**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Projeto Integrado

Relatório Técnico

Aisoftware Tracker

Sistema de controle de frotas

Igor Araújo

Belo Horizonte, Brasil

Agosto, 2022

# Projeto Integrado – Arquitetura de Software Distribuído

**Sumário**

[Projeto Integrado – Arquitetura de Software Distribuído 2](#_Toc95767457)

[1. Introdução 3](#_Toc95767458)

[2. Cronograma do Trabalho 5](#_Toc95767459)

[3. Especificação Arquitetural da solução 6](#_Toc95767460)

[3.1 Restrições Arquiteturais 6](#_Toc95767461)

[3.2 Requisitos Funcionais 7](#_Toc95767462)

[3.3 Requisitos Não-funcionais 9](#_Toc95767463)

[3.4 Mecanismos Arquiteturais 10](#_Toc95767464)

[4. Modelagem Arquitetural 11](#_Toc95767465)

[4.1 Diagrama de Contexto 12](#_Toc95767466)

[Etapa 2 - Pendente 12](#_Toc95767467)

[Etapa 3 - Pendente 13](#_Toc95767468)

[Referências 13](#_Toc95767469)

## Introdução

Locomover-se e transportar cargas sempre foi uma necessidade humana, seja por terra, água ou ar. Na ultima década, principalmente no Brasil a frota de veículos terrestres vem aumentando vertiginosamente. Segundo dados do IBGE a frota de veículos terrestres no Brasil somando todas as categorias cresceu aproximadamente 35 % comparando os números absolutos entre 2010 e 2018, mostrando como é grande e promissor este mercado.

Em 2010 o número de veículos era de 64,8 milhões, já em 2018 este número subiu para mais de 100,7 milhões, em 2021, neste cenário adverso vivido pelo mundo em plena pandemia este número cresceu para 111,4 milhões de veículos.

Focando mais nos veículos de transporte terrestre como carros, caminhões, vans, e etc, temos números consideráveis também. Em 2010 havia cerca de 37,2 milhões de automóveis e 2,1 milhões de caminhoes, e em 2021 estes números vao para 59,2 milhões de 2,9 milhões respectivamente.

Com esta crescente vários novos tipos de negócios são desenvolvidos, não só pela possibilidade de levar seus produtos para os mais diversos locais do país, mas também com serviços, além dos bens de consumo. Dado a este crescimento, mecanismos para gerenciar e cuidar destes veículos são cada vez mais comuns, tanto para pessoas físicas, com sistemas de rastreamentos e geolocalização, como principalmente para empresas. Onde podem ter uma visão geral de toda a sua frota, além de trazer mais segurança e informação logística.

## Cronograma do Trabalho

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 01/06/2022 | 01/06/2022 | 1. Cronograma do Trabalho | Construção desta tabela |
| 02/06/2022 | 02/06/2022 | 1. Contextualização do trabalho | Construção da contextualização deste projeto |
| 03/06/2022 | 05/06/2022 | 1. Definição dos requisitos Arquiteturais | Lista dos requisitos Arquiteturais identificados |
| 03/06/2022 | 05/06/2022 | 1. Definição dos requisitos Funcionais | Lista dos requisitos funcionais identificados |
| 03/06/2022 | 05/06/2022 | 1. Definição dos requisitos Não-funcionais | Lista dos requisitos Não-funcionais identificados |
| 03/06/2022 | 05/06/2022 | 1. Definição dos Mecanismos Arquiteturais | Lista dos Mecanismos Arquiteturais identificados |
| 06/06/2022 | 08/06/2022 | 1. Construção dos Diagramas de Contextos – Modelo C4 | Diagrama de contexto criado no Draw.io e documentado |
| 08/06/2022 | 15/06/2022 | 1. Revisão da Etapa 1 | Documento Etapa 1 revisado |
| 16/06/2022 | 20/06/2022 | 1. Construção do vídeo de apresentação da Etapa 1 | Vídeo criado da Etapa 1 |
| 10/08/2022 | 10/08/2022 | 1. Apresentação em PPT da Etapa 1 | PPT |
| 10/08/2022 | 10/08/2022 | 1. Publicação no repositório Github Etapa 1 | Arquivos produzidos no Github disponíveis abertamente |
| 11/08/2022 | 12/08/2022 | 1. Construção dos Diagramas de Contêineres | Diagramas de contêineres |
| 13/08/2022 | 15/08/2022 | 1. Construção dos Diagramas de Componentes | Diagramas de componentes |
| 16/08/2022 | 18/08/2022 | 1. Desenho dos Wireframes da POC | Protótipos de telas de baixa fidelidade |
| 19/08/2022 | 30/09/2022 | 1. Código da aplicação | Aplicação com 3 requisitos implementados |
| 30/09/2022 | 30/09/2022 | 16. Publicação no repositório Github Etapa 2 | Arquivos produzidos no Github disponíveis abertamente |
| 01/10/2022 | 03/10/2022 | 17. Análise das abordagens arquiteturais | Seção do documento produzido |
| 04/10/2022 | 05/10/2022 | 18. Cenários | Seção do documento produzido |
| 06/10/2022 | 07/10/2022 | 19. Evidências da avaliação | Seção do documento produzido |
| 08/10/2022 | 09/10/2022 | 20. Resultados obtidos | Seção do documento produzido |
| 10/10/2022 | 11/10/2022 | 21. Avaliação crítica dos resultados | Seção do documento produzido |
| 12/10/2022 | 13/10/2022 | 22. Conclusão | Seção do documento produzido |
| 14/10/2022 | 19/10/2022 | 23. Construção do vídeo de apresentação da Etapa 3 | Vídeo da etapa 3 disponível |
| 20/10/2022 | 20/10/2022 | 24. Publicação no repositório Github Etapa 3 | Arquivos produzidos no Github disponíveis abertamente |

