

DOMAIN DRIVEN DESIGN USING JAVA

Faculdade de Informática e Administração Paulista

CHALLENGE 2025

2º Semestre / Sprint 4

Equipe: SkillShift

Integrantes:

Fabício Henrique Pereira — RM 563237

Leonardo José Pereira — RM 563065

Pedro Henrique de Oliveira — RM 562312

São Paulo, 2025

SUMÁRIO

1. Objetivo e Escopo do Projeto
2. Descrição da Solução
3. Funcionalidades Principais
4. Funcionalidades Implementadas (Destaques)
5. Endpoints da API REST
6. Diagrama de Classes
7. Modelo Entidade-Relacionamento (MER)
8. Scripts SQL e Dados
9. Prints da Solução

1. Objetivo e Escopo do Projeto

A SkillShift.AI responde ao desafio “Futuro do Trabalho — Requalificação Profissional com IA” ao:

- Identificar risco de automação e obsolescência para cada usuário.
- Propor ações de requalificação personalizadas e priorizadas.
- Recomendar trilhas de aprendizagem baseadas em IA, integrando cursos internos e externos.
- Atender usuários finais e empresas (B2C + B2B), incluindo vínculo usuário-empresa e relatórios corporativos.

Escopo contemplado:

- API Java Quarkus com RBAC (USER, EMPRESA, ADMIN), autenticação e CRUD completo de entidades.
- Modelo relacional Oracle com scripts DDL/DML/DQL e massa de teste.
- Integração real com a IA em Flask publicada em Render.
- Regras de negócio como validação de permissões, geração de recomendações, logging e auditoria básica.

2. Descrição da Solução

A solução possui três camadas:

1. **Backend Java/Quarkus:** núcleo operacional, responsável por autenticação, controle de acesso, CRUD (usuários, cursos, empresas, vínculos, recomendações) e orquestração do fluxo de recomendações.
2. **Banco Oracle:** modelo relacional com tabelas `t_skillshift_*`, constraints, scripts de carga e consultas analíticas.
3. **API de IA em Flask/Python:** hospedada em <https://skillshift-ai-platform.onrender.com>, treinada com o dataset `Data_final.csv` usando Random Forest (macro-área profissional) e K-Means (cluster/recomendações).

Fluxo /recomendacoes/gerar:

1. API Java busca o usuário no Oracle.
2. Monta JSON com features (OP_score, CO_score, EX_score etc.).
3. Chama /cluster-profile da IA Flask.
4. Recebe cluster + lista de cursos recomendados.
5. Mapeia nomes retornados para IDs reais (`t_skillshift_curso` e `t_skillshift_curso_alias`).
6. Calcula score e cluster.
7. Persiste na tabela `t_skillshift_recomendacao` e registra log em `t_skillshift_recomendacao_ia_log`.
8. Retorna ao front/chatbot as recomendações persistidas.

A IA fica publicada no Render; o backend pode rodar em qualquer provedor (Render, Railway, AWS, local). Durante o desenvolvimento, Java e Flask podem executar localmente usando as mesmas URLs.

3. Funcionalidades Principais

- CRUD completo de usuários, cursos, empresas e vínculos usuário-empresa.
- Autenticação por email/senha e RBAC simples (tipo_perfil).
- Geração de recomendações via IA (REST Client).
- Persistência dos resultados com payload, cluster e score.
- Histórico e filtros de recomendações por usuário.
- Integração Oracle via JDBC com ConnectionFactory.

