Documento de Requisitos de Sistema

**Instituição:** Universidade de São Paulo

**Nome do projeto:** Sistema de Leilão

**Autores:**

* Alberto Lages
* Thaisa Barros
* Uéviton Assunção Santos
* Patrick Douglas
* Gabriel Felipe da Silva
* Anderson Barbosa Cunha
* Gabriel de Moraes Macedo
* Caique Evaristo Santos da Silva
* Leandro de Oliveira Santos

**Orientador:** Rodolfo Riyoei Goya

**Local e Data:** São Paulo,1de novembro 2022

# Prefácio

Essa é a primeira versão do sistema de leilão. Essa solução foi desenvolvida tanto para pessoas físicas quanto jurídicas com o intuito de obter o melhor valor possível na venda dos produtos ofertados no sistema. Nesse software é possível cadastrar um produto para venda ou comprar um produto já cadastrado de acordo com a necessidade do cliente.

# Introdução

O objetivo deste documento é esclarecer aos clientes o funcionamento do Sistema de Leilão. O sistema foi criado para atender as necessidades de venda remunerada de maneira simples, prática e rápida para uma pessoa ou instituição, permitindo o cadastro de produtos a serem leiloados de forma segura e teremos uma restrição para divulgação de produtos lícitos. Mas como funciona? O cliente irá criar uma conta de acesso através de um pequeno formulário disponível na tela principal, o processo de cadastro é simples, será solicitado a validação de alguns dados como CPF, Nome, e-mail, data de nascimento, telefone e endereço e após criar seu acesso ele autenticará no sistema e após isso ele irá decidir se irá fazer uma oferta ou cadastrar um produto para o leilão. Caso o cliente queira cadastrar uma oferta ele será direcionado para uma tela onde será solicitado o nome do produto a ser ofertado, a descrição e o valor mínimo; Caso o cliente queira realizar um lance de um produto ele será direcionado direto para a tela de produtos disponíveis, após isso ele poderá escolher o produto e fazer a oferta que desejar.

# Glossário

>Diagrama de Casos de Uso: Descreve a funcionalidade proposta para um novo sistema que será projetado, é uma excelente ferramenta para o levantamento dos requisitos funcionais do sistema.

>Diagrama de Classes: É uma representação da estrutura e relações das classes que servem de modelo para objetos.

>Diagrama de Atividades: É essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra e serão empregados para fazer a modelagem de aspectos dinâmicos do sistema. Na maior parte, isso envolve a modelagem das etapas sequenciais em um processo

>Diagrama de Sequência: É um diagrama usado em UML, representando a sequência de processos num programa de computador. Como um projeto pode ter uma grande quantidade de métodos em classes diferentes, pode ser difícil determinar a sequência global do comportamento.

>Padrão MVC: É um padrão de projeto de software, ou padrão de arquitetura, focado no reuso de código e a separação de conceitos em três camadas

>Java: É uma linguagem de programação orientada a objetos. Diferente das linguagens de programação modernas, que são compiladas para código nativo, Java é compilada para um bytecode que é interpretado por uma máquina virtual (Java Virtual Machine, abreviada JVM).

> Front-End: Está relacionado com a interface gráfica do projeto. Ou seja, é onde se desenvolve a aplicação com a qual o usuário irá interagir diretamente, seja em softwares, sites, aplicativos, etc.

>Back-End: Cuida das engrenagens de uma aplicação web, criando códigos para que as funções do site sejam executadas. Como o próprio nome sugere, é um trabalho de bastidores.

>HTTP: É um protocolo de transferência que possibilita que as pessoas que inserem a URL do seu site na Web possam ver os conteúdos e dados que nele existem. A sigla vem do inglês Hypertext Transfer Protocol.

>Servidor: É um computador que inclui programas necessários para atender coletivamente aos requisitos de uma empresa, em vez de um usuário individual, unidade ou aplicativo específico.

>Aplicação Cliente: é o componente solicitante de um serviço e envia solicitações para vários tipos de serviços ao servidor. As solicitações feitas pelo cliente ao servidor serão realizadas através de métodos HTTP, como: GET, POST, DELETE, PUT entre outros.

>Aplicação Servidor: É o componente que é o provedor de serviços e fornece continuamente serviços ao cliente conforme as solicitações. As solicitações HTTP são recebidas e processada s pelo servidor e ele pode executar retornar alguma tarefa solicitada pelo cliente.

>Entity - Entity representa os objetos do banco de dados mapeados em classes, ou seja, funcionam como espécies de espelhos da base. Essa camada, possui apenas os atributos e seus devidos recursos de getters e setters com as referências para as colunas das consultas feitas na base de dados.

