Sistema de Leilão

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 17/06/2019 | 1.0 | Coleta de Requisitos | Vinicius, Felipe |
| 19/06/2019 | 1.0 | Arquitetura do Sitema/ Desenvolvimento | Vinicius, Felipe |
| 21/06/2019 | 1.0 | Desenvolvimento | Vinicius, Felipe |
| 22/06/2019 | 1.0 | Desenvolvimento/ Documentação | Vinicius, Felipe |

# Introduction

Sistema de genérico de leilão, onde há dois atores, o Administrador e o Cliente. O cliente deve poder propor leilões tomando o papel de proponente do leilão em questão podendo fechá-lo e consultar todos seus leilões tanto os em andamento quanto os finalizados. O cliente no papel de participante do leilão pode dar quantos lances quiser. O administrador pode consultar todos os leilões criados no sistema, os em andamento e também os finalizados.

## Purpose

Este documento fornece uma visão geral da arquitetura do sistema, usando várias visões de arquitetura diferentes para representar diferentes aspectos do sistema. Destina-se a capturar e transmitir as decisões arquiteturais significativas que foram tomadas no sistema.

## Scope

Este documento abrange todas decisões arquiteturais para desenvolvimento da solução back-end utilizando a linguagem C#.

## Definitions, Acronyms, and Abbreviations

DAO: Objeto de acesso a dados

DTO: Objeto de Transferência de Dados

UML: Linguagem de Modelagem Unificada

# Architectural Representation

O sistema de leilões foi implementado utilizando o padrão de arquitetura em Camadas, sendo que em cada camada desenvolvida foi utilizado um padrão de projeto. Foram utilizadas as camadas de Negócio, Persistência, Apresentação e Controller. Os padrões de projetos em cada camada foram, factory na camada controller, DTO na camada de negócio, DAO na camada de persistência e ?? na camada de Apresentação.

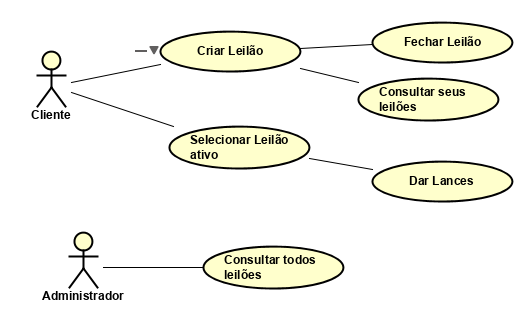
# Architectural Goals and Constraints

Por ser um Sistema de leilão e contar com transações monetárias a segurança é um requisito não funcional de alto impacto para o sistema, por isso a utilização da camada de persistência com um padrão de acesso a dados lidando com eventuais exceções e problemas que podem haver, além disso a camada Controller utilizando o padrão factory para exibição dos dados corretamente para cada usuário, assim evitando a exibição de dados indevidos para algum usuário, também abrangendo outro requisito de alto impacto no sistema que é o de privacidade.

Para termos um bom desempenho aliado a uma linguagem robusta que nos proporcione um desenvolvimento seguro e com bom desempenho optamos pela utilização da linguagem C# que é uma linguagem orientada a objetos fortemente tipada.

A equipe possui dois desenvolvedores, Felipe Bordignon e Vinícius Branco.

# Use-Case View



# Logical View

O sistema é composto por quatro principais pacotes sendo um para cada camada, então seria o pacote de Persistência, o de Negócio, o de Controller e a de Interface para exibição.

## Overview

- Persistência:

-DAOLeilão

-DAOUsuario

- Negócio:

-DTOLeilao

-DTOUsuario

-StatusLeilao

- Controller:

- LeilaoController

- UsuarioController

- Interface:

-TelaInicial

- DashboardUsuario

- DashboardAdmin

## Architecturally Significant Design Packages

- Persistência:

-DAOLeilão: Faz o acesso à base de Dados para o objeto do leilão e suas informações, como lances, dono, produto, entre outros.

-DAOUsuario Faz o acesso à base de Dados para o objeto do usuário e suas informações, como nome, cpf, entre outras.

- Negócio:

-DTOLeilao: Consta a model do Leilão contendo todos seus atributos.

-DTOUsuario: Consta a model do Usuário contendo todos seus atributos.

-StatusLeilao: Consta um enum com os possíveis estados do leilão.

- Controller:

- LeilaoController: Faz as funções necessárias para realização do leilão se comunicando com a camada de persistência e de exibição.

- UsuarioController: Realiza as funções do usuário dependo do usuário que está acessando as funções utilizando o padrão factory para devolver dados para os usuários.

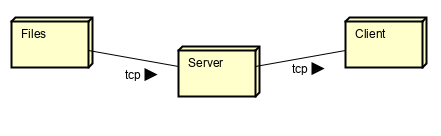
- Interface:

-TelaInicial: Tela inicial do sistema, contendo opções de login ou cadastro

- DashboardUsuario: Dashboard do usuário contendo todas funções disponíveis dependendo do seu papel

- DashboardAdmin: Dashboard do administrador contendo as funções disponíveis para o administrador

# Deployment View



# Implementation View

## 

# Data View (optional)

A camada da Persistência lida com o acesso a dados, buscando, inserindo e alterando. Utilizamos o padrão de projeto DAO para acesso a dados, assim lidando com eventuais problemas.

# Quality

Optando pela arquitetura em camadas, o sistema fica claramente dividido e cada camada com a sua responsabilidade, assim fica mais fácil de executar a manutenção do sistema e também está mais apto a possíveis alterações.