

Definição e Documentação das Decisões Arquiteturais

Grupo: Objetivo

1. Como e onde o sistema deve rodar (web, mobile, desktop, IoT, etc)?

O sistema será web/mobile, visto que o objetivo é atender até milhares de clientes com múltiplos leilões acontecendo ao mesmo tempo, é necessário que tenha suporte para ambas plataformas. A ideia de ser um sistema desktop, poderia ser para um grupo menor de usuário, imaginando que existem leilões presenciais, e eventos presenciais que poderiam utilizar esse sistema.

2. Quais linguagens de programação serão utilizadas (e.g., Java, JavaScript, Python)?

O sistema será desenvolvido em com a linguagem Java, principalmente porque é um sistema robusto que consegue suportar grandes quantidades de usuários, utilizando do paradigma da programação orientada objeto, também conta com a funcionalidade de threads, que permite ao sistema operacional intercalar rapidamente os processos ativos para ocuparem a CPU, dando a impressão de que estão sendo executados simultaneamente.

3. Quais são as principais dependências necessárias para este sistema (frameworks, bibliotecas, etc)?

Os frameworks necessários para este projeto serão o Spring, Hibernate, e para o front end utilizaremos o Vue.js.

- Spring: O spring conta com um conjunto de funcionalidades que conseguem otimizar o tempo e aumentar a produtividade, visto que ele simplifica o desenvolvimento de aplicações para os desenvolvedores.
- Hibernate: É um framework para mapeamento objeto-relacional escrito em linguagem Java, responsável pela ligação com o banco de dados.
- Vue.JS: Responsável pela visão do usuário.

4. Qual ou quais tipos de armazenamento são os ideais para este sistema (relacional, baseado em documentos, chave-valor, etc)?

Para este nosso sistema é necessário um banco de dados relacional, como por exemplo o banco de dados MySQL Server, pois além de garantir a integridade dos dados, o MYSQL Server pode ser configurado para gerenciar o sistema de backup dos dados salvos no banco.

5. Quais mecanismos de autenticação e segurança são necessários?

Validação de email, celular, de fotos de documentos, será implementado a autenticação em dois fatores, e recaptchas para garantir que aquela pessoa que está acessando é de fato um ser humano, e não um BOT.

6. Como será mantida a integridade e disponibilidade dos dados?

Firewalls Cisco Endpoints com prevenção multifacetada, menor tempo de resposta e unificação de processos, resolvendo a maioria dos problemas de ataques, controle de tráfego, acesso a portas, liberação de acessos a locais/dispositivos, se tornando um serviço terceirizado, tratado diretamente de um servidor alocado ao mesmo, criptografia múltipla (Código de Ofuscação, Código de Virtualização, Código de criptografia, proteção de tabela de importação, compactação inteligente, etc.) para proteger a lógica de codificação.

7. Quais tipos de servidores são capazes de atender a demanda do sistema? Dica: estimem o possível número de requisições por segundo, tempo de resposta adequado, número estimado de usuários ativos simultâneos, possíveis picos de utilização (e.g., black friday e eventos similares em uma loja online), etc.

A ideia é utilizar um tipo de servidor que tenha a característica de elasticidade, que consiste em manter o provedor encarregado de prover as configurações da máquina, conforme a demanda dos clientes ao servidor, relacionados ao balanceamento de carga, escalabilidade e monitoramento.

A AWS oferece uma grande quantidade de serviços, como tecnologias de infraestrutura, armazenamento e bancos de dados, a tecnologias emergentes como machine learning e inteligência artificial, data lakes, análises, entre outros. A AWS também conta 90 normas de segurança e certificações de conformidade, e todos os 117 serviços da AWS que armazenam dados de clientes oferecem criptografia de dados.

Outra vantagem da AWS, é o Amazon Elasticsearch Service, que permite que você pague somente pelo que usa: não há custos antecipados nem requisitos de uso.

8. Quais padrões de organização interna serão utilizados (e.g, MVC, Clean Architecture, Layers, etc)?

Com o backend sendo desenvolvido e disponibilizado via REST, Web Service, é possível utilizá-lo para comunicar com o front-end web, aplicação mobile e aplicação desktop, desde que a integridade entre as aplicações seja mantida

Um dos padrões de organização interna que será utilizado é o MVC, dividindo as responsabilidades entre o model, view e o controller. O model é o modelo definido dos dados, o view é aquilo que o usuário irá ver na tela, e o controller faz a ligação entre o model.