## Especificação Arquitetural da solução

Esta seção apresenta a especificação básica da arquitetura da solução a ser desenvolvida, incluindo diagramas, restrições e requisitos definidos pelo autor, tal que permitem visualizar a macroarquitetura da solução.

## Restrições Arquiteturais

Os Requisitos Arquiteturais são todos os requisitos, sejam eles Funcionais ou Não-Funcionais que têm **impacto direto** sobre a Arquitetura do Sistema. Dessa forma, o Arquiteto precisa analisar os requisitos do sistema identificando algumas propriedades e então “filtrando” os Requisitos Arquiteturais. A lista a seguir apresenta os requisitos arquiteturais que foram identificados para implementação inicial da plataforma.

|  |  |
| --- | --- |
| **ID** | **Descrição** |
| RA01 | Deve ser usado tecnologias abertas (Open source) para o desenvolvimento de toda a plataforma. |
| RA02 | Deve ser usado o serviço de nuvem da Amazon Web Services como provedora da infraestrutura necessária para a plataforma. |
| RA03 | Deve ser usado o serviço JWT para autenticação, com a possibilidade inicial de criação de conta diretamente na plataforma por outro usuário que tenha acesso de ADM. |
| RA04 | A solução móvel (Apps Mobile) o painel do sistema deve ser responsivo, para atender a demanda mobile. Inicialmente não terá Apps nativos para mobile |
| RA05 | Deve se implementar uma API RESTful para prover todos os dados e comunicação com os clientes de forma agnóstica e desacoplada com o backend, facilitando uma possível mudança tecnológica no frontend. Consumo dee API de terceiro para obter informações dos rastreadores instalados nos veículos |

## Requisitos Funcionais

Os Requisitos Funcionais são todos aqueles que estão associados às funcionalidades que ditam **o que** sistema deve fazer. A lista a seguir apresenta os requisitos funcionais identificados para o desenvolvimento inicial da plataforma.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | O aplicativo deve permitir o cadastro de Usuário através de uma conta do Painel admin, por um outro usuário Administrador, contendo e-mail e senha | B | A |
| RF02 | ~~O aplicativo deve permitir o cadastro de usuário através de um e-mail e senha.~~ | B | A |
| RF03 | O aplicativo deve permitir a atualização do cadastro do usuário, podendo adicionar: endereço, coordenadas e telefone. | M | A |
| RF04 | O aplicativo deve permitir o cadastro de Motoristas | A | A |
| RF05 | O aplicativo deve permitir o cadastro de Dispositivos (veículos), contendo informações de: Cliente, IMEI, Categoria, Data de instalação, placa, modelo roteador | M | A |
| RF06 | O aplicativo deve permitir a geração de Relatório de Rotas | M | A |
| RF07 | O aplicativo deve permitir a geração de Relatório de Eventos | M | A |
| RF08 | O aplicativo deve permitir a geração de Relatório de Resumo | M | A |
| RF09 | ~~O aplicativo deve permitir o cadastro de fotos do pet, com legenda e data referência.~~ | A | A |
| RF10 | ~~O aplicativo deve permitir o cadastro de vídeos do pet, com legenda e data referência.~~ | A | A |
| RF11 | ~~O aplicativo deve permitir a atualização de um pet, informando: Peso, Medidas, Medicamentos, Vacinas, Fotos e Vídeos.~~ | A | A |
| RF12 | ~~O aplicativo deve permitir os prestadores a cadastrar e disponibilizar seus serviços.~~ | M | B |
| RF13 | ~~O aplicativo deve permitir os vendedores a cadastrar e disponibilizar seus produtos.~~ | M | B |
| RF14 | ~~O aplicativo deve permitir os donos a pesquisar e adquirir produtos ou serviços.~~ | A | B |
| RF15 | ~~O aplicativo deve permitir os prestadores a consultar um relatório com estatísticas sobre seus serviços.~~ | M | B |
| RF16 | ~~O aplicativo deve permitir os vendedores a consultar um relatório com estatísticas sobre seus produtos.~~ | M | B |
| RF17 | ~~O aplicativo deve permitir os donos configurarem quais notificações e frequência deseja receber.~~ | M | B |
| RF18 | ~~O aplicativo deve permitir o dono avaliar o produto ou serviço adquirido.~~ | M | B |
| RF19 | ~~O aplicativo deve permitir o prestador de serviço configurar as notificações que deseja em relação a seus serviços disponíveis.~~ | M | B |
| RF20 | ~~O aplicativo deve permitir o vendedor configurar as notificações que deseja em relação a seus produtos disponíveis.~~ | M | B |
| RF21 | ~~O sistema deve enviar informações referentes aos serviços para o prestador respeitando suas configurações.~~ | M | B |
| RF22 | ~~O sistema deve enviar informações referentes aos produtos para o vendedor respeitando suas configurações.~~ | M | B |

