thalita:

## **🌍 1. O que significa aplicar ESG em um CRM de Seguros**

Adicionar ESG ao CRM significa **usar dados e automações** para apoiar **decisões sustentáveis, éticas e socialmente responsáveis** dentro do processo de relacionamento com clientes e gestão de seguros.

No contexto do seu projeto, isso pode ser feito de **três formas principais:**

| **Pilar ESG** | **Aplicação no CRM de Seguros** |
| --- | --- |
| 🌱 **E (Environmental)** | Incentivar seguros de veículos elétricos, imóveis sustentáveis ou clientes com práticas ecológicas, oferecendo **descontos e benefícios**. |
| 🤝 **S (Social)** | Acompanhar indicadores de impacto social, como programas de seguros voltados a famílias de baixa renda ou inclusão financeira. |
| 🏛️ **G (Governance)** | Garantir transparência nos dados de clientes, segurança da informação e controle ético nas interações (com logs e autenticação JWT). |

## **💭 2. Problema a Resolver**

Empresas do setor de seguros enfrentam o desafio de **tornar seus produtos e operações mais sustentáveis e socialmente responsáveis**, mas muitas não têm **sistemas inteligentes** que integrem **dados ESG** ao processo de decisão comercial.

## **🎯 3. Objetivo do Sistema**

Desenvolver um **CRM para seguros com foco em ESG**, que permita:

* Centralizar informações de clientes e apólices;
* Controlar o ciclo de oportunidades e contratos;
* **Mensurar e incentivar práticas sustentáveis**;
* Garantir **transparência e segurança de dados** (Governance);
* Automatizar **regras ESG** dentro do sistema.

## **🧱 4. Entidades Sugeridas**

| **Entidade** | **Atributos principais** | **Relacionamentos** |
| --- | --- | --- |
| **Usuário (Corretor)** | id, nome, email, senha, cargo | 1–N com Oportunidades |
| **Cliente** | id, nome, cpf, email, tipoCliente, impactoAmbiental | 1–N com Apólices |
| **Apólice (Seguro)** | id, tipoSeguro, valor, validade, esgScore, status | N–1 com Cliente |
| **Oportunidade** | id, descricao, tipoSeguro, valorProposto, statusNegocio | N–1 com Cliente e Usuário |

## **🧮 5. Funcionalidades Especiais (Service com ESG)**

Aqui estão **três ideias de lógicas inteligentes** que o grupo pode implementar no Service:

1. **Cálculo de ESG Score**
   * Cada cliente recebe uma pontuação ESG com base em seu perfil (tipo de seguro, histórico, sustentabilidade).

if (cliente.getTipoSeguro().equals("Residencial") && cliente.isImovelSustentavel()) {

apolice.setEsgScore(90);

} else {

apolice.setEsgScore(70);

}

1. **Descontos Sustentáveis**
   * Aplicar **desconto de 15%** no valor do seguro para clientes com veículos elétricos ou imóveis com certificação verde.
2. **Relatórios ESG**
   * Criar um endpoint que retorne estatísticas como:  
     + Quantos clientes têm selo sustentável;
     + Valor total economizado em seguros “verdes”;
     + Média de ESG Score das apólices ativas.

CRM

**Personalização Extrema e Previsão de Necessidades**.

### **CRM de Fidelidade e Engajamento de Influenciadores Foco:** Transformar clientes satisfeitos em promotores ativos (influenciadores).

### **CRM Preditivo de Vendas e Up-sell**

**Foco:** Usar dados históricos para prever a próxima compra e o valor dela.

**O CRM para Seguros**

| **Entidade Principal** | **Entidades Chave** | **Racional para Iniciantes** |
| --- | --- | --- |
| **Cliente** | **Apólice**, **AlertaRenovacao** | **Clareza de Negócio:** As entidades são intuitivas (Quem? O que comprou? Quando precisa renovar?). |
|  |  | **Foco no JWT:** A segurança será crucial para que um Cliente só possa ver suas próprias Apólices e AlertaRenovacao. |
|  |  | **Desafio de Lógica:** O AlertaRenovacao é gerado automaticamente com base na data de vencimento da Apólice, o que permite um ótimo exercício de lógica na camada de *Service* do Spring. |

| **Entidade** | **Propósito no Negócio** | **Atributos Essenciais para CRUD** |
| --- | --- | --- |
| **1. Cliente (Usuário)** | A pessoa que possui o seguro (para autenticação JWT). | id, nome, email, senha (criptografada), perfil (CLIENTE, CORRETOR), dataNascimento |
| **2. Apólice** | O contrato de seguro ativo do cliente. | id, numeroApolice, tipoSeguro (Auto, Vida, Residencial), valorSegurado, dataInicioVigencia, **dataFimVigencia** (crucial!), idCliente (FK) |
| **3. AlertaRenovacao** | O registro de acompanhamento para o Corretor/Cliente. | id, dataGatilho, dataContatoEfetivo, status (Pendente, Em Contato, Renovado, Perdido), idApolice (FK), corretorResponsavel (FK para outro Usuário) |