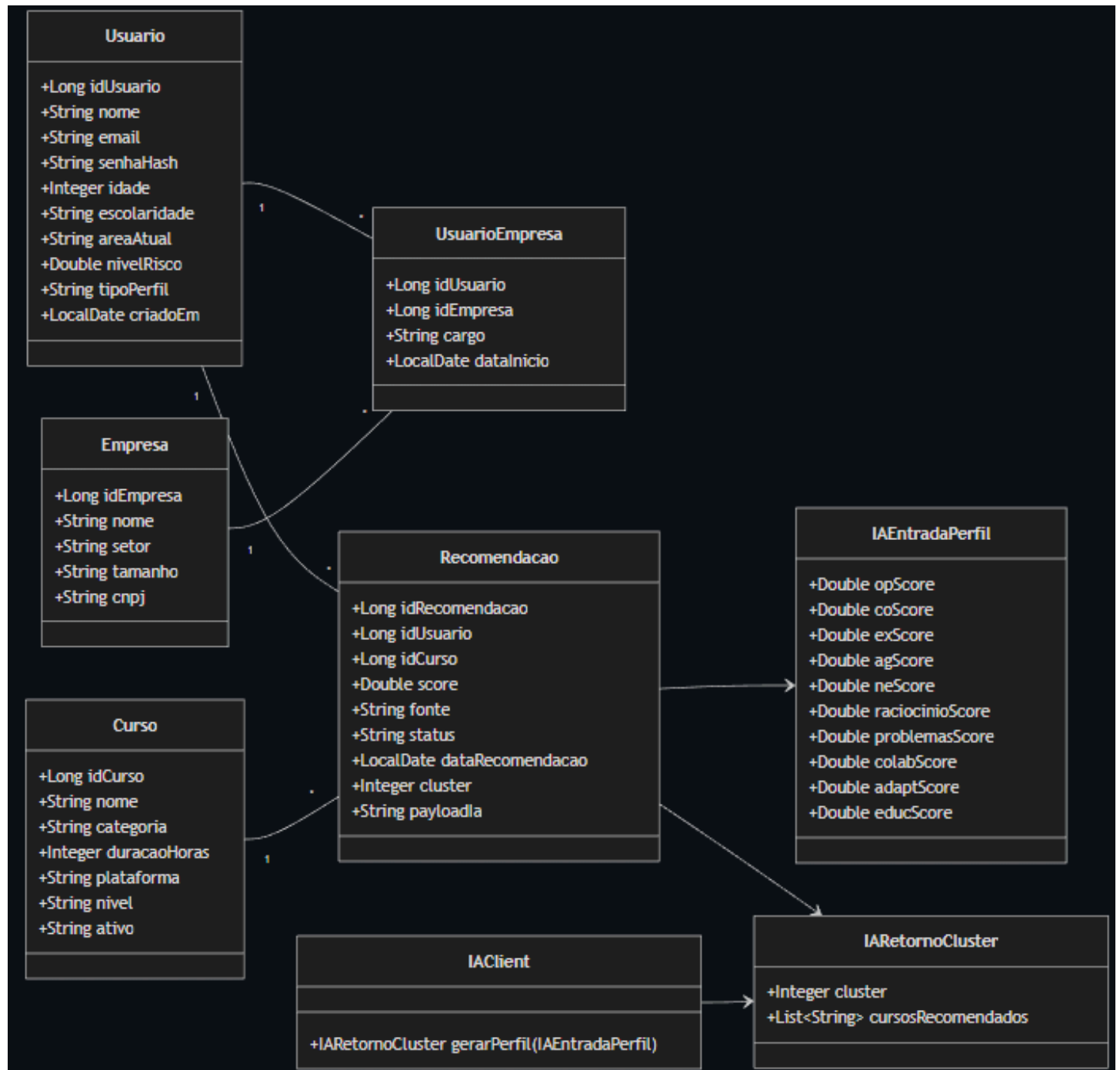
4. Funcionalidades Implementadas (Destaques)

- API Java operante (Resources → BO → DAO) com validações.
- Modelo relacional Oracle completo + scripts de carga em blocos.
- IA hospedada/pronta e integração REST implementada.
- Armazenamento real das recomendações e logs de IA.
- Filtro de score e normalização.
- Fluxo de regras de negócio completo em /recomendacoes/gerar.

