

# RELATÓRIO TÉCNICO DE BANCO DE DADOS

## Projeto Includ.IA - Global Solution 2025

Resumo Executivo:

Documentação técnica da arquitetura de dados, modelagem relacional, lógica PL/SQL e estratégias de integração NoSQL para a plataforma de recrutamento inclusivo Includ.IA.

Autor: Luiz Eduardo Da Silva Pinto (RM555213)

Repositório Oficial: <https://github.com/IncludIA/IncludIA-DataBase>

---

## 1. Introdução e Contexto

O projeto Includ.IA nasce com a missão de mitigar vieses inconscientes nos processos de recrutamento e seleção. Para suportar uma aplicação que utiliza Inteligência Artificial para anonimização de currículos e *matching* de vagas, foi desenhada uma arquitetura de dados híbrida e robusta.

Este relatório detalha a implementação do módulo de persistência, que utiliza **Oracle Database 21c** como fonte da verdade (OLTP) para garantir a integridade das transações financeiras e contratuais, e **MongoDB** como base analítica e de leitura rápida para as funcionalidades de IA e Feed de Vagas.

A solução foi containerizada utilizando **Docker**, garantindo que todo o ambiente (banco relacional, NoSQL e aplicações) possa ser replicado em qualquer infraestrutura com um único comando.

---

## 2. Modelagem de Dados Relacional

A modelagem foi realizada estritamente seguindo a **3<sup>a</sup> Forma Normal (3FN)** para evitar redundâncias e anomalias de atualização. Utilizamos o *Oracle SQL Developer Data Modeler* para a engenharia reversa e geração dos diagramas.

### 2.1. Dicionário de Dados Resumido

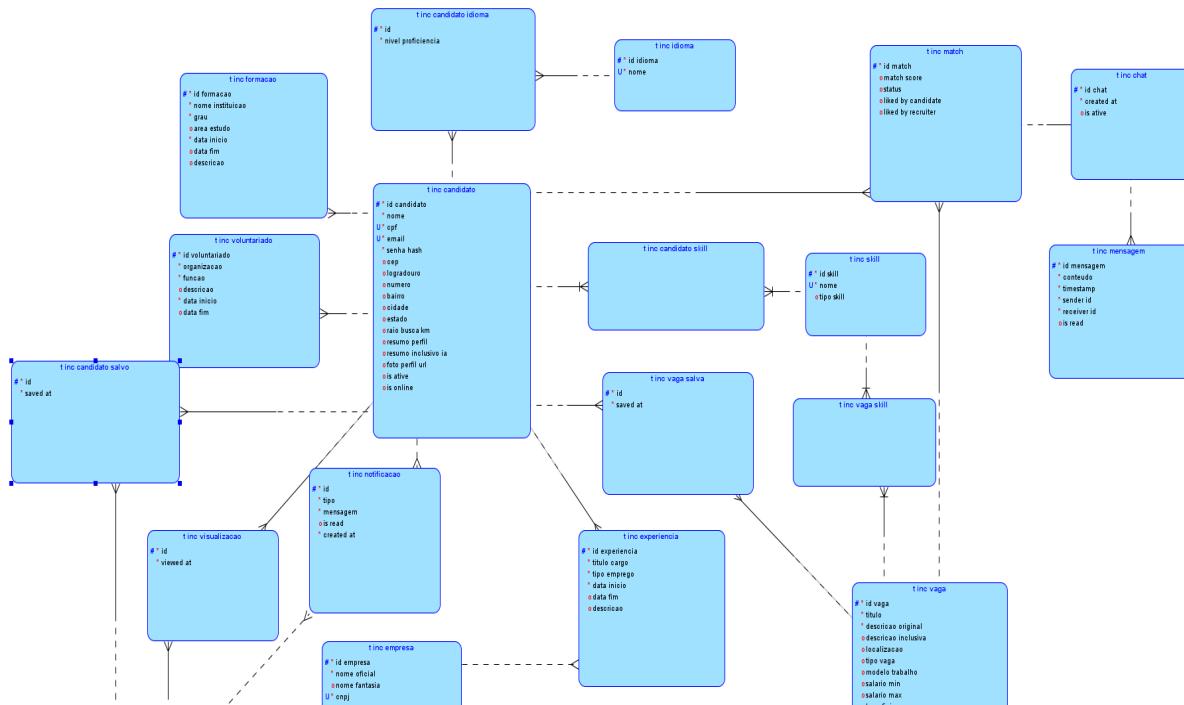
As tabelas foram prefixadas com **T\_INC\_** (Tabela Includ.IA) para organização no schema **SYSTEM**.

