Sistema de Leilão

Software Architecture Document

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

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 |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Architectural Representation 4

3. Architectural Goals and Constraints 4

4. Use-Case View 4

5. Logical View 4

5.1 Overview 4

5.2 Architecturally Significant Design Packages 4

5.3 Use-Case Realizations 4

6. Process View 4

7. Deployment View 4

8. Implementation View 4

8.1 Overview 4

8.2 Layers 4

9. Data View (optional) 4

10. Size and Performance 4

11. Quality 4

Software Architecture Document

# 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

## References

[This subsection provides a complete list of all documents referenced elsewhere in the **Software Architecture Document**. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]

## Overview

[This subsection describes what the rest of the **Software Architecture Document** contains and explains how the **Software Architecture Document** is organized.]

# 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.

[This section describes what software architecture is for the current system, and how it is represented. Of the **Use-Case**, **Logical**, **Process**, **Deployment**, and **Implementation Views**, it enumerates the views that are necessary, and for each view, explains what types of model elements it contains.]

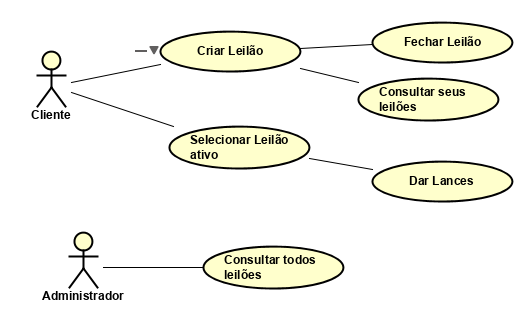
# Architectural Goals and Constraints

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

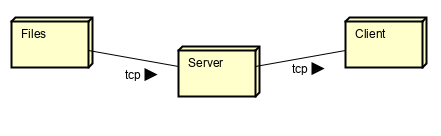
- DashboardUsuario: Dashboard do usuário contendo todas funções disponíveis dependendo do seu papel

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

[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.]

# Deployment View



# Implementation View

[This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components.]

## Overview

[This subsection names and defines the various layers and their contents, the rules that govern the inclusion to a given layer, and the boundaries between layers. Include a component diagram that shows the relations between layers. ]

## Layers

[For each layer, include a subsection with its name, an enumeration of the subsystems located in the layer, and a component diagram.]

# Data View (optional)

[A description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data, or the translation between the Design Model and the Data Model is trivial.]

# Quality

[A description of how the software architecture contributes to all capabilities (other than functionality) of the system: extensibility, reliability, portability, and so on. If these characteristics have special significance, such as safety, security or privacy implications, they must be clearly delineated.]