#### **Relacionamentos-Chave**

* **Cliente ↔ Apólice:** **One-to-Many** (Um Cliente pode ter várias Apólices).
* **Apólice ↔ AlertaRenovacao:** **One-to-Many** (Uma Apólice pode gerar vários Alertas ao longo da sua vida, especialmente se for renovada). O Alerta serve como um **log** de follow-up.

### **Desafio de CRUD e JWT**

1. **CRUD em Cliente:** Padrão (Cadastro e Consulta protegida por JWT).
2. **CRUD em Apólice:** **Corretores** criam e atualizam. **Clientes** apenas consultam as suas (filtrando pelo idCliente do JWT).
3. **CRUD em AlertaRenovacao:** O POST (criação) é uma lógica de negócio (ex: criar 90, 60 e 30 dias antes da dataFimVigencia da Apólice). O PUT é feito pelo Corretor para atualizar o status (ex: de "Pendente" para "Renovado").

Este modelo é desafiador o suficiente para aprender os relacionamentos no Spring Data JPA e a aplicação do JWT para diferentes perfis, mas é claro e direto.

## **Projeto: SeguroConecta — “CRM que protege e conecta pessoas.”**

### **Contradição central**

Combina **CRM corporativo de seguros** com **impacto social e engajamento comunitário**.  
 Paradoxo: CRMs tradicionais focam exclusivamente em dados de clientes, vendas e riscos financeiros, enquanto projetos sociais exigem impacto humano e mensuração de benefícios sociais.  
 O **SeguroConecta** resolve isso integrando informações de clientes com iniciativas de responsabilidade social, criando uma experiência de seguro que vai além do contrato: ele *gera valor social tangível* para clientes e comunidade.

### **Tendências de mercado incorporadas**

1. **SaaS + API-first:** plataforma modular que permite integração com sistemas de seguros existentes e com APIs de programas sociais de parceiros.
2. **IA/ML leve e explicável:** análise de perfis de risco + recomendação de iniciativas sociais personalizadas (ex.: campanhas de doação, voluntariado, educação financeira).
3. **Privacy-by-design + ESG mensurável:** coleta de dados com consentimento, anonimização e relatórios de impacto social vinculados ao compliance de LGPD/GDPR e relatórios ESG corporativos.

### **Problema real que resolve**

* Seguradoras enfrentam dificuldade em **engajar clientes com responsabilidade social** sem comprometer dados sensíveis.
* Clientes querem sentir que suas interações com seguros têm **impacto positivo na sociedade**, mas não existem sistemas que integrem CRM com monitoramento de impacto social.
* O gap é mensurável: baixa retenção e engajamento em programas de fidelidade social.

### **Público-alvo & contexto corporativo**

* **Clientes pagantes:** seguradoras de médio e grande porte, fintechs de seguros e plataformas de benefits corporativos.
* **Usuários:** gerentes de conta, equipes de relacionamento, e clientes finais interessados em impacto social.
* **Cenário:** integração com sistemas de CRM existentes da seguradora, painel web para análise de engajamento social e dashboards de impacto.

### **Proposta de valor**

* Aumenta **retenção de clientes** e fidelização, mostrando que a seguradora gera impacto social real.
* KPI principal: engajamento em programas sociais (% de clientes ativos em ações), renovação de seguros, satisfação (NPS).
* Diferencial competitivo: conecta proteção financeira com propósito social, transformando o seguro em ferramenta de impacto.

### **Descrição funcional (MVP)**

* **API CRM**: endpoints para CRUD de clientes, seguros e interações.  
  + /api/clients — cadastro e atualização de clientes.
  + /api/policies — registro de apólices e status de cobertura.
  + /api/social-actions — registro de participação em programas sociais (ex.: doações, voluntariado).
* **Dashboard web**: visualização de métricas de impacto social e engajamento de clientes.
* **Integração mock**: simulação de campanhas sociais de parceiros (ONGs, programas de educação, saúde).
* **Gamificação leve**: ranking de clientes ou equipes mais engajadas socialmente.