>Controller - O primeiro componente é o controller, ele é responsável por fazer a interface como usuário, ou seja, é ele quem recebe os parâmetros de entrada, faz as devidas validações de dados, repassa para os próximas componentes e aguarda o processamento para fazer o retorno ao usuário.

>Service: O componente service é responsável por encapsular as regras de negócio e fazer o orquestramento entre controllers e repositories. A camada service pode utilizar as entities para preparar um objeto para ser persistido na base ou para receber um objeto já persistido.

>Repository: O componente repository é responsável por fazer a abstração para o acesso ao banco de dados, é nessa camada onde as operações necessárias a base são feitas. O repository faz o mapeamento da base para as entidades e a persistências das entidades para o banco de dados..

>Framework: Significa estrutura. De maneira geral, essa estrutura é feita para resolver um problema específico. Na programação, um framework é um conjunto de códigos genéricos capaz de unir trechos de um projeto de desenvolvimento.

>Spring Boot: É uma ferramenta que nasceu a partir do Spring, um framework desenvolvido para a plataforma Java baseado nos padrões de projetos, inversão de controle e injeção de dependência.

>Aplicação Web Services API REST (Representational State Transfer): é um estilo de arquitetura para sistemas de hipermídia distribuídos, como a World Wide Web.

>Entidades controllers: Local de armazenamento de controle

>Entidade services: Local de armazenamento de serviço

>Entidade repositories: Local de armazenamento de inserções, consultas e atualizações

>Método GET: Get é usado para recuperar ou obter as informações de um determinado servidor usando um determinado URL

>Método POST: usado para enviar dados ao servidor, como fazer upload de um arquivo ou transferir alguns dados ou adicionar uma nova linha à tabela

>Método DELETE: Método para excluir toda a coleção

>Método PUT Método: utilizado para atualizar recursos.

>Requisitos Funcionais: São todos os problemas e necessidades que devem ser atendidos e resolvidos pelo Software por meio de funções ou serviços.

# Definição de Requisitos de Usuário

Requisitos de Usuário:

- O sistema irá permitir o cadastro de usuários com nome, e-mail e senha.

- O sistema irá permitir o login dos usuários através do e-mail e senha.

- O sistema irá permitir que um usuário cadastre um produto para ser leiloado.

- O sistema não apresentará a outros usuários dados de cunho privativo.

- O sistema deverá guardar os dados de usuários de forma segura.

- O sistema não deverá exceder o tempo de resposta de 10s pra cada requisição.

- O sistema deverá não permitir que um produto cadastrado por um usuário seja leiloado para o mesmo.

- O sistema deverá ter alta disponibilidade, por exemplo, 99% do tempo.

- O sistema não deverá permitir que um lance de um usuário em um produto seja menor que o valor especificado pelo leiloeiro.

- O sistema irá fornecer suporte ao usuário após êxito na compra de um produto leiloado.

- O sistema não irá disponibilizar meios de pagamentos na plataforma.

- O sistema não irá disponibilizar meios de entrega para produtos adquiridos na plataforma.

- O sistema terá tempo limite para cada leilão.

DIAGRAMAS :Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Gráfico, Gráfico de cascata

Descrição gerada automaticamente

# Arquitetura do Sistema

O sistema será desenvolvido na arquitetura Cliente/Servidor padrão MVC (Model, View, Controller) , como uma aplicação Web Services API REST (Representational State Transfer) utilizando a linguagem JAVA e o framework Spring Boot, formada por clientes, servidores e recursos, com solicitações gerenciadas por HTTP. O funcionamento será através da Aplicação Cliente (Front-end): páginas estáticas e dinâmicas, Aplicação Servidor(Back-end): entidades, controllers, services, repositories, e base de dados.

Aplicação Cliente: é o componente solicitante de um serviço e envia solicitações para vários tipos de serviços ao servidor. As solicitações feitas pelo cliente ao servidor serão realizadas através de métodos HTTP, como: GET, POST, DELETE, PUT entre outros.

Aplicação Servidor: É o componente que é o provedor de serviços e fornece continuamente serviços ao cliente conforme as solicitações. As solicitações HTTP são recebidas e processada s pelo servidor e ele pode executar retornar alguma tarefa solicitada pelo cliente.

Entity - Entity representa os objetos do banco de dados mapeados em classes, ou seja, funcionam como espécies de espelhos da base. Essa camada, possui apenas os atributos e seus devidos recursos de getters e setters com as referências para as colunas das consultas feitas na base de dados.

Controller - O primeiro componente é o controller, ele é responsável por fazer a interface como usuário, ou seja, é ele quem recebe os parâmetros de entrada, faz as devidas validações de dados, repassa para os próximas componentes e aguarda o processamento para fazer o retorno ao usuário.

