.

Faculdade de Informática e Administração Paulista

Domain Driven Design

Challenge Salesforce

Grupo Tokito Techs

RM553472 – Claudio Silva Bispo

RM552981 – Patricia Naomi Yamagishi

RM554021 – Sara Ingrid da Silva Pereira

Sumário

1. Introdução 3
2. Contexto 3
3. Funcionalidades 4
4. Entidades 5
5. Modelo Lógico Relacional (MER) 7
6. Funcionamento do Projeto 7
7. Tabela dos Endpoints (API Restful) 10
8. Exemplos Para Teste Endpoints 11
9. Apresentação Site 13
10. Links do Projeto 14
11. Diagrama de Classes 15

## Introdução

Em um mundo impulsionado pela tecnologia, nosso projeto nasce com o propósito de transformar a interação digital em uma experiência inclusiva e igualitária. Estamos comprometidos em redefinir como as informações no site da Salesforce são acessadas, assegurando que todas as pessoas, com ou sem habilidades técnicas, naveguem com facilidade e sem complicações.

            Identificamos que o atual acesso ao site apresenta desafios, e é nossa meta criar um portal que organiza e simplifica o fluxo de informações, tornando cada interação uma oportunidade de conexão e não um empecilho. Os formulários, longe de serem barreiras, serão vias de comunicação eficiente e intuitiva, facilitando cada passo do usuário no site da Salesforce.

## Contexto

O objetivo deste projeto é finalizar a camada backend para o novo portal da Salesforce, com foco em acessibilidade e melhoria da experiência do usuário. Nossa solução visa simplificar a resolução de dúvidas, agilizar o contato com os usuários e gerar leads com maior eficácia, sempre priorizando a acessibilidade.

Com a finalização da nossa sprint, conseguimos concluir a integração completa com o banco de dados. Isso garante que todos os dados digitados no site pelos usuários sejam armazenados de maneira segura e eficiente. A conclusão desta fase permite que nossa solução esteja pronta para ser testada e utilizada, proporcionando uma base sólida para futuras melhorias e expansões.

## Funcionalidades

Nosso portal oferece uma série de funcionalidades voltadas para melhorar a experiência do cliente e garantir acessibilidade a todos os usuários:

Funcionalidades Gerais

**Layout Estilizado:** Interface com ênfase na estética, funcionalidade e acessibilidade.

**Tour Guiado Interativo:** Tour pelos produtos oferecidos, com opções de aprendizado por textos, vídeos ou conversando com um agente virtual. **ChatBot:** Suporte refinado para sanar dúvidas com precisão e sem burocracia.

**Experiência do Cliente:** Monitoramento da experiência do visitante para melhorar nossos serviços.

**Login Simples:** Login rápido usando contas de plataformas parceiras (Google, Linkedin, GitHub).

**Formulário de Feedback:** Sessão para usuários deixarem suas opiniões sobre nossos serviços.

Funcionalidades de Acessibilidade

**Documentação Completa:** Página com textos, vídeos e áudios para guiar os usuários.

**Videos com Transcrição:** Descrição textual de todos os vídeos para facilitar o aprendizado.

**Tour Guiado pelo Teclado:** Navegação pelo teclado com foco em acessibilidade.

**Leitura em Libras:** Tradução de conteúdo em Libras para usuários com deficiência auditiva.

**Alteração de Tema:** Opção de tema claro ou escuro para melhor leitura.

**Ajuste de Tamanho da Fonte:** Aumento ou diminuição do tamanho da fonte utilizando o teclado.

**Leitura de Texto:** Opção de leitura de conteúdos por áudio. Definição do **Idioma:** Escolha do idioma preferido para a navegação.

Funcionalidades para Gerar Leads

**Formulário de Descoberta:** Auxílio na definição de produtos conforme as respostas do visitante.

**Rankeamento:** Análises com IA para identificar potenciais clientes. Dados Acessíveis: Consultas rápidas no banco de dados para gerar leads.

## Entidades

No desenvolvimento do novo portal da Salesforce, uma parte fundamental é a coleta e gerenciamento de dados dos usuários. Para isso, foram implementados formulários no site que capturam informações permitindo à Salesforce entender melhor as necessidades dos visitantes, fornecer suporte adequado e personalizar as interações. As entidades a seguir representam as principais interações do usuário trabalhados no site:

Questionário

A entidade Questionario armazenará as informações do visitante que tem interesse em entender qual ou quais produtos da Salesforce melhor se encaixam para resolver a dor que a sua empresa possui naquele momento. Esses dados serão armazenados de maneira eficaz, permitindo que a Salesforce realize uma análise mais precisa com as informações fornecidas. Isso proporciona um overview do problema que um possível cliente deseja solucionar, facilitando a tomada de decisões e a personalização das interações com os clientes. A coleta de informações já no primeiro contato ajuda a Salesforce a entender o que o usuário deseja, tornando as futuras interações mais eficientes e direcionadas.