### **Arquitetura e stack técnico sugeridos**

* **Backend:** Java + Spring Boot + Spring Data JPA.
* **Banco de dados:** PostgreSQL (ou MySQL) para clientes, apólices e ações sociais.
* **Frontend:** React.js para dashboard, responsivo e visual.
* **Autenticação:** JWT + Spring Security.
* **Mock API**: para dados de parceiros sociais, evitando integrações complexas externas.

### **Privacidade, compliance & ESG**

* Coleta mínima de dados, anonimização e consentimento explícito.
* Relatórios de impacto social exportáveis para frameworks ESG corporativos.
* Cumprimento da LGPD/GDPR com registro de consentimento e possibilidade de exclusão de dados.

### **Monetização & go-to-market**

* **Modelo:** SaaS B2B (mensalidade por cliente/usuário ativo).
* **Freemium piloto:** até 50 clientes para validação de MVP.
* **Piloto:** parceria com 1 seguradora para integrar CRM + dashboard de impacto social.
* **Canal inicial:** aceleradoras, feiras de inovação PwC e programas de inovação aberta.

### **Justificativa de viabilidade técnica para iniciantes**

* CRUDs simples de clientes, apólices e ações sociais.
* Dashboard pode ser feito com bibliotecas prontas de visualização (Chart.js, Recharts).
* APIs externas simuladas, sem necessidade de integração paga.
* Escopo enxuto, entregável em 6–8 semanas por equipe de 5 iniciantes.

### **Riscos principais e mitigação**

1. **Risco técnico:** complexidade de integração com sistemas de CRM existentes → iniciar com mock API e endpoints REST simples.
2. **Risco legal:** dados sensíveis de clientes → aplicar anonimização, consentimento explícito e auditoria de logs.
3. **Risco de adoção:** clientes não se engajarem em ações sociais → gamificação, recompensas e comunicação de impacto claro.

### **Por que se destacaria numa feira da PwC**

* Une **CRM tradicional + propósito social**, provocando reflexão sobre inovação no setor de seguros.
* Mostra como tecnologia simples pode gerar impacto mensurável, alinhando **responsabilidade e rentabilidade**.
* MVP exequível por iniciantes, com narrativa clara e storytelling estratégico, ideal para impressionar avaliadores PwC.

## **Projeto Final: SeguroConecta — “CRM que protege, conecta e gera impacto”**

### **Contradição central**

Combina CRM corporativo de seguros com **impacto social e ambiental mensurável**.  
Paradoxo: CRMs tradicionais focam em clientes e contratos; aqui, eles também **monitoram e incentivam práticas sustentáveis e sociais**, gerando valor real para clientes e comunidade. Funciona porque conecta **proteção financeira** a **propósito social**, tornando seguro algo tangível e emocionalmente relevante.

### **Tendências de mercado incorporadas**

1. **SaaS + API-first:** API modular para integração com sistemas de seguros existentes e parceiros sociais.
2. **IA/ML leve:** cálculo de ESG Score personalizado e recomendações inteligentes de apólices sustentáveis.
3. **Privacy-by-design + ESG mensurável:** coleta mínima de dados, anonimização e dashboards exportáveis para relatórios ESG corporativos.

### **Problema real que resolve**

* Seguradoras precisam **atrair e engajar clientes com propósito social**, mas não têm ferramentas para integrar decisões de negócio e impacto social.
* O gap é mensurável: baixa retenção de clientes e baixa participação em iniciativas sociais ligadas ao seguro.

### **Público-alvo & contexto corporativo**

* Seguradoras de médio e grande porte, fintechs de seguros, e empresas de benefits corporativos.
* Usuários internos: corretores e gestores.
* Cenário: painel web para análise de impacto, CRM integrado para apólices e clientes, gamificação para engajamento social.

### **Proposta de valor**

* Fidelização de clientes e engajamento com propósito.
* Métricas-chave: % clientes engajados em ações sociais, renovação de seguros verdes, satisfação (NPS), CO₂ economizado ou impacto social estimado.
* Diferencial competitivo: seguro transforma-se em ferramenta de impacto social mensurável.