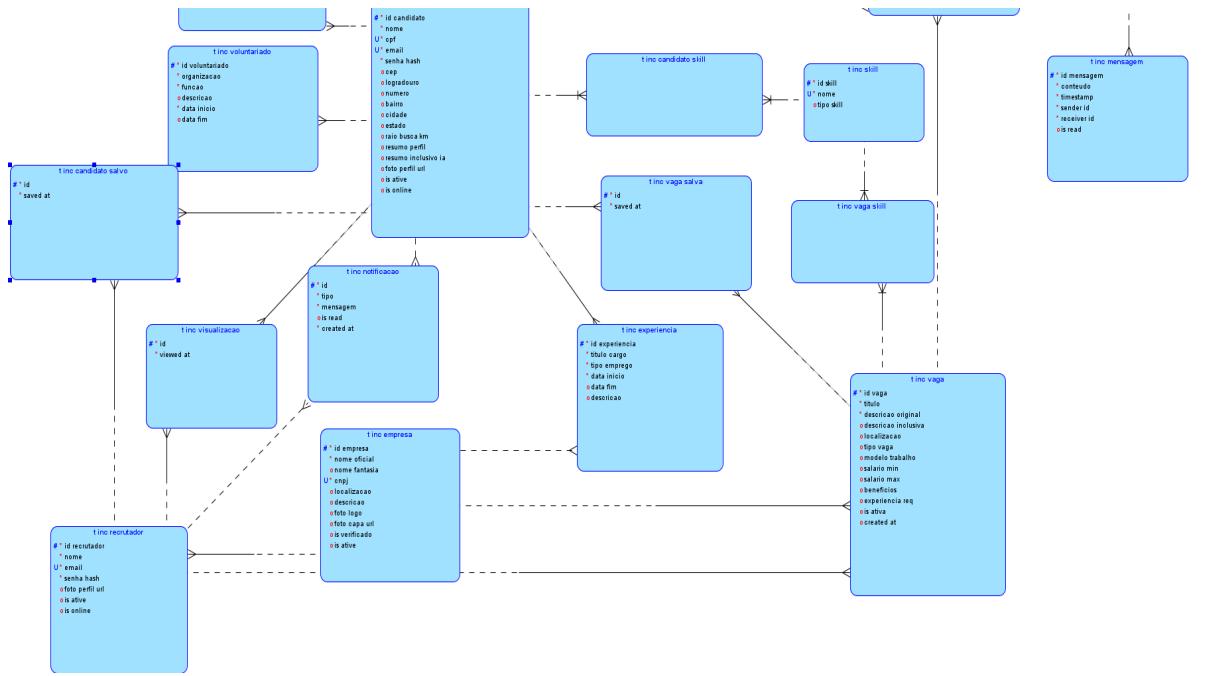
- **T\_INC\_EMPRESA**: Entidade centralizadora das vagas e recrutadores.
- **T\_INC\_CANDIDATO**: Armazena dados sensíveis e o perfil profissional. Possui campos **CLOB** para armazenar textos longos processados pela IA.

- **T\_INC\_VAGA**: Tabela transacional que registra as oportunidades. Contém a **DESCRICAO\_INCLUSIVA** (gerada por IA).
- **T\_INC\_MATCH**: Tabela associativa com atributos que gerencia o estado do "Swipe" (Like/Dislike) entre Candidato e Vaga.
- **T\_INC\_SKILL**: Catálogo de competências (Hard e Soft Skills).