Service - O componente service é responsável por encapsular as regras de negócio e fazer o orquestramento entre controllers e repositories. A camada service pode utilizar as entities para preparar um objeto para ser persistido na base ou para receber um objeto já persistido.

Repository - O componente repository é responsável por fazer a abstração para o acesso ao banco de dados, é nessa camada onde as operações necessárias a base são feitas. O repository faz o mapeamento da base para as entidades e a persistências das entidades para o banco de dados.

# Especificação de requisitos do sistema

Requisitos Funcionais:

- Todo leilão cadastrado deve ter o nome do vendedor.

- Apenas usuários cadastrados deverão efetuar lances em um leilão.

- Para efetuar um lance em um leilão o usuário deve estar logado.

- Um usuário não logado poderá listar os leilões disponíveis.

- Todo lance efetuado em um leilão deve ter o nome do arrematante.

- Todo lance efetuado em um leilão deverá ter um valor igual ou maior que o lance mínimo estipulado pelo vendedor.

- Para efetuar um cadastro de usuário todos os campos devem estar preenchidos.

- Para efetuar o cadastro de um leilão todos os campos devem estar preenchidos.

- Para efetuar o cadastro de um novo usuário o sistema deve verificar se o usuário já está cadastrado.

- Para efetuar o login do usuário, o sistema deve verificar se o usuário está cadastrado.

- Requisitos Não Funcionais:

- O sistema deverá guardar os dados de usuários de forma segura.

- O sistema não deverá exceder o tempo de resposta de 10ms pra cada requisição.

- O sistema deve rodar na maioria dos navegadores(Chrome, Edge, Mozilla, Brave ...).

- O sistema deve estar disponível na versão mobile.

- O sistema deverá ser desenvolvido em linguagem Java.

- O banco de dados deverá ser não relacional (mongoDB).

- O sistema deverá disponibilizar várias formas de pagamentos.

- O sistema deverá disponibilizar opções de entrega para leilões arrematados.

# Modelos do Sistema

Uma imagem contendo Linha do tempo

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente com confiança média

# Evolução do Sistema

O sistema se baseia em um leilão, onde um usuário irá se cadastrar no site e realizar tanto compras como cadastrar qualquer tipo de produto para ser leiloado.

O leilão será um site, portanto não irá exigir um hardware com configurações altas do usuário, porém onde será hospedado o servidor esse sim tem alguns requisitos básicos para que a aplicação funcione.

No caso de mudanças para melhor experiencia do usuário, a aplicação suportara alterações e atualizações caso elas sejam necessárias. No fluxograma da aplicação estão todas as funcionalidades e informações de banco de dados, facilitando uma possível alteração no código ou no banco de dados, capacitando qualquer profissional com um conhecimento em programação e banco de dados a realizar alterações ou atualizações no sistema.

# Apêndices

O sistema Leilão será desenvolvido utilizando a linguagem JAVA, framework Spring Boot, e banco de dados NoSQL (Não Relacional) MongoDB.

Especificações de Hardware - Quanto de Hardware precisa para rodar o servidor/sistema e requisitos de bancos de dados.

Servidor/Banco de Dados:

- Processador Intel i3 ou superior

- Memória RAM:4 GB

- Espaço em Disco: 10 GB disponível para instalação da Aplicação e Banco de Dados, podendo variar conforme o volume de dados do cliente.

Requisitos de Banco de Dados:

- Servidor de Acesso a Base de Dados

- Base de Dados

- Tabelas/Documentos

- Relacionamento entre tabelas/Documentos 1 para muitos

# Índice

1. Prefácio ...............................................................................................................................2
2. Introdução ...........................................................................................................................2
3. Glossário ..............................................................................................................................2
4. Requisitos de usuário ...........................................................................................................4
5. Diagramas .............................................................................................................................5
   1. Figura 1 - Diagrama de casos de uso...............................................................................5
   2. Figura 2 - Diagrama de classes .......................................................................................5
   3. Figura 3 - Diagrama de atividades...................................................................................6
   4. Figura 4 - Diagrama de sequência ..................................................................................6
6. Arquitetura do sistema .........................................................................................................7
7. Especificação de requisitos do sistema ................................................................................8
   1. Requisitos funcionais......................................................................................................8
   2. Requisitos não funcionais ..............................................................................................8
8. Modelo do sistema ...............................................................................................................9
   1. Figura 5 - Modelo de fluxo de dados .............................................................................9
   2. Figura 6 - Modelo de objetos .........................................................................................9
9. Evolução do sistema ...........................................................................................................10
10. Apêndices ...........................................................................................................................10