QuestFeedback

A entidade QuestFeedback captura os detalhes dos feedbacks fornecidos pelos usuários através do formulário de avaliação no site. Este formulário é projetado para coletar opiniões e sugestões dos usuários, permitindo que a empresa entenda melhor as áreas de sucesso e as que precisam de melhorias. Isso é essencial para melhorar continuamente a experiência do cliente e os serviços oferecidos.

Contato

A entidade Contato captura os detalhes das mensagens enviadas através do formulário de contato no site, permitindo que a empresa gerencie e responda às consultas de forma eficiente. Esta tabela é fundamental para garantir que a empresa possa manter um registro organizado e acessível das comunicações com os usuários.

Cadastro

A entidade Cadastro captura os detalhes associados ao processo de registro dos usuários no site. Ela é essencial para gerenciar as informações dos usuários, permitindo que a empresa ofereça um acesso personalizado e seguro à plataforma.

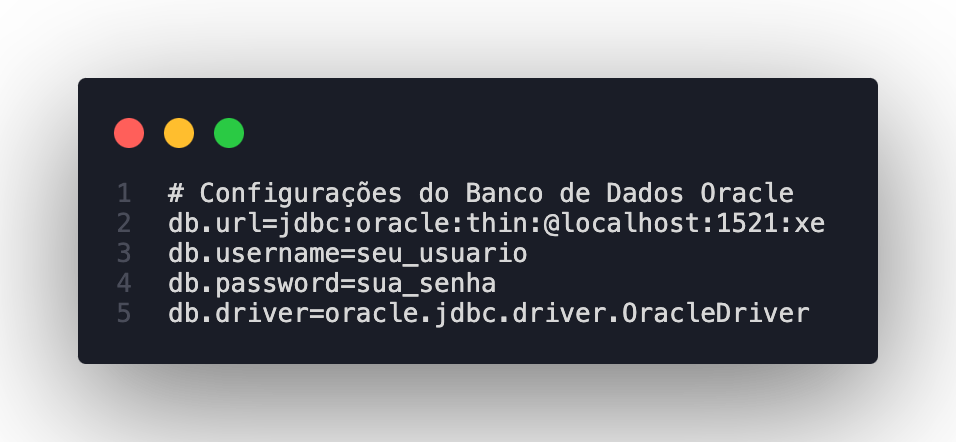
## Modelo Lógico Relacional (MER)modelo-relacional.png

## Funcionamento do Projeto

O projeto backend desenvolvido para o portal da Salesforce foi estruturado para garantir alta performance, escalabilidade e acessibilidade. A seguir, destacamos os componentes principais e seu funcionamento:

Configurações do Projeto

**Arquivo config.properties:**

O arquivo config.properties, localizado no diretório src/main/resources, contém as configurações essenciais para a conexão com o banco de dados Oracle. Este arquivo é lido pela classe de configuração durante a inicialização da aplicação, garantindo que as credenciais e a URL do banco de dados sejam carregadas corretamente.  
Exemplo de configuração no config.properties:

**Tomcat:**

O Apache Tomcat é utilizado como servidor de aplicação para hospedar e executar o backend do projeto. Após a construção do projeto usando Maven (mvn clean install), um arquivo .war é gerado e pode ser implementado no Tomcat. O Tomcat gerencia as requisições HTTP, direcionando-as para os endpoints RESTful definidos nas classes de recursos.

**Banco de Dados:**

O banco de dados Oracle é fundamental para o armazenamento de dados do projeto. Todas as informações capturadas através dos formulários no site, como cadastros, contatos, questionários e feedbacks, são armazenadas no Oracle Database. As operações de CRUD (Create, Read, Update, Delete) são realizadas utilizando os DAOs implementados, garantindo que as interações com o banco de dados sejam eficientes e seguras.

Funcionamento Geral

**Inicialização:**

Quando a aplicação é iniciada, o Tomcat carrega o contexto da aplicação e as configurações do config.properties.

A classe Config lê as propriedades do arquivo e as torna acessíveis para o restante da aplicação.

**Recepção de Requisições:**

Os usuários interagem com o frontend do portal, preenchendo formulários e enviando dados.

Essas requisições são recebidas pelos endpoints RESTful definidos nas classes de recursos (Resource).

**Processamento:**

Cada requisição é processada pelos controladores (Controller), que utilizam os objetos de negócio (BO) para aplicar regras de validação e lógica de negócio.

Os BOs interagem com os DAOs para realizar operações no banco de dados Oracle.

**Resposta:**

Após o processamento, as respostas são construídas e enviadas de volta ao cliente (frontend) através dos endpoints RESTful.