## 2.2. Diagrama Lógico

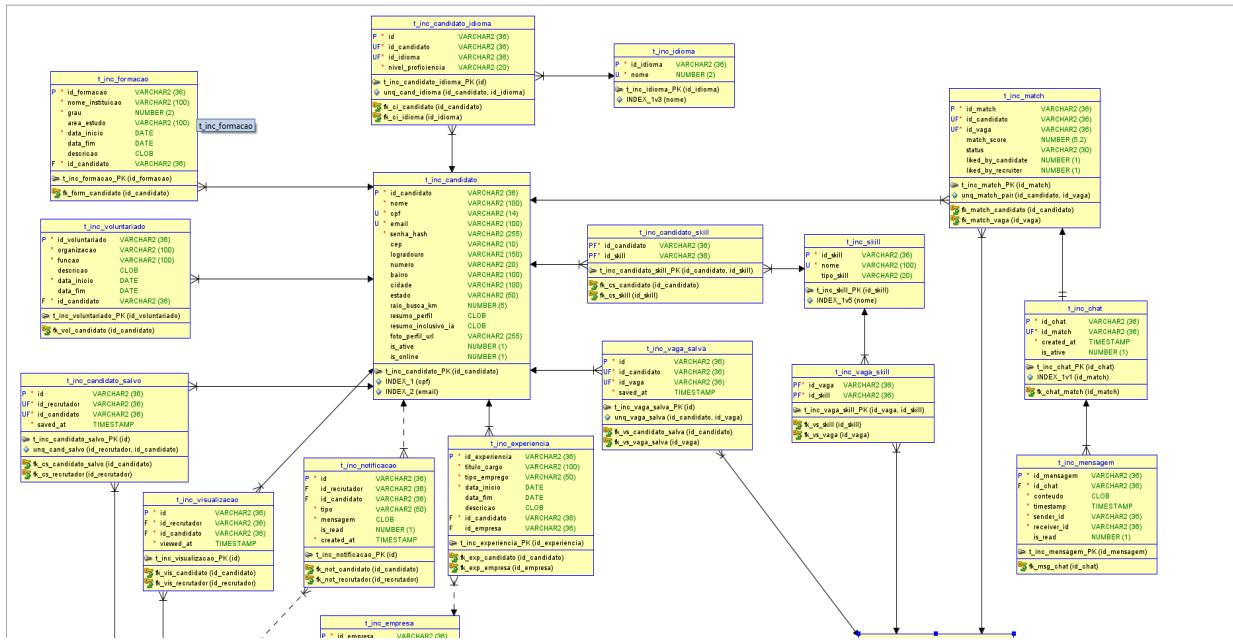
O modelo lógico abaixo representa as entidades de negócio e seus relacionamentos na notação *Information Engineering* (Pé de Galinha).

