., and de Sousa, F. H. F. (2021). Microsserviços/microservices. *Brazilian Journal of Development*, 7(3):21829–21833.
- Garcia, G. B. and Pagani, C. Do monolítico ao microserviço.
- Lucio, J. P. D. et al. (2017). Análise comparativa entre arquitetura monolítica e de microsserviços.