### **Entidades Aprimoradas**

| **Entidade** | **Atributos principais** | **Relacionamentos** | **Observações ESG** |
| --- | --- | --- | --- |
| **Usuário (Corretor/Gestor)** | id, nome, email, senha, cargo, scoreESGGestor | 1–N com Clientes e InteraçõesSociais | Score interno de performance ESG do gestor |
| **Cliente** | id, nome, cpf, email, tipoCliente, impactoAmbiental, impactoSocial, engajamentoGovernance | 1–N com Apólices, 1–N com InteraçõesSociais, 1–N com CertificacaoESG | Indicadores de ESG separados por pilar; valores percentuais ou categorias |
| **Apólice (Seguro)** | id, tipoSeguro, valor, validade, esgScore, status, incentivoESG, dataUltimaAtualizacao | N–1 com Cliente | esgScore dinâmico; incentivoESG registra descontos ou bônus por práticas sustentáveis |
| **InteracaoSocial** *(substitui Oportunidade)* | id, tipoInteracao, descricao, impactoEstimado, status, recomendacaoESG | N–1 com Cliente, N–1 com Apólice, N–1 com Usuário | Representa ações sociais ou ambientais vinculadas ao cliente/apólice; reforça engajamento ESG |
| **AcaoSocial** | id, tipoAcao, descricao, impactoEstimado, statusParticipacao | N–1 com Cliente, N–1 com Apólice | Participação em programas sociais ou ambientais; contribui para cálculo de ESG Score |
| **CertificacaoESG** | id, tipoCertificacao, dataEmissao, validade, impactoScore | N–1 com Cliente, N–1 com Apólice | Selos de sustentabilidade ou governança; influenciam ESG Score |

### **Descrição funcional (MVP)**

* **CRUD completo** para clientes, apólices e interações sociais.
* **Cálculo automático de ESG Score**: combina atributos ambientais, sociais e de governança.
* **Dashboard web**: métricas de impacto e engajamento de clientes, gráficos simples (ex.: CO₂ economizado, participação social).
* **Gamificação leve**: ranking de clientes e corretores mais engajados.
* **Endpoints sugeridos:**
  + /api/clientes → CRUD + indicadores ESG
  + /api/apolices → CRUD + cálculo de ESG Score
  + /api/interacoes-sociais → registrar ações vinculadas a clientes/apólices
  + /api/acoes-sociais → registrar participação em programas externos
  + /api/relatorios-esg → estatísticas consolidadas para gestores

### **Arquitetura e stack técnico sugeridos**

* **Backend:** Java + Spring Boot + Spring Data JPA
* **Banco de dados:** PostgreSQL ou MySQL
* **Frontend:** React.js para dashboard web
* **Autenticação:** JWT + Spring Security
* **Mock APIs:** para dados de programas sociais, evitando dependência de sistemas pagos
* **IA leve:** cálculo de ESG Score e recomendações inteligentes

### **Privacidade, compliance & ESG**

* Consentimento explícito, anonimização de dados sensíveis.
* Dashboards exportáveis para relatórios ESG corporativos.
* Logs e auditoria de interações garantem governança e compliance (LGPD/GDPR).

### **Monetização & go-to-market**

* Modelo **SaaS B2B**, mensalidade por cliente ativo.
* Piloto inicial: seguradora média para validar cálculo de ESG Score e dashboards de engajamento.
* Canal inicial: feiras de inovação (PwC), aceleradoras e programas de inovação aberta.