As respostas incluem códigos de status HTTP que indicam o sucesso ou falha das operações realizadas.

Fluxo de Trabalho

**Cadastro de Usuários:** O usuário preenche um formulário de cadastro, os dados são enviados para o endpoint /cadastro/inserir, onde são validados e armazenados no banco de dados.

**Contato:** Mensagens enviadas através do formulário de contato são processadas no endpoint /contato/inserir e armazenadas, permitindo que a equipe de suporte responda às consultas.

**Questionário:** O formulário de descoberta coleta informações sobre as necessidades da empresa do usuário, enviadas para o endpoint /questionario/inserir, ajudando na personalização de recomendações.

**Feedback:** Feedbacks dos usuários são enviados para o endpoint /feedback/inserir, armazenados e analisados para melhorar continuamente os serviços.

## Tabela dos Endpoints (API Restful)

|  |  |  |  |
| --- | --- | --- | --- |
| **Endpoint** | **Verbo HTTP** | **Descrição** | **Códigos de Status** |
| /cadastro/inserir | POST | Inserir um novo cadastro | 200 (OK), 500 (Internal Server Error) |
| /cadastro/atualizar | PUT | Atualizar um cadastro existente | 200 (OK), 500 (Internal Server Error) |
| /contato/inserir | POST | Inserir um novo contato | 200 (OK), 400 (Bad Request), 500 (Internal Server Error) |
| /questionario/inserir | POST | Inserir um novo questionário | 200 (OK), 500 (Internal Server Error) |
| /feedback/inserir | POST | Inserir um novo feedback de questionário | 201 (Created), 400 (Bad Request), 500 (Internal Server Error) |

## Exemplos Para Teste Endpoints

Cadastro - Inserir - POST

<http://localhost:8080/projetoTeste/rest/cadastro/inserir>

{

"nome": "João",

"sobrenome": "Silva",

"email": "joao@example.com",

"celular": "11999999999",

"senha": "senha123",

"data": "2023-05-15T10:00:00",

"generoId": 1,

"sobreEmpresaId": 1

}

Cadastro - Atualizar - PUT

<http://localhost:8080/projetoTeste/rest/cadastro/atualizar>

{

"id": 1,

"nome": "João",

"sobrenome": "Silva",

"email": "joao@example.com",

"celular": "11999999999",

"senha": "novaSenha123",

"data": "2023-05-15T10:00:00",

"generoId": 1,

"sobreEmpresaId": 1

}

Contato - Inserir - POST

<http://localhost:8080/projetoTeste/rest/contato/inserir>

{

"nome": "João Silva",

"email": "joao@example.com",

"telefone": "123456789",

"segmento": "Tecnologia",

"cargo": "Gerente de TI",

"mensagem": "Gostaria de mais informações sobre os produtos.",

"produtoId": 1,

"tamanhoEmpresaId": 2

}

Questionario - Inserir - POST

<http://localhost:8080/projetoTeste/rest/questionario/inserir>

{

"nome": "João Silva",

"telefone": 123456,

"email": "joao@example.com",

"nomeEmpresa": "Empresa Exemplo",

"segmento": "Tecnologia",

"conheceSalesforce": "Sim",

"necessidadeEmpresa": "Automatização de processos",

"produtoImplantado": "Sim",

"empresaFuncionamento": 1,

"estrategiasMarketing": 4,

"iaAutomacoes": 3,

"desenvolverEstrategia": 4,

"melhorarComunicacao": 4,

"ambienteIntegrado": 3,

"aumentoProdutividade": 4,

"melhorarCapacitacao": 3,

"reducaoCusto": 4,

"aumentarConversao": 4,

"potencializarVendas": 4,

"acelerarVendas": 3,

"riquezaDados": 4,

"melhorarExperiencia": 4,

"mostrarDiferencial": 3,

"criarJornada": 4,

"mktOportunidade": 3,

"data": "2024-05-20T10:00:00Z"

}

Feedback - Inserir - POST

<http://localhost:8080/projetoTeste/rest/feedback/inserir>

{

"nome": "Teste",

"email": "teste@exemplo.com",

"avaliacao": 5,

"data": "2024-05-14T10:00:00Z",

"mensagem": "Ótimo serviço!"

}

## Apresentação SiteCaptura de Tela 2024-05-20 às 19.17.03.pngCaptura de Tela 2024-05-20 às 19.17.19.png

## Links do Projeto

**Link para o repositório do Gi****thub:** <https://github.com/patinaomi/sprint-3-fiap/tree/main/Domain%20Driven%20Design/sprint-4-ddd/projetoTeste>

**Link do nosso projeto hospedado:** <https://homologacao-challenge-salesforce.vercel.app>

## Diagrama de Classesumlsprint4.drawio.png