**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Arquitetura de *Software* Distribuído**

Projeto Integrado

Relatório Técnico

Micro Serviços - Gerenciamento de Locação de Automóveis

Caio Gomes

Lucas Tondo Sendeski

Curitiba, Paraná

Dezembro 2021.

# Projeto Integrado – Arquitetura de Software Distribuído

**Sumário**

[Projeto Integrado – Arquitetura de Software Distribuído 2](#_heading=h.3dy6vkm)

[1.](#_heading=h.z337ya) Introdução 3

[2.](#_heading=h.3j2qqm3) Cronograma do Trabalho 5

[3.](#_heading=h.1y810tw) Especificação Arquitetural da solução 6

[3.1](#_heading=h.4i7ojhp) Restrições Arquiteturais 6

[3.2](#_heading=h.e4qgsirqegbo) Requisitos Funcionais 6

[3.3](#_heading=h.2xcytpi) Requisitos Não-funcionais 7

[3.4](#_heading=h.1ci93xb) Mecanismos Arquiteturais 7

[4.](#_heading=h.3whwml4) Modelagem Arquitetural 8

[4.1](#_heading=h.2bn6wsx) Diagrama de Contexto 8

[4.2](#_heading=h.qsh70q) Diagrama de Container 9

[4.3](#_heading=h.3as4poj) Diagrama de Componentes 9

[5.](#_heading=h.1pxezwc) Prova de Conceito (PoC) 11

[5.1](#_heading=h.49x2ik5) Integrações entre Componentes 11

[5.2](#_heading=h.2p2csry) Código da Aplicação 11

[6.](#_heading=h.147n2zr) Avaliação da Arquitetura (ATAM) 13

[6.1.](#_heading=h.3o7alnk) Análise das abordagens arquiteturais 13

[6.2.](#_heading=h.ihv636)  Cenários 13

[6.3.](#_heading=h.32hioqz) Evidências da Avaliação 14

[6.4. Resultados Obtidos 15](#_heading=h.1hmsyys)

[7.](#_heading=h.41mghml) Avaliação Crítica dos Resultados 16

[8.](#_heading=h.2grqrue) Conclusão 17

[Referências 18](#_heading=h.vx1227)

## Introdução

Existem diversas maneiras de desenvolver softwares, desde técnicas simples até técnicas sofisticadas. O desenvolvimento de software está em constante evolução e as técnicas abrangentes para desenvolver também evoluem.

Não existe certo ou errado ao desenvolver um sistema, entretanto é possível aplicar algumas técnicas de desenvolvimento para extrair mais performance, usabilidade, escalabilidade do objetivo proposto.

Alguns sistemas acabam sendo desenvolvidos como monolito (“Obra construída em uma só pedra”) e enfrenta algumas dificuldades por ter essa arquitetura de projeto já enraizada.

Utilizar micro serviços na arquitetura de desenvolvimento traz alguns benefícios que auxiliam a equipe de desenvolvimento envolvida no projeto. É possível entregar mais em menos tempo com a facilidade e a separação do projeto em pequenos blocos o que proporciona melhor manutenibilidade.

As complexidades de utilizar micro serviços são poucas e sua arquitetura pode ser acoplada a *design-pattern* distintos com objetivo de melhorar ainda mais a entrega do software.

O objetivo deste trabalho é apresentar o funcionamento de um micro serviço e explanar suas vantagens e desvantagens para o mercado de tecnologia.

Os objetivos específicos propostos são:

* Realizar um estudo sobre os micros serviços, levantar quando foi implementado os primeiros micros serviços e quais a principais tecnologias utilizadas;
* Desenvolver um sistema simples de aluguel de veículos utilizando o conceito de micro serviço em uma linguagem *open-source* com uma documentação visual;

## Cronograma do Trabalho

A seguir é apresentado o cronograma proposto para as etapas deste trabalho.

| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| --- | --- | --- | --- |
| **De** | **Até** |
| 01 / 01 / 22 | 10 / 02 / 22 | 1. Desenvolvimento do relatório técnico |  |
| 15 / 02 / 22 | 15 / 02 / 22 | 1. Apresentação do relatório técnico |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 1. Desenvolvimento da apresentação visual do projeto. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 1. Apresentação do visual do projeto |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 1. Desenvolvimento do projeto. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 1. Apresentação do projeto |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 7. |  |

## Especificação Arquitetural da solução

O projeto será feito com três micro serviços:

* Serviço de controle de usuários.
* Serviço de controle de veículos.
* Serviço de controle de aluguel de veículos.

Ambos os micros serviços serão desenvolvidos com o *design-pattern* CQRS(*Command Query Responsibility Segregation*) com o objetivo de separar a responsabilidade de escrita e leitura dos dados do projeto.