### **Justificativa de viabilidade técnica**

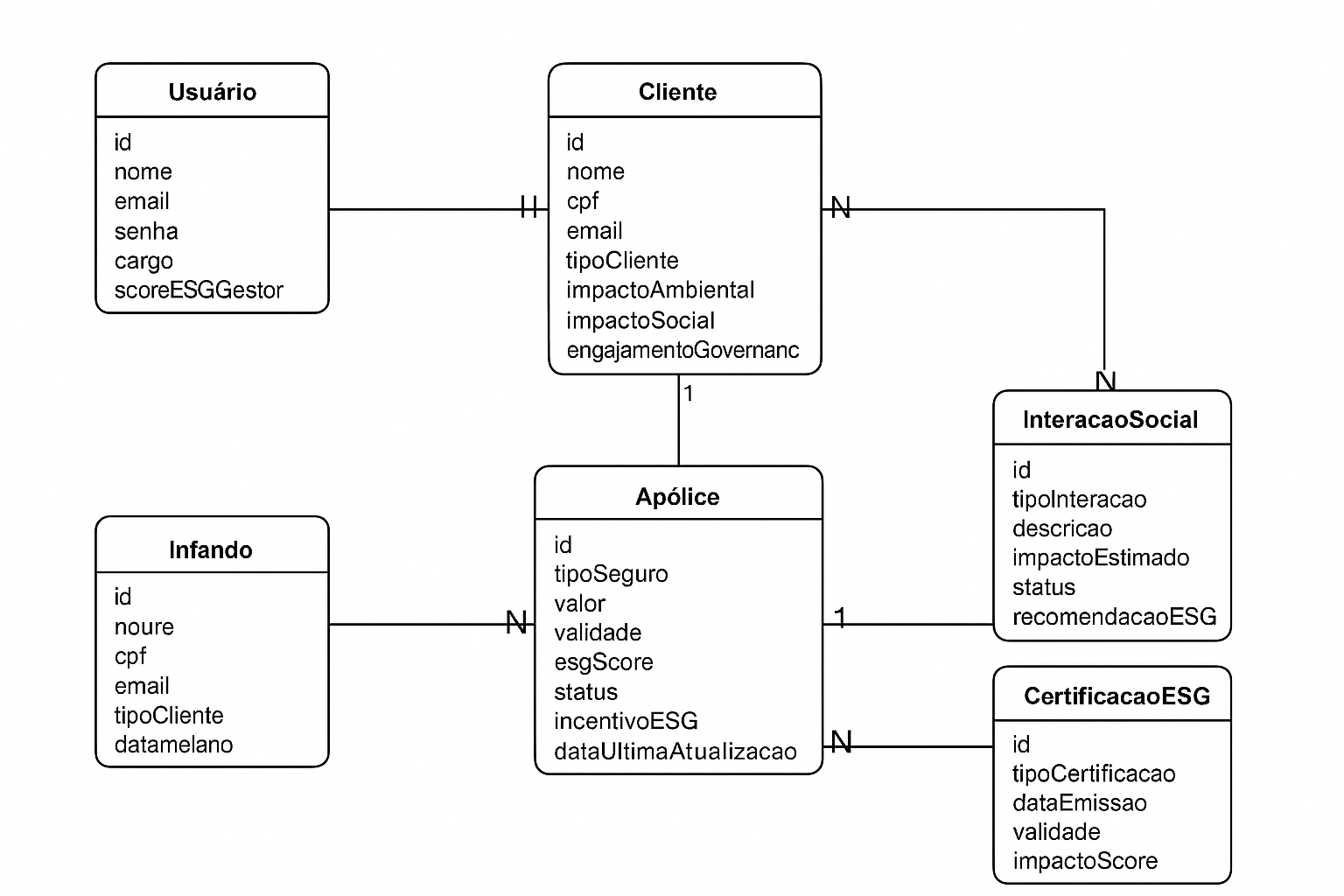
* CRUDs e dashboards simples permitem MVP em 6–8 semanas.
* Mock APIs para programas sociais simplificam integração.
* Cálculo de ESG Score pode ser incremental e testável com poucos dados.

### **Riscos principais e mitigação**

1. **Complexidade de dados ESG** → começar com indicadores básicos e expandir depois.
2. **Engajamento baixo dos clientes** → gamificação e comunicação clara de impacto social.
3. **Conformidade legal** → logs, consentimento e anonimização para LGPD/GDPR.

### **Por que se destacaria numa feira da PwC**

* Une **CRM tradicional + propósito social mensurável**, provocando reflexão e inovação.
* Mostra **tecnologia simples + impacto real**, alinhando ESG e retenção de clientes.
* MVP exequír iniciantes, mas com narrativa estratégica e storytelling corporativo convincente.



### **1️⃣ Interação Social (substitui Oportunidade)**

**O que é na prática:** Essencialmente, são os *contatos, reuniões, campanhas ou engajamentos* que o corretor/gestor faz com o cliente para incentivá-lo a adotar práticas ESG ou melhorar o desempenho ESG da apólice/empresa. Pense como um “registro de relacionamento sustentável”.

**Atributos realistas e aplicáveis:**

* **id** – Identificador único (obvio, básico)
* **tipoInteracao** – Tipo de contato, ex: “reunião presencial”, “webinar ESG”, “email de alerta”, “call consultiva”
* **descricao** – Detalhes da interação, ex: “Apresentada a política de reciclagem corporativa para cliente X”
* **impactoEstimado** – Impacto esperado no ESG do cliente/apólice; poderia ser algo mais realista como pontuação de 1–5 ou porcentagem de melhoria no score ESG
* **status** – “Planejada”, “Concluída”, “Cancelada”, para acompanhar execução
* **recomendacaoESG** – Recomendação prática ao cliente: ex: “Instalar painéis solares”, “Reduzir uso de papel”, “Adotar política de diversidade”

**Como usar na vida real:** Imagine que você é corretor e visita uma empresa cliente. Você registra: “Reunião sobre eficiência energética – cliente se compromete a reduzir consumo em 10%”. Isso aumenta o score ESG do cliente e pode gerar desconto na apólice.