\*B=Baixa, M=Média, A=Alta.

## Requisitos Não Funcionais

Os Requisitos Não Funcionais estão associados às restrições de funcionalidades que **ditam como** o sistema deve fazer. A lista a seguir apresenta os requisitos não funcionais identificados para o desenvolvimento inicial da plataforma.

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | A plataforma deve habilitar a autenticação baseado no modelo JWT diretamente no sistema. | A |
| RNF02 | O sistema deve operar em tempo integral em 24h x 7d, tendo disponibilidade mínima de 90%, e para atender essa necessidade deverá usar a hospedagem Heroku com mecanismo de escala automática para responder ao aumento de demanda. | A |
| RNF03 | A comunicação entre o sistema Backend e Web da plataforma deve ser implementado através do padrão de serviços REST. | A |
| RNF04 | O sistema deverá utilizar uma API de terceiro para realizar as persistências de informações dos dispositivos rastreadores. | A |
| RNF05 | As notificações por e-mail ou push devem operar por meio de filas de mensagens por não haver a necessidade de serem em tempo real e atenderem a grande demanda. | A |
| RNF06 | A versão Web da plataforma deve suportar os navegadores modernos, prioritariamente Google Chrome e Mozila Firefox. | M |

## Mecanismos Arquiteturais

Os mecanismos arquiteturais representam conceitos técnicos fundamentais que serão padronizados por toda a solução. Eles são refinados durante o projeto em três estados, representados pelas três categorias de Mecanismos Arquiteturais:

* Mecanismo de Análise, que dá ao mecanismo um nome, uma descrição resumida e alguns atributos básicos derivados dos requisitos do projeto.
* Mecanismo de Design, que são mais concretos e assumem alguns detalhes do ambiente de implementação.
* Mecanismo de Implementação, que especifica a exata implementação de cada mecanismo.

