BoltCHAT

Documento de Arquitetura de Software

*Versão 1.0*

# Histórico de Revisão

| **Data** | **Versão** | **Descrição** | **Autores** |
| --- | --- | --- | --- |
| 22/05/23 | 1.0 | Elaboração do documento |  |
|  |  |  |  |

# Sumário

[**Histórico de Revisão 2**](#_92ysk0btisfx)

[**Sumário 2**](#_uhb2u64flm6g)

[**1. Introdução 3**](#_6w1sxadrdjv5)

[1.1 Finalidade 3](#_vpxkuaz3s0ox)

[1.2 Escopo 3](#_pu1t4wvu2l0k)

[1.3 Definições, Acrônimos e Abreviações 3](#_ta1qein7c58r)

[1.4 Visão Geral 3](#_5aaepsi1sek5)

[**2. Representação Arquitetural 3**](#_fxh0lu126apd)

[**3. Metas e Restrições da Arquitetura 4**](#_d0fsa8hg02il)

[**4. Visão de Casos de Uso 4**](#_k08bufz5v1oa)

[**5. Visão Lógica 4**](#_99v222svi4zs)

[5.1 Visão Geral 4](#_precez250aj6)

[5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 5](#_8tmhdpfx1j7)

[**6. Visão de Implantação 5**](#_c23ope1ql1t6)

[**7. Visão da Implementação 5**](#_7r70szmjecji)

[**8. Tamanho e Desempenho 5**](#_rsxnvuizlxi6)

[**9. Qualidade 5**](#_m5nlpwqoenlu)

# 1. Introdução

O presente documento tem como objetivo descrever o documento de arquitetura do projeto BoltCHAT Analytics, um serviço de gerenciamento de dados para data-analytics para plataformas de atendimento omnichannel. A ideia básica por trás do gerenciamento dos dados do sistema é que eles serão armazenados em um repositório centralizado do tipo data warehouse (OLAP), preparado para data analytics via SQL.

## 1.1 Finalidade

Este documento oferece uma visão arquitetural geral do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do mesmo. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas no decorrer do processo de desenvolvimento.

## 1.2 Escopo

Este documento auxilia os envolvidos no projeto a captar aspectos arquiteturais do sistema que são necessários para o desenvolvimento de uma solução que atenda às necessidades dos usuários finais. Além de auxiliar no entendimento do sistema por novos membros da equipe.

## 1.3 Definições, Acrônimos e Abreviações

*MVC* – Padrão de arquitetura de software onde M significa modelo sendo responsável pela parte de regras de negócio, V a visualização responsável pela parte de interfaces e C a parte de controle dos dados.

OLAP (Online Analytical Processing) é um termo utilizado na área de banco de dados e análise de dados para descrever um conjunto de técnicas e ferramentas que permitem a análise multidimensional de dados.

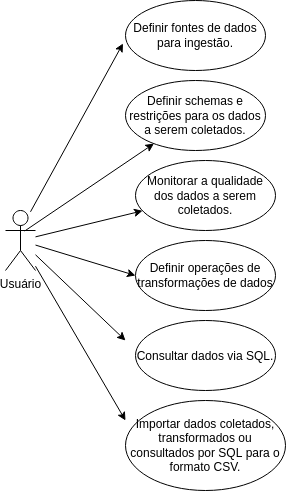
OLTP - OLTP (Online Transaction Processing), em português Processamento de Transações Online, refere-se a um tipo de sistema de banco de dados projetado para processar e gerenciar transações em tempo real. Essas transações podem incluir operações de inserção, atualização, exclusão e recuperação de dados em um ambiente de negócios em tempo real.