### **2️⃣ Ação Social**

**O que é na prática:** São programas ou atividades **participadas pelo cliente** que têm impacto social ou ambiental. Diferente da interação social (que é *o que você faz para engajar*), a ação social é *o que o cliente efetivamente faz*.

**Atributos mais realistas:**

* **id** – Identificador
* **tipoAcao** – Ex: “Doação”, “Voluntariado”, “Programa de Reciclagem”, “Campanha de inclusão”
* **descricao** – Breve descrição da ação: “Cliente realizou 50h de voluntariado em ONG ambiental”
* **impactoEstimado** – Pode ser pontuação ou efeito mensurável: ex: “Reduziu 200kg de CO₂”, “Ajudou 100 famílias”
* **statusParticipacao** – “Confirmado”, “Em andamento”, “Finalizado”

**Como usar no mercado:** Essas ações podem gerar *incentivos ESG* nas apólices, além de serem usadas em relatórios internos ou relatórios de sustentabilidade de clientes corporativos. É algo que o mercado de seguros já começa a cobrar: mostrar evidências de ESG dos clientes para justificar riscos e benefícios.

### **3️⃣ Certificação ESG**

**O que é na prática:** Selos ou certificações que clientes ou apólices recebem, atestando práticas de sustentabilidade, responsabilidade social ou governança corporativa. É a parte “concreta” que valida todo o engajamento anterior.

**Atributos realistas:**

* **id** – Identificador
* **tipoCertificacao** – Ex: ISO 14001 (ambiental), SA8000 (social), ISO 37001 (governança)
* **dataEmissao** – Quando foi emitida
* **validade** – Data de expiração do selo
* **impactoScore** – Pontuação ou peso que essa certificação dá ao ESG Score da apólice/cliente

**Como usar na vida real:** Se um cliente possui ISO 14001, seu risco ambiental é menor, o que pode reduzir o prêmio do seguro. Além disso, registra-se no sistema como prova de boas práticas, ajudando em relatórios de ESG corporativo.

### **✅ Resumo para deixar palpável:**

| **Entidade** | **Foco** | **Exemplo prático** |
| --- | --- | --- |
| Interação Social | Engajamento/contato | Reunião sobre eficiência energética; envio de e-mail educativo |
| Ação Social | Atividade efetiva do cliente | Participação em voluntariado, programa de reciclagem |
| Certificação ESG | Validação concreta | ISO 14001, selo de diversidade, certificação de governança |

1️⃣ InteraçãoSocial

package com.empresa.crm.model;

import java.time.LocalDateTime;

public class InteracaoSocial {

private Long id;

private String tipoInteracao; // Ex: "Reunião", "Webinar", "Email", "Call"

private String descricao; // Detalhes do que foi feito

private double impactoEstimado; // Pontuação de 1 a 5 ou percentual de melhoria

private String status; // "Planejada", "Concluída", "Cancelada"

private String recomendacaoESG; // Recomendação prática para o cliente

private LocalDateTime dataInteracao;

// Relacionamentos

private Cliente cliente;

private Apolice apolice;

private Usuario corretorGestor;

// Construtores, getters e setters

}

**✅ Observações práticas:**

* O **impactoEstimado** deve ser baseado em regras do sistema (ex: reunião sobre energia = +0,5 no score ambiental do cliente).
* Pode ser usado para alimentar **relatórios de engajamento ESG**.
* Ajuda a justificar descontos ou bônus em apólices.

2️⃣ AcaoSocial

package com.empresa.crm.model;

import java.time.LocalDate;

public class AcaoSocial {

private Long id;

private String tipoAcao; // Ex: "Voluntariado", "Doação", "Programa de Reciclagem"

private String descricao; // Detalhe da ação: "50h de voluntariado em ONG ambiental"

private double impactoEstimado; // Ex: CO2 reduzido, horas de voluntariado, famílias beneficiadas

private String statusParticipacao; // "Confirmado", "Em andamento", "Finalizado"

private LocalDate dataInicio;

private LocalDate dataFim;

// Relacionamentos

private Cliente cliente;

private Apolice apolice;

// Construtores, getters e setters

}

**✅ Observações práticas:**

* O **impactoEstimado** precisa ser mensurável, para poder influenciar diretamente o **ESG Score** do cliente/apólice.
* Pode gerar benefícios comerciais: clientes engajados = menor risco e descontos em seguro.