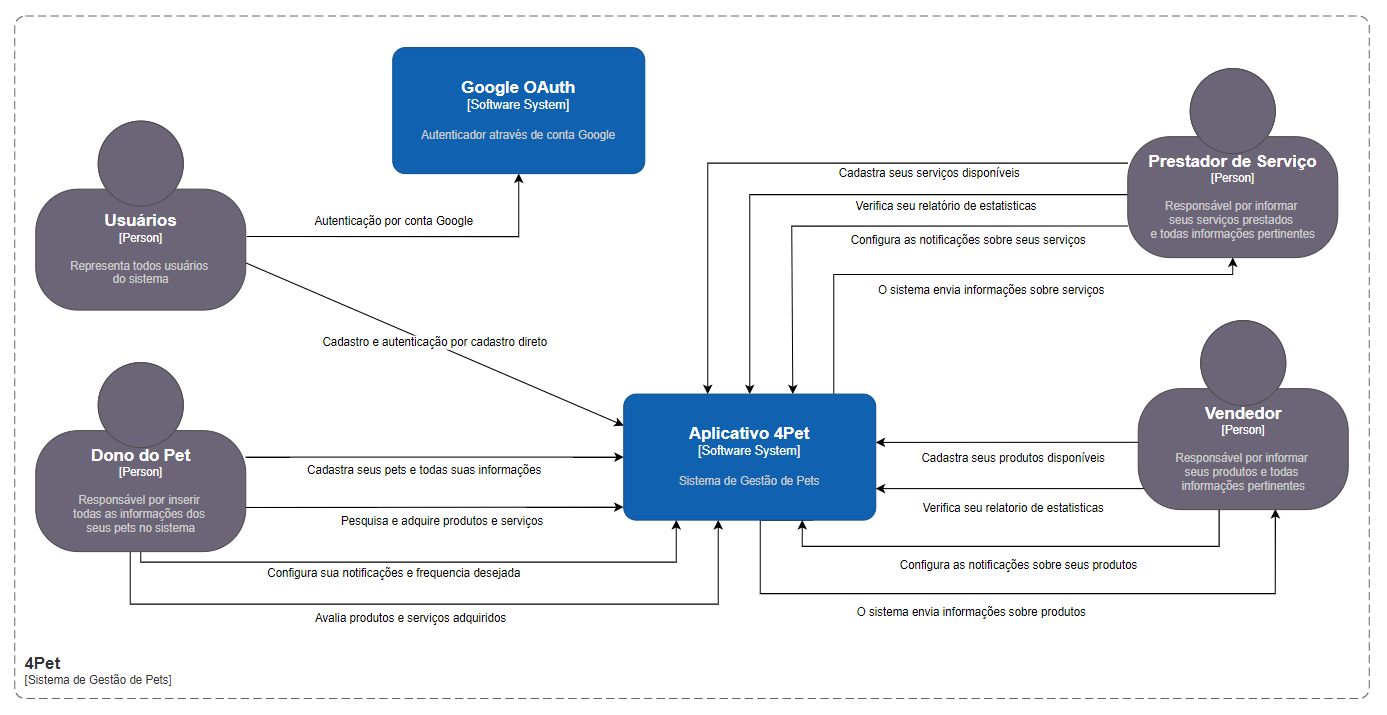
|  |  |  |
| --- | --- | --- |
| **Análise** | ***Design*** | **Implementação** |
| ~~Persistência~~ | ~~ORM~~ | ~~JPA / Hibernate / Spring Data~~ |
| ~~Persistência~~ | ~~Banco de Dados NoSQL~~ | ~~Amazon DynamoDB~~ |
| ~~Front end~~ | ~~GraphQL~~ | ~~Spring GraphQL~~ |
| Front end | Web | Asp Net Core |
| Front end | Navegador Web | Mozilla Firefox / Google Chrome |
| Teste de Software | Testes Unitários | XUnit |
| Usabilidade | Notificação por email | Asp Net |
| Autenticação | JWT | Microsoft.AspNetCore.Authentication.JwtBearer |

## Modelagem Arquitetural

Esta seção apresenta a modelagem arquitetural da solução proposta, de forma a permitir seu completo entendimento visando à implementação da Prova de Conceito (PoC) da plataforma Aisoftware Tracker na seção 5.

Para esta modelagem arquitetural optou-se por utilizar o modelo C4 para documentação de arquitetura de software. Mais informações a respeito podem ser encontradas aqui: <https://c4model.com/> e aqui: <https://www.infoq.com/br/articles/C4-architecture-model/>. Dos quatro níveis que compõem o modelo C4 três serão apresentados aqui e somente o Código será apresentado na próxima seção (5).

## 4.1 Diagrama de Contexto



**Figura 1 - Visão Geral da Solução 4Pet.**

A figura 1 mostra a especificação o diagrama geral da solução proposta, com todos seus principais sistemas e pessoas envolvidas nos processos de gestão de vida animal.

## 

## Etapa 2 - Pendente

*<Conteúdo a ser produzido – Data final 15 de abril>*

## Etapa 3 - Pendente

*<Conteúdo a ser produzido – Data final 15 de junho>*

## Referências

BRF INGREDIENTS. **Crescimento do mercado pet e seu papel na nutrição animal.**

Disponível em: <https://bit.ly/3OBxYIt>.

Acesso em: 24 de junho de 2022.

EXAME. **Mercado sem crise: com alta de 13,5% em ano de pandemia, o setor pet crescerá mais em 2021.**

Disponível em: <https://bit.ly/3u0Fsw4>.

Acesso em: 24 de junho de 2022.

CNN BRASIL. **Mercado pet deve ter crescimento de 14% em 2022, projeta instituto.**

Disponível em: <https://bit.ly/3ndS5jN>.

Acesso em: 24 de junho de 2022.

SEBRAE. **Mercado PET fatura quase 35 bi ao ano e tende a crescer.**

Disponível em: <https://bit.ly/3u3vhqO>.

Acesso em: 24 de junho de 2022.