## 

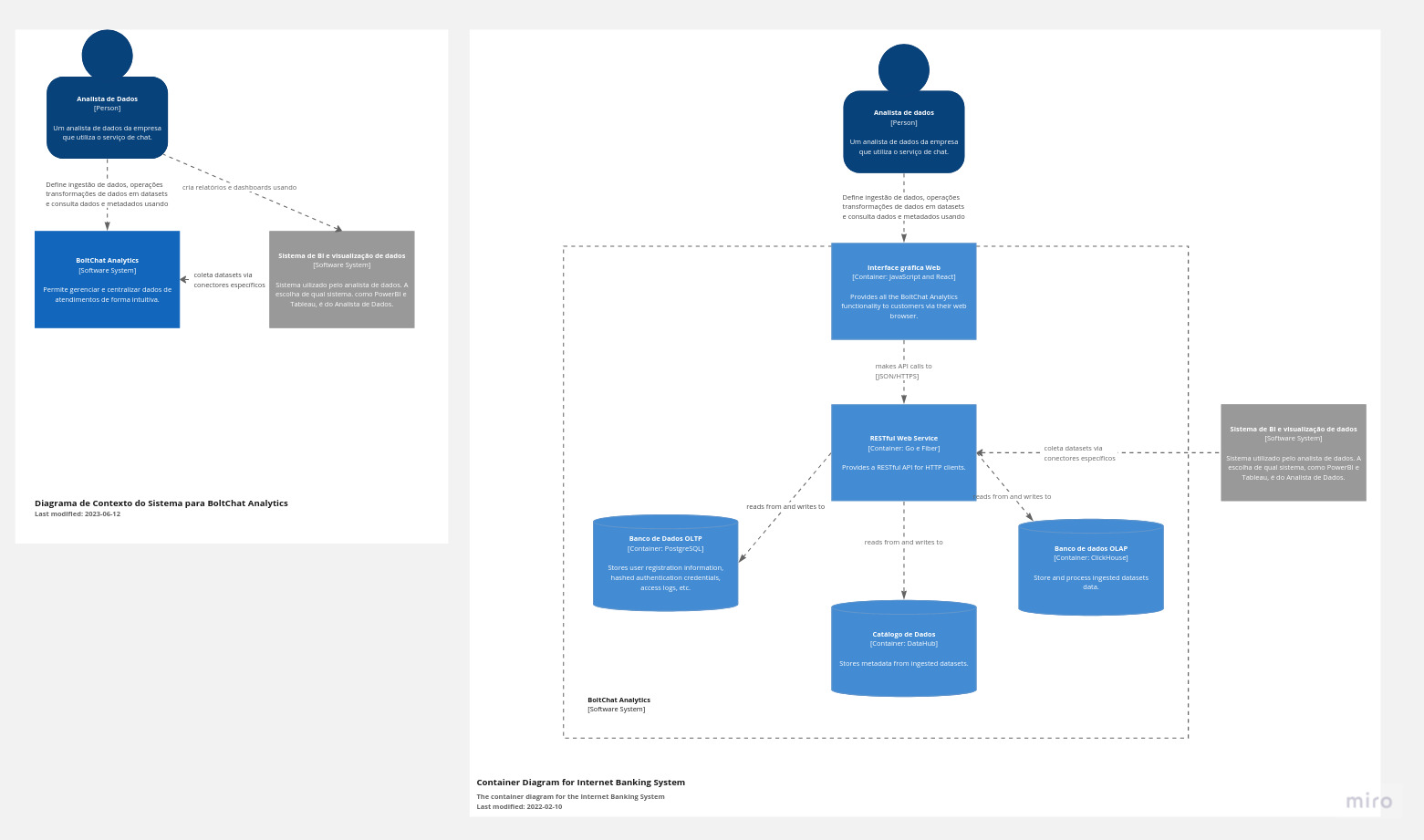
## 1.4 Visão Geral

# 2. Representação Arquitetural

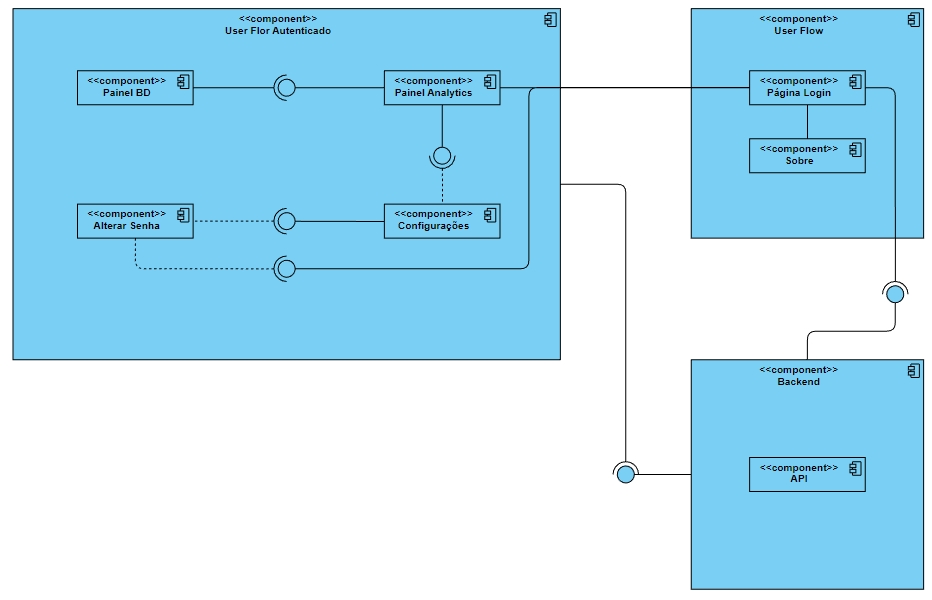
*Visão de caso de uso:*



*Visão de contexto e de contêineres:*



*Visão de componentes: front-end*



# 3. Metas e Restrições da Arquitetura

As principais restrições de arquitetura são os seguintes atributos de qualidade:

* Discoverabilidade de dados.
* UI intuitiva.
* Flexibilidade nos formatos de dados para importação e exportação.

# 4. Visão de Casos de Uso

Os casos de uso do sistema serão listados abaixo:

(imagem)

# 5. Visão Lógica

## 5.1 Visão Geral

A visão lógica define a estrutura da arquitetura. Abaixo será especificado o padrão utilizado para o desenvolvimento do sistema, no caso, MVC.

(imagem)

* View: Componente que contém as visões do projeto, ou seja, as interfaces, formulários e etc.;
* Controller: Componente que recebe as informações e requisições do pacote view e os despacha para devida classe de controle, o controller se comunica com classes BusinessLogic que contém as regras de negócio do sistema;
* Action - Model: pacote que recebe as informações e requisições do pacote controller e os atribui às respectivas classes do pacote bean e do pacote persistence, utilizando dados de seus modelos;
* Bean: pacote que recebe e armazena as informações referentes a seus respectivos objetos;
* Persistence: pacote que recebe as requisições e realiza as operações relacionadas ao Banco de Dados, utilizando dados provindos do pacote bean. O pacote persistence também pode enviar dados para o pacote view.

## 5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura

(imagem)

((legenda da imagem))

# 6. Visão de Implantação

O sistema será implantado quando for validado entre os stakeholders.

# 7. Visão da Implementação

O sistema será implementado utilizando conceitos de Programação Orientada a Objetos através do framework Ionic e Laravel, estrutura MVC, TypeScript, Web Container Apache, Linguagem de Programação PHP e Bancos de Dados Firebase e MySql.

# 8. Tamanho e Desempenho

A aplicação deve rodar através de um site próprio na versão web para os hemocentros e por meio de dispositivos móveis na versão mobile que será para o uso do usuário doadores.

# 9. Qualidade

O padrão de arquitetura adotado no projeto tem como finalidade garantir uma melhor organização do código-fonte, o que auxilia na manutenibilidade do software, bem como a portabilidade do mesmo.