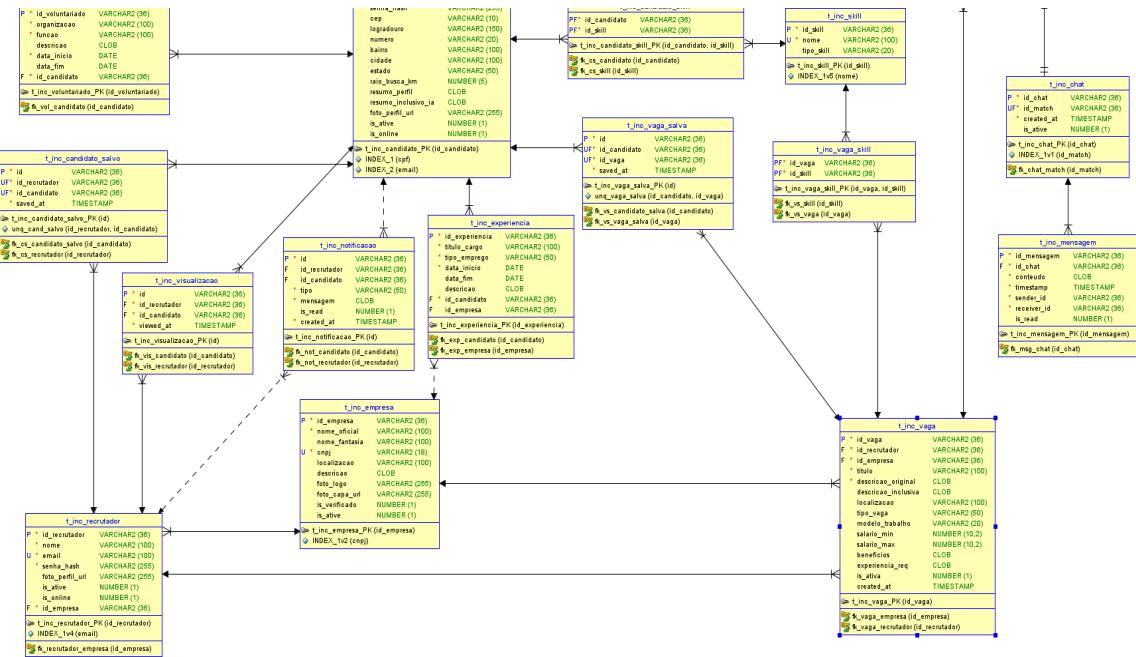




## 2.3. Diagrama Físico

O modelo físico detalha a implementação no Oracle, exibindo tipos de dados (**VARCHAR2**, **NUMBER**, **CLOB**), chaves primárias (**PK**) e estrangeiras (**FK**).





### **3. Engenharia de Dados e PL/SQL**

Para garantir a segurança e a consistência das regras de negócio diretamente no banco de dados, desenvolvemos um pacote PL/SQL robusto.

### 3.1. Pacote **PKG\_INCLUIDA**

Centralizamos todas as procedures e functions neste pacote para modularização e facilidade de manutenção.

## **Funcionalidades Implementadas:**

#### 1. Validação de Dados (FUN\_VALIDAR\_EMAIL):

- Utiliza Expressões Regulares (`REGEXP_LIKE`) para garantir que nenhum e-mail com formato inválido seja inserido no banco, prevenindo erros de comunicação futura.

## 2. Geração de JSON Manual (**FUN\_GERAR\_JSON\_CANDIDATO**):

- **Destaque Técnico:** Desenvolvemos uma função complexa que converte dados relacionais (Candidato + Skills) em um documento JSON formatado.
  - A função realiza a concatenação de strings (**CLOB**) manualmente, iterando sobre cursores, sem depender de funções nativas como **JSON\_OBJECT**. Isso demonstra domínio sobre manipulação de strings e estruturas de repetição em PL/SQL.

### 3. Inserção Segura (PRC\_INSERIR\_CANDIDATO):

- Procedure que encapsula o **INSERT**, realizando a validação de e-mail antes da persistência e tratando exceções de duplicidade (**DUP VAL ON INDEX**).

#### 4. Log de Auditoria (TRG\_AUDIT\_CANDIDATO):

- Implementamos uma Trigger de auditoria que monitora as operações **INSERT**, **UPDATE** e **DELETE** na tabela de candidatos, garantindo rastreabilidade total das alterações de dados sensíveis.
- 

## 4. Integração NoSQL (MongoDB)

A arquitetura do Includ.IA prevê o uso do MongoDB para alta performance no *feed* do aplicativo móvel.

### 4.1. Estratégia de Migração

A integração é feita através da exportação dos dados do Oracle (utilizando a função **FUN\_GERAR\_JSON\_CANDIDATO** descrita anteriormente) e importação na coleção **candidatos** do MongoDB.

Isso permite que o aplicativo móvel consulte perfis completos (Dados + Skills) em uma única leitura de documento, sem a necessidade de *joins* custosos em tempo real.

#### Exemplo de Documento Gerado:

```
JSON
{
  "id": "4434c766...",
  "nome": "João da Silva",
  "email": "joao@email.com",
  "resumo": "Desenvolvedor Java...",
  "skills": ["Java Advanced", "React Native"],
  "origem": "Oracle Database"
}
```

---

## 5. Conclusão

O projeto de banco de dados do Includ.IA atende a todos os requisitos funcionais e não funcionais propostos na Global Solution. A combinação de **Oracle** para integridade transacional com **MongoDB** para flexibilidade, unida à lógica embarcada em **PL/SQL**, resultou em um backend sólido, seguro e preparado para escalar.

A infraestrutura automatizada via Docker garante que o ambiente de desenvolvimento e produção sejam idênticos, facilitando o processo de DevOps e entrega contínua.