Diagrama do projeto:



*Figura 1*

## Restrições Arquiteturais

A arquitetura de micro serviço possui algumas restrições para não deixar de ser um micro serviço.

O micro serviço não pode ter diversas responsabilidades ou ser um projeto grande, tirando o objetivo de ser pequeno e objetivo.

R1: O micro serviço não deve ter muitas funcionalidades.

R2: O micro serviço não deve ter muitas responsabilidades

R3: O sistema será apenas para plataforma WEB

## Requisitos Funcionais

Cada micro serviço terá seu funcionamento apartado.

| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| --- | --- | --- | --- |
| RF01 | O sistema deve permitir a inserção de novos usuários | M | A |
| RF02 | O sistema deve permitir a edição dos usuários existentes | B | M |
| RF03 | O sistema deve permitir a exclusão lógica dos usuários | M | M |
| RF04 | O sistema deve permitir a inserção de novos veículos | M | A |
| RF05 | O sistema deve permitir a edição dos veículos existentes | B | M |
| RF06 | O sistema deve permitir a exclusão lógica dos veículos existentes | B | M |
| RF07 | O sistema deve permitir a inserção do aluguel de veículos | M | M |
| RF08 | O sistema deve permitir a edição do aluguel de veículos | B | B |
| RF09 | O sistema deve permitir a exclusão lógica do aluguel de veículos | M | M |
| RF10 | O sistema deve permitir consultar quais veículos estão alocados | M | M |
| RF11 | O sistema deve validar se o veículo está precisando de manutenção preventiva | A | A |
| RF12 | O sistema deve validar se o usuário logado tem permissão para aprovar a locação de algum veículo | M | A |
| RF13 | O sistema deve notificar ao administrador que existe uma solicitação de locação. | M | A |
| RF14 | O sistema deve gerar relatórios dos veículos que estão sendo mais utilizados. | B | M |
| RF15 | O sistema deve validar um veículo que já está alocado e não permitir a locação do mesmo. | B | M |
| RF16 | O sistema deve exibir apenas os menus de acordo com as permissões do usuário logado | B |  |
| RF17 | O sistema deve permitir o usuário logar na aplicação | B | A |
| RF18 | O sistema deve permitir o administrador bloquear acesso de um usuário | B | A |
| RF19 | O sistema deve permitir a alteração das permissões do usuário | M | M |
| RF20 | O sistema deve autenticar o usuário por meio de um token JWT | A | A |

\*B=Baixa, M=Média, A=Alta.

## Requisitos Não-funcionais

Os requisitos não funcionais de ambos os micros serviços serão os mesmos e são:

| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| --- | --- | --- |
| RNF01 | O sistema deve ser apresentar disponibilidade 24 X 7 X 365 | A |
| RNF02 | O sistema deve utilizar o banco de dados MySql | A |
| RNF03 | O sistema deve ser intuitivo | A |
| RNF04 | O sistema deve ser fácil de utilizar | A |
| RNF05 | O sistema deve ser responsivo | A |
| RNF06 | O sistema deve utilizar o padrão de orientação a objetos sob a plataforma .NetCore | A |

**Obs**: acrescente quantas linhas forem necessárias.

## Mecanismos Arquiteturais

O sistema será dividido em três micro serviços, todos desenvolvidos em .Net Core com o *design-pattern* CQRS. O primeiro é responsável pelas informações que envolvem os usuários do sistema e a autenticação. O segundo será responsável pelas informações que envolvem os veículos do sistema. O terceiro será responsável pelas informações relacionadas aos veículos do sistema.

Todos os serviços partilharam do mesmo banco de dados. Cada micro serviço não irá acessar informações de outro serviço.

A usabilidade e o visual do sistema serão apresentados em Angular, consumindo as informações disponibilizadas pelos micros serviços. As funcionalidades e união das funções disponibilizadas pelos micros serviços serão imperceptíveis para o usuário.

Todos os serviços serão disponibilizados na AWS, utilizando o ECS e o S3 para o armazenamento e publicação dos serviços em um servidor.

| **Análise** | ***Design*** | **Implementação** |
| --- | --- | --- |
| Persistência | Dapper | MySql |
| Front end | Single Page Application | Angular |
| Back end | Micro serviços | .Net Core |
| Deploy | EC2/S3 | AWS |

## Modelagem Arquitetural

A modelagem arquitetural apresentada na seção 4.1 será utilizada em ambos os micros serviços.

Para esta modelagem arquitetural optou-se por utilizar o modelo C4 para a documentação de arquitetura de software junto com componentes de serviços cloud no caso do banco de dados a ser utilizado.

## 4.1 Diagrama de Contexto

A figura 2 mostra a especificação o diagrama geral da solução proposta, com todos seus principais módulos e suas interfaces

Video 1 - https://photos.app.goo.gl/uo9V6rDbJFFp8uJi7