3️⃣ CertificacaoESG

package com.empresa.crm.model;

import java.time.LocalDate;

public class CertificacaoESG {

private Long id;

private String tipoCertificacao; // Ex: "ISO 14001", "SA8000", "ISO 37001"

private LocalDate dataEmissao;

private LocalDate validade;

private double impactoScore; // Peso que influencia o ESG Score da apólice/cliente

// Relacionamentos

private Cliente cliente;

private Apolice apolice;

// Construtores, getters e setters

}

**✅ Observações práticas:**

* É a prova “concreta” do engajamento ESG do cliente.
* Pode ser usada em cálculo automático do **ESG Score**: se possui ISO 14001, soma +10 pontos no score ambiental.
* Mantém histórico de certificações e validade, permitindo avisos de renovação.

### **💡 Sugestão de ESG Score Dinâmico**

No **Cliente** ou **Apólice**, você pode ter algo assim:

private double esgScore;

// Método para atualizar score automaticamente

public void atualizarESGScore() {

double score = 0;

// Baseado nas Interações Sociais

for(InteracaoSocial interacao : interacoes) {

score += interacao.getImpactoEstimado();

}

// Baseado nas Ações Sociais

for(AcaoSocial acao : acoes) {

score += acao.getImpactoEstimado();

}

// Baseado nas Certificações

for(CertificacaoESG cert : certificacoes) {

score += cert.getImpactoScore();

}

this.esgScore = score;

}

Isso deixa o sistema **realista, mensurável e alinhado com necessidades do mercado de seguros**, sem métricas abstratas demais.

| **Entidade** | **Atributos principais** | **Relacionamentos** | **Observações ESG / Uso no mundo real** |
| --- | --- | --- | --- |

| **Usuário (Corretor/Gestor)** | id, nome, email, senha, cargo, scoreESGGestor | 1–N com Oportunidade | Score interno de performance ESG do gestor; pode indicar engajamento em clientes ESG |
| --- | --- | --- | --- |

| **Cliente** | id, nome, cpf, email, tipoCliente, impactoAmbiental, impactoSocial, engajamentoGovernance | 1–N com Oportunidade  1–N com InteracaoSocial  1–N com AcaoSocial  1–N com CertificacaoESG | Pontuações ou categorias separadas por pilar ESG; base para cálculo de incentivos ou bônus |
| --- | --- | --- | --- |

| **Oportunidade** | id, descricao, status, valorPotencial, dataCriacao, tipoOportunidade, recomendacaoESG | N–1 com Cliente  N–1 com Usuário | Mantém histórico de negócios ou leads; pode registrar oportunidades de engajamento ESG |
| --- | --- | --- | --- |

| **InteracaoSocial** | id, tipoInteracao, descricao, impactoEstimado, status, recomendacaoESG, dataInteracao | N–1 com Cliente  N–1 com Apolice  N–1 com Usuário | Registra ações de engajamento ESG; influência direta no ESG Score do cliente |
| --- | --- | --- | --- |

| **AcaoSocial** | id, tipoAcao, descricao, impactoEstimado, statusParticipacao, dataInicio, dataFim | N–1 com Cliente  N–1 com Apolice | Ações efetivas do cliente (voluntariado, programas de reciclagem); gera pontos de ESG Score |
| --- | --- | --- | --- |

| **CertificacaoESG** | id, tipoCertificacao, dataEmissao, validade, impactoScore | N–1 com Cliente  N–1 com Apolice | Valida práticas ESG; peso direto no ESG Score da apólice ou cliente |
| --- | --- | --- | --- |

| **Apólice (Seguro)** | id, tipoSeguro, valor, validade, esgScore, status, incentivoESG, dataUltimaAtualizacao | N–1 com Cliente  1–N com InteracaoSocial e AcaoSocial e CertificacaoESG | ESG Score dinâmico; incentivos como descontos ou bônus por práticas sustentáveis |
| --- | --- | --- | --- |

### 

### **Como ficam os relacionamentos principais com Oportunidade:**

1. **Cliente 1–N Oportunidade:** Um cliente pode ter várias oportunidades (vendas, upsell, engajamentos ESG).
2. **Oportunidade N–1 Cliente:** Cada oportunidade pertence a um cliente.
3. **Usuário 1–N Oportunidade:** Um corretor/gestor pode ter várias oportunidades.
4. **Oportunidade N–1 Usuário:** Cada oportunidade é gerenciada por um único usuário.

💡 **Observação:** Assim mantemos a estrutura tradicional de CRM, mas complementamos com **atributos ESG** para tornar o sistema mais relevante para decisões de seguro sustentável e engajamento do cliente.

1️⃣ Modelo de ESG Score do Cliente

package com.empresa.crm.model;

import java.util.List;

public class Cliente {

private Long id;

private String nome;

private double esgScore; // Score total do cliente

// Relacionamentos

private List<InteracaoSocial> interacoes;

private List<AcaoSocial> acoes;

private List<CertificacaoESG> certificacoes;

// Construtor

public Cliente(Long id, String nome) {

this.id = id;

this.nome = nome;

this.esgScore = 0.0;

}

// Atualiza o ESG Score com base nas interações, ações e certificações

public void atualizarESGScore() {

double score = 0.0;

// Interações sociais

if(interacoes != null) {

for(InteracaoSocial i : interacoes) {

if("Concluída".equals(i.getStatus())) {

score += i.getImpactoEstimado();

}

}

}

// Ações sociais

if(acoes != null) {

for(AcaoSocial a : acoes) {

if("Finalizado".equals(a.getStatusParticipacao())) {

score += a.getImpactoEstimado();

}

}

}

// Certificações ESG

if(certificacoes != null) {

for(CertificacaoESG c : certificacoes) {

score += c.getImpactoScore();

}

}

this.esgScore = score;

}

// Getters e setters

public double getEsgScore() { return esgScore; }

public void setInteracoes(List<InteracaoSocial> interacoes) { this.interacoes = interacoes; }

public void setAcoes(List<AcaoSocial> acoes) { this.acoes = acoes; }

public void setCertificacoes(List<CertificacaoESG> certificacoes) { this.certificacoes = certificacoes; }

}

2️⃣ Classes de suporte (dados de exemplo)

public class InteracaoSocial {

private String tipoInteracao;

private double impactoEstimado;

private String status;

public InteracaoSocial(String tipo, double impacto, String status) {

this.tipoInteracao = tipo;

this.impactoEstimado = impacto;

this.status = status;

}

public double getImpactoEstimado() { return impactoEstimado; }

public String getStatus() { return status; }

}

public class AcaoSocial {

private String tipoAcao;

private double impactoEstimado;

private String statusParticipacao;

public AcaoSocial(String tipo, double impacto, String status) {

this.tipoAcao = tipo;

this.impactoEstimado = impacto;

this.statusParticipacao = status;

}

public double getImpactoEstimado() { return impactoEstimado; }

public String getStatusParticipacao() { return statusParticipacao; }

}

public class CertificacaoESG {

private String tipoCertificacao;

private double impactoScore;

public CertificacaoESG(String tipo, double impacto) {

this.tipoCertificacao = tipo;

this.impactoScore = impacto;

}

public double getImpactoScore() { return impactoScore; }

}

3️⃣ Testando com dados de exemplo

import java.util.Arrays;

public class TestESG {

public static void main(String[] args) {

Cliente cliente = new Cliente(1L, "Empresa X");

// Criando interações sociais

InteracaoSocial i1 = new InteracaoSocial("Reunião ESG", 2.0, "Concluída");

InteracaoSocial i2 = new InteracaoSocial("Call consultiva", 1.5, "Planejada");

// Criando ações sociais

AcaoSocial a1 = new AcaoSocial("Voluntariado", 3.0, "Finalizado");

AcaoSocial a2 = new AcaoSocial("Programa de reciclagem", 2.5, "Em andamento");

// Criando certificações ESG

CertificacaoESG c1 = new CertificacaoESG("ISO 14001", 5.0);

CertificacaoESG c2 = new CertificacaoESG("SA8000", 4.0);

// Setando listas no cliente

cliente.setInteracoes(Arrays.asList(i1, i2));

cliente.setAcoes(Arrays.asList(a1, a2));

cliente.setCertificacoes(Arrays.asList(c1, c2));

// Calculando ESG Score

cliente.atualizarESGScore();

System.out.println("ESG Score do Cliente: " + cliente.getEsgScore());

}

}

visando safisfação e fidelização dos clientes, aumentando os resultados