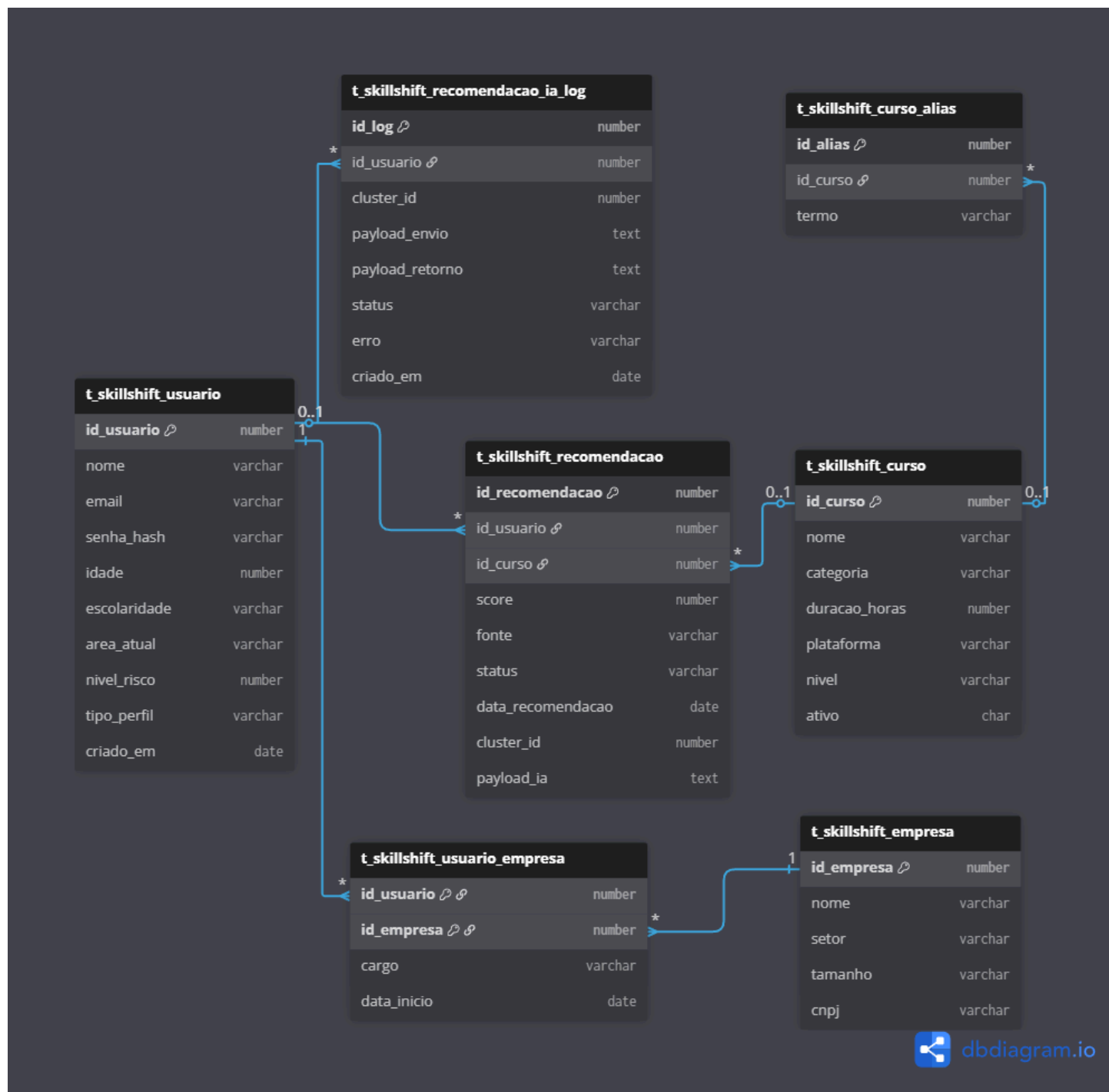
5. Endpoints da API REST

URI	Método	Descrição	Corpo Req.	Corpo Resp.	Códigos
/auth/login	POST	Autentica usuário	{ email, senha }	{ token, perfil }	200, 400, 401, 500
/usuarios	GET	Lista usuários	—	Array de usuários	200, 500
/usuarios/{id}	GET	Detalha usuário	—	Usuário	200, 404, 500
/usuarios	POST	Cria usuário	JSON com campos obrigatórios	Usuário criado	201, 400, 409, 500
/usuarios/{id}	PUT	Atualiza usuário	JSON	Usuário atualizado	200