# CRM Usando Orientação a Objetos

Uma empresa deseja um sistema básico de **CRM** para gerenciar seus clientes e registrar interações sem usar estruturas dinâmicas como ArrayList ou List.

Cada **cliente** tem um **registro fixo de interações** com a empresa. Para esse exercício, o número máximo de interações por cliente será **5**. O sistema deve permitir:

1. **Criar clientes** com nome, email e telefone.

2. **Registrar interações** (chamadas, reuniões, emails trocados) para um cliente.

3. **Exibir informações do cliente** e o histórico de interações.

4. **Buscar interações específicas** do cliente com base em palavras-chave.

5. **Atualizar o status do cliente** baseado na última interação.

6. **Remover interações mais antigas automaticamente** se o limite de 5 for atingido.

**📌 Regras**

• O número máximo de interações por cliente é **5**. Se um cliente já tiver 5 interações, a mais antiga deve ser substituída pela nova.

• O status do cliente pode ser: **“Novo”, “Engajado”, “Fechando Negócio”, “Cliente Perdido”**.

• A última interação define o status:

• Se a última interação for um **email ou ligação** → status = "Engajado"

• Se for uma **reunião** → status = "Fechando Negócio"

• Se não houver interação por mais de **30 dias** → status = "Cliente Perdido"

**📌 Classe Cliente**

Crie uma classe Cliente com os seguintes atributos:

• nome (String)

• email (String)

• telefone (String)

• interacoes (Array de **5** posições para armazenar as interações do cliente)

• status (String) → Define o status atual do cliente

**📌 Métodos**

1. registrarInteracao(String tipo, String descricao, String data)

• Registra uma nova interação no histórico do cliente.

• Se o histórico já tiver **5 interações**, substitui a mais antiga.

• Atualiza o status do cliente.

2. exibirInformacoes()

• Exibe os dados do cliente e suas interações.

3. buscarInteracao(String palavraChave)

• Retorna interações que contêm a palavra-chave na descrição.

4. atualizarStatus()

• Atualiza o status do cliente baseado na última interação.

5. removerInteracaoAntiga()

• Se o cliente já tem **5 interações**, remove a mais antiga antes de adicionar uma nova.

Abaixo está a lista dos **tipos de dados** e **métodos** utilizados no código do exercício de CRM.

**1️⃣ Tipos de Dados Utilizados**

**Tipos Primitivos**

|  |  |
| --- | --- |
| **Tipo** | **Descrição** |
| String | Usado para armazenar nomes, emails, telefones e descrições de interações. |
| int | Contador de interações (contadorInteracoes) para gerenciar a posição no array. |
| long | Armazena o número de dias passados entre interações (diasPassados). |

**Tipos Não Primitivos**

|  |  |
| --- | --- |
| **Tipo** | **Descrição** |
| String[][] | Matriz para armazenar interações do cliente (interacoes[5][3]). Cada linha representa uma interação e contém: [tipo, descrição, data]. |
| LocalDate | Usado para manipulação de datas e cálculos de diferença entre datas. |

**2️⃣ Métodos Utilizados**

**Métodos de String**

|  |  |
| --- | --- |
| **Método** | **Descrição** |
| .contains(String) | Verifica se a String contém um determinado texto. Usado para buscar interações por palavra-chave. |
| .equals(String) | Compara duas Strings para verificar se são iguais. Usado para comparar tipos de interações (Email, Ligação, Reunião). |
| .parse(String) | Converte uma String em um objeto LocalDate. Usado para transformar datas de texto em objetos manipuláveis. |

**Métodos de LocalDate**

|  |  |
| --- | --- |
| **Método** | **Descrição** |
| LocalDate.parse(String) | Converte uma String no formato YYYY-MM-DD para um objeto LocalDate. |
| ChronoUnit.DAYS.between(LocalDate, LocalDate) | Calcula a diferença em dias entre duas datas. Usado para definir o status do cliente com base na última interação. |

**Métodos de Controle de Arrays**

|  |  |
| --- | --- |
| **Método** | **Descrição** |
| interacoes[i] = interacoes[i + 1] | Move os elementos de um array para a esquerda, removendo o mais antigo. |
| interacoes[4] = null | Remove a última interação ao substituir seu valor por null. |

**Métodos Personalizados**

|  |  |
| --- | --- |
| **Método** | **Descrição** |
| registrarInteracao(String tipo, String descricao, String data) | Adiciona uma nova interação ao array, removendo a mais antiga se necessário. |
| exibirInformacoes() | Exibe os dados do cliente e o histórico de interações. |
| buscarInteracao(String palavraChave) | Busca interações que contenham a palavra-chave informada. |
| atualizarStatus() | Atualiza o status do cliente com base na última interação e no tempo passado. |
| removerInteracaoAntiga() | Remove a interação mais antiga quando o limite de 5 interações é atingido. |

**📌 Como Testar**

public class Main {  
 public static void main(String[] args) {  
 Cliente cliente1 = new Cliente("João Silva", "[joao@email.com](mailto:joao@email.com)", "11999999999");  
  
 cliente1.registrarInteracao("Email", "Enviei proposta de serviço", "2024-01-10");  
 cliente1.registrarInteracao("Ligação", "Cliente pediu mais detalhes", "2024-01-15");  
 cliente1.registrarInteracao("Reunião", "Negociação finalizada", "2024-01-20");  
 cliente1.registrarInteracao("Email", "Cliente confirmou contratação", "2024-01-25");  
 cliente1.registrarInteracao("Ligação", "Ajustes no contrato", "2024-01-30");  
 cliente1.registrarInteracao("Email", "Cliente com dúvidas adicionais", "2024-02-05");  
  
 cliente1.exibirInformacoes();  
 cliente1.buscarInteracao("contrato");  
 }  
}