|  |  |  |
| --- | --- | --- |
| **Entidade** | **Propósito** | **Atributos Chave** |
| **USUARIO** | Armazena dados dos usuários (comerciantes e consultores/especialistas). | id\_usuario (PK), nome, email, senha\_hash, tipo\_usuario (Comerciante, Consultor), plano (Gratuito, Premium), ra, curso, telefone |
| **NEGOCIO** | Armazena informações sobre o pequeno negócio do comerciante. | id\_negocio (PK), id\_usuario (FK), nome\_negocio, tipo\_negocio (Ex: Bar, Padaria, Mercadinho), descricao\_inicial |
| **DIAGNOSTICO** | Armazena as respostas do formulário guiado e o resultado. | id\_diagnostico (PK), id\_negocio (FK), data\_diagnostico, respostas\_json (para as respostas detalhadas), pontuacao\_digital |
| **ROADMAP** | Armazena o roteiro de digitalização gerado automaticamente. | id\_roadmap (PK), id\_diagnostico (FK), data\_geracao, etapas\_detalhadas (Plano de Ação), status (Em andamento, Concluído) |
| **CONSULTORIA** | Armazena dados dos agendamentos online e histórico do suporte (Chat e Vídeo). | id\_consultoria (PK), id\_usuario\_comerciante (FK), id\_usuario\_consultor (FK), data\_hora\_agendamento, link\_reuniao (para vídeo), historico\_chat, observacoes\_suporte |
| **CONTEUDO\_EDUCATIVO** | Armazena artigos, vídeos e materiais para capacitação. | id\_conteudo (PK), titulo, categoria, url\_acesso, versao\_acesso (Gratuito, Premium) |

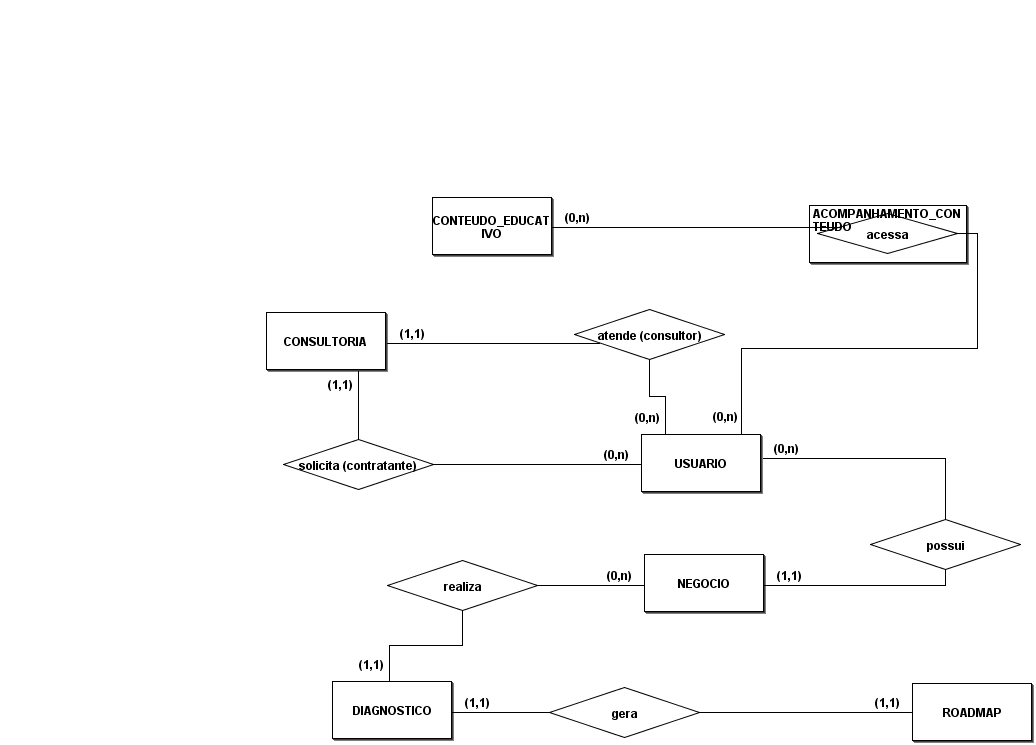
**Estrutura Banco de Dados NextWork, utilizando PostgreSQL**:

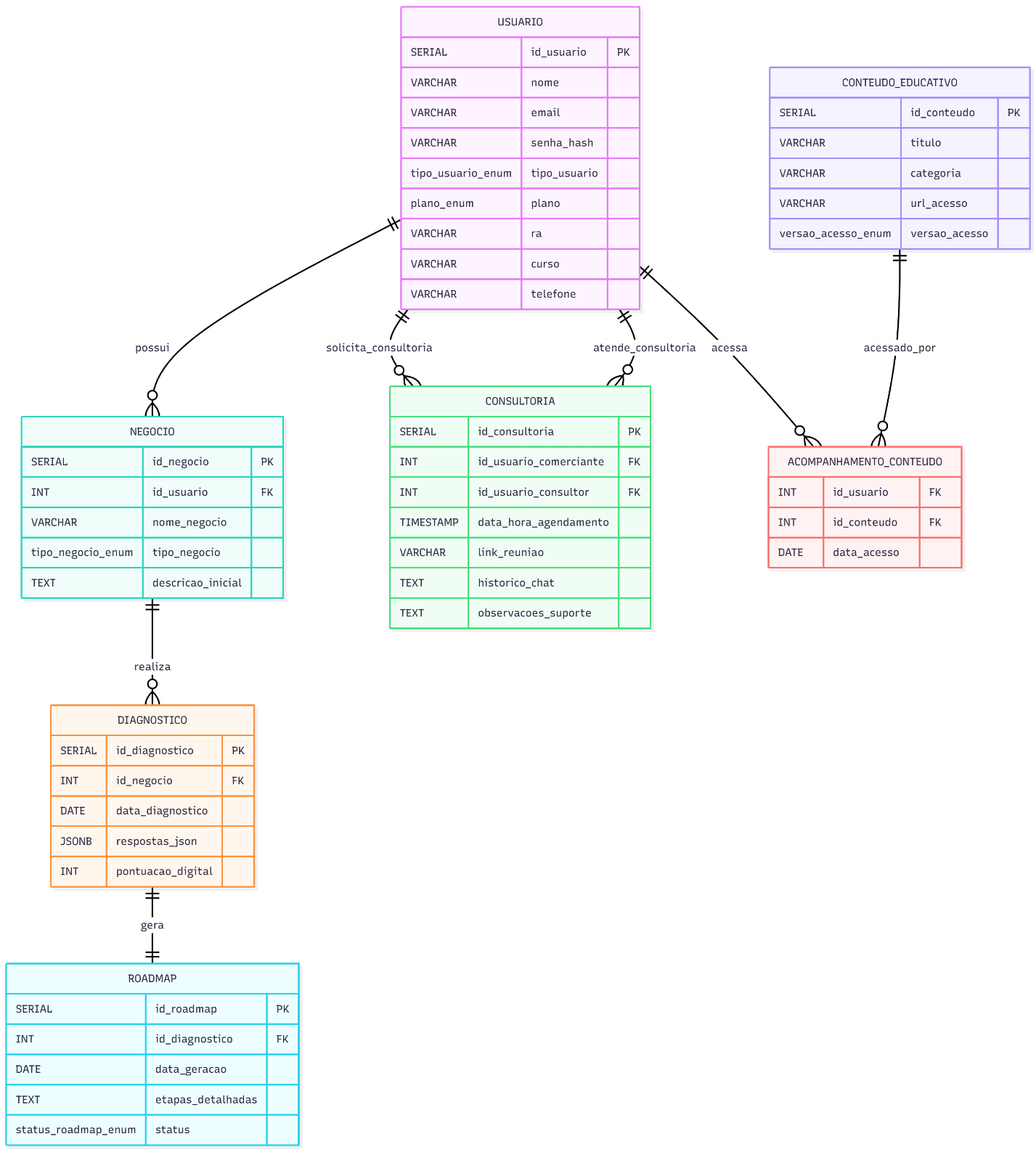
***Legenda:***

* **PK:** Chave Primária (Identificador único da tabela)
* **FK:** Chave Estrangeira (Ligação com outra tabela)
* **JSON:** Tipo de dado para armazenar estruturas de texto complexas (como as respostas do formulário ou os relatórios detalhados).

**Observação sobre a Persistência em Java:** Para um servidor em Java, a forma ideal de interagir com este banco de dados relacional é através de uma tecnologia ORM (Object-Relational Mapping), como **JPA (Java Persistence API)** e sua implementação mais popular, o **Hibernate**. As entidades listadas acima (USUARIO, NEGOCIO, DIAGNOSTICO, etc.) serão mapeadas diretamente para Classes Java com o mesmo nome, usando anotações JPA.

**Modelo Entidade Relacional (MER) Simplificado:**

****

**Tabelas:**

**Código Fonte:**

<https://github.com/OctavioNascimento23/SI-PI4-2025-T1-G05>

**Instruções:**

1. Baixe e instale o pgAdmin 4 (<https://www.pgadmin.org/download/>)
2. Crie um servidor
3. Crie um banco de dados
4. Abra a *Query Tool* (alt+shift+q)
5. Acesse o GitHub do projeto:
6. Execute o conteúdo do arquivo ***CreateTable.sql***
7. Caso der erro, utilizar o ***CreateTable\_wAlterTable.sql***
8. Execute o conteúdo do arquivo ***Samples.sql***