Trabalho – Projeto e Arquitetura de Software

Gabriel Brunichaki, Marcelo Bernardy

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <24/09/19> | <1.0> | Primeira versão | Marcelo Bernardy, |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introdução 3](#__RefHeading___Toc563_393369374)

[1.1 Proposta 3](#__RefHeading___Toc565_393369374)

[1.2 Escopo 3](#__RefHeading___Toc567_393369374)

[1.3 Definições, Acrônimos e Abreviações 4](#__RefHeading___Toc569_393369374)

[1.4 Visão geral 4](#__RefHeading___Toc573_393369374)

[2. Representação da arquietetura 4](#__RefHeading___Toc575_393369374)

[3. Metas de arquitetura 4](#__RefHeading___Toc577_393369374)

[4. Use-Case View 4](#__RefHeading___Toc579_393369374)

[5. Logical View 5](#__RefHeading___Toc581_393369374)

[5.1 Overview 5](#__RefHeading___Toc583_393369374)

[5.2 Architecturally Significant Design Packages 5](#__RefHeading___Toc585_393369374)

[5.3 Use-Case Realizations 5](#__RefHeading___Toc587_393369374)

[6. Processos 6](#__RefHeading___Toc589_393369374)

[7. Execução 6](#__RefHeading___Toc591_393369374)

[8. Implementation View 6](#__RefHeading___Toc593_393369374)

[8.1 Camadas 7](#__RefHeading___Toc597_393369374)

[8.3 Padrões de projeto 7](#__RefHeading___Toc1637_393369374)

[9. Tamanho e performance 7](#__RefHeading___Toc599_393369374)

[10. Qualidade 7](#__RefHeading___Toc601_393369374)

# Introdução

O documento a seguir visa apresentar a arquitetura utilizada para o pograma Leilões, para com que possamos seguir boas práticas de arquitetura. Neste documento poderá ser visualizado o problema, a representação da arquitetura, visualização do processo, entre outras boas práticas de programação.

## Proposta

A proposta deste documento é prover uma arquitetura compatível com o escopo definido seguindo os melhores padrões e metas de reusabilidade, escala, etc.

## Escopo

O escopo é construir uma arquitetura de camadas perante os seguintes requisitos:

1. um cliente pode fazer lances e propor leilões;

2. no papel de proponente de um leilão, pode também fechar um leilão;

3. no papel de cliente pode dar quantos lances quiser;

4. no papel de administrador uma pessoa pode consultar os leilões concluídos e pendentes;

5. no papel de proponente, uma pessoa pode consultar todos os seus leilões ativos e fechados.

E no que diz respeito aos padrões de projeto:

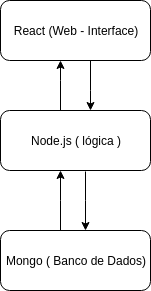
1. deve ser utilizado, pelo menos, 1 padrão de projeto em cada uma das camadas da aplicação;

## Definições, Acrônimos e Abreviações

## Visão geral

# Representação da arquietetura

A arquitetura utilizada vai seguir o padrão de Camadas, com uma parte de visualização realizada em React, camada de persistência e de negócio realizada em Node.js. Nosssa base de dados vai ser feita em Mongo.

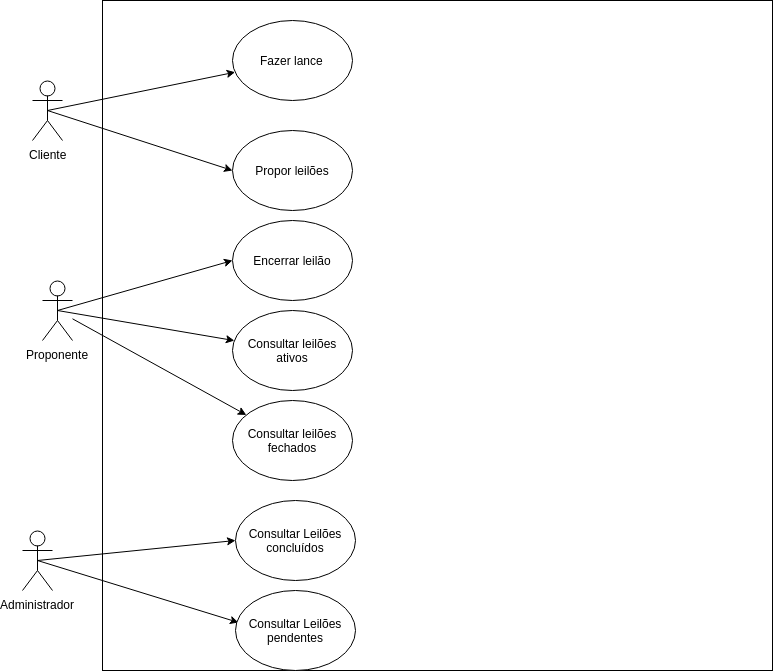


# Metas de arquitetura

A aplicação tem como objetivo obter performance, ter padrões de projeto bem definidos, além de

# Use-Case View

Os casos de uso dos requisitos estipulados são os seguintes:



# Logical View

## Overview

## Architecturally Significant Design Packages

[For each significant package, include a subsection with its name, its brief description, and a diagram with all significant classes and packages contained within the package.

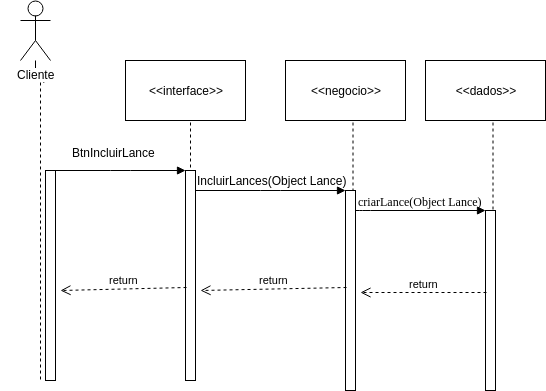
For each significant class in the package, include its name, brief description, and, optionally, a description of some of its major responsibilities, operations, and attributes.]

## Use-Case Realizations

[This section illustrates how the software actually works by giving a few selected use-case (or scenario) realizations, and explains how the various design model elements contribute to their functionality.]

# Processos

6.1 Cliente pode fazer e propor leilões

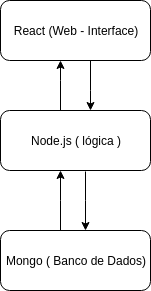
 Cliente clica em adicionar leilão …

# Execução

Para executar a aplicação é necessário acesso a internet. Para desenvolvimento é necessário um servidor web.

# Implementation View

Para a implementação será utilizado React (camada de apresentação), Node.js (API) e Mongo (Base de dados).



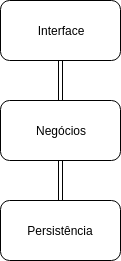
## Camadas

Para a arquitetura do projeto utilizamos as seguintes camadas:

Interface : Camada de exibição realizado em React.

Negócio : Camada onde toda a lógica de negócios.

Persistência : Camada de acesso ao banco de dados.



## 8.3 Padrões de projeto

# Tamanho e performance

A aplicação tem um grau de complexidade relativamente simples, o que causa por consequência um menor tamanho para a aplicação. No quesito de performance por a aplicação conter schema também é simples, e por utilizar o protocolo padrão REST API, a aplicação deve conter uma boa performance.

# Qualidade

A arquitetura definida deve manter uma boa performance, simplicidade e testabilidade do software, o índice de acoplamento deve ser baixo e de alta reusabilidade.