

ARQUITETURA DO SISTEMA DE AUTOMATIZAÇÃO DE ORÇAMENTOS XSA UNIFORMES

1. Visão Geral do Sistema

Este documento descreve a arquitetura do sistema de automação de orçamentos para a empresa de confecção de uniformes XSA Uniformes. O sistema permitirá que os clientes realizem orçamentos de forma automatizada por meio de um formulário online, recebendo um valor calculado com base em critérios predefinidos. O sistema também oferecerá um fluxo de pagamento, gerará comprovantes, e fornecerá informações detalhadas tanto para o cliente quanto para a empresa.

2. Tecnologias Utilizadas

- **Linguagem de Programação:** Java
 - **Framework Backend:** Spring Boot
 - **Middleware:** JDBC (Java Database Connectivity)
 - **Banco de Dados:** Apache Derby
 - **ORM:** Java Persistence Architecture (JPA)
 - **Frontend:** HTML, CSS, JavaScript, Angular
-

3. Requisitos Funcionais

1. Formulário de Orçamento:

- O cliente preenche um formulário com os seguintes campos:
 - Quantidade de peças.
 - Tamanhos (XP, P, M, G, GG).
 - Cores.
 - Material.
 - Prazo desejado.
 - Design (upload de arquivos).
 - Local de entrega.
 - Modelos e tipos de peça.
 - Informações adicionais.
- Ao clicar em "Fazer orçamento", o sistema calcula o valor com base em critérios predefinidos.

2. Cálculo de Orçamento:

- O sistema utiliza algoritmos para calcular o valor do orçamento com base em:
 - Quantidade de peças.
 - Tipo de material.
 - Cores e complexidade do design.
 - Prazo de entrega.
 - O valor é exibido ao usuário.
 - 3. Processo de Pagamento:**
 - Após receber o orçamento, o cliente pode optar por:
 - Prosseguir para pagamento.
 - Retornar ao formulário.
 - Sair (com aviso de perda de informações).
 - 4. Pagamento:**
 - O cliente insere seus dados (nome, e-mail, telefone).
 - Aceita os termos do contrato (com link).
 - Escolhe a forma de pagamento:
 - Pix (chave de pagamento gerada).
 - Cartão de crédito (dados de cartão e parcelamento).
 - Após confirmação do pagamento:
 - Se bem-sucedido, o cliente recebe uma mensagem de sucesso e um e-mail com detalhes do pedido.
 - Se mal-sucedido, o sistema avisa sobre o erro e oferece opções para tentar novamente.
 - 5. Notificações por E-mail:**
 - Cliente recebe e-mail com:
 - Comprovante de pagamento.
 - Cópia do contrato.
 - Informações do pedido.
 - Empresa recebe um e-mail com todos os detalhes do pedido.
-

4. Arquitetura do Sistema

4.1 Camadas do Sistema

- **Apresentação (Frontend):**
 - **Tecnologias:** HTML, CSS, JavaScript, Angular.
 - O frontend será responsável pela interface de usuário, capturando os dados do formulário, interagindo com o backend para cálculos de orçamento e processo de pagamento.
- **Lógica de Negócio (Backend):**
 - **Tecnologia:** Java com Spring Boot.

- O backend será responsável pelo processamento das informações inseridas no formulário, cálculo do orçamento e validação dos dados de pagamento.
- **Padrões de Projeto:** Serão aplicados padrões como MVC (Model-View-Controller) e DAO (Data Access Object) para garantir a separação de responsabilidades e facilitar a manutenção.
- **Persistência de Dados:**
 - **Tecnologia:** JPA com JDBC e banco de dados Derby.
 - A camada de persistência será responsável pelo armazenamento dos dados do cliente, informações de pedidos e orçamentos.
 - O JPA será usado para mapear as entidades Java para as tabelas do banco de dados.

4.2 Fluxo de Dados

1. O usuário preenche o formulário de orçamento.
2. O frontend envia os dados para o backend através de requisições HTTP.
3. O backend recebe os dados e calcula o valor do orçamento com base em regras predefinidas.
4. O sistema exibe o valor calculado e oferece as opções de pagamento.
5. Caso o cliente prossiga com o pagamento, os dados são validados e armazenados.
6. O pagamento é processado e o sistema envia notificações por e-mail.

5. Estrutura de Dados

- **Entidade Cliente:**
 - idCliente
 - nome
 - email
 - telefone
- **Entidade Pedido:**
 - idPedido
 - idCliente (chave estrangeira)
 - quantidade
 - tamanhos
 - cores
 - material
 - prazoDesejado

- design (arquivo)
 - localEntrega
 - modelos
 - informacoesAdicionais
 - valorFinal
 - **Entidade Pagamento:**
 - idPagamento
 - idPedido (chave estrangeira)
 - metodoPagamento
 - statusPagamento
 - dataPagamento
-

6. Fluxo de Pagamento

1. O usuário seleciona a forma de pagamento (Pix ou Cartão).
 2. No caso de Pix, o sistema gera uma chave e exibe para o cliente.
 3. No caso de cartão, o cliente preenche as informações e o sistema valida os dados com a operadora de cartão.
 4. O backend valida o pagamento e, se aprovado, registra no banco de dados.
-

7. Segurança

- **Criptografia:** Informações sensíveis, como dados de pagamento e endereços, serão armazenadas de forma segura utilizando criptografia.
 - **Validação:** Todos os dados inseridos pelos usuários serão validados no frontend e backend para evitar falhas de segurança, como injeção de SQL ou scripts maliciosos.
-

8. Considerações Finais

Este sistema será desenvolvido utilizando boas práticas de Programação Orientada a Objetos (POO) e arquitetura em camadas. A utilização de Spring Boot e JDBC permitirá escalabilidade e integração eficiente com o banco de dados. Além disso, o uso de frameworks modernos no frontend garantirá uma experiência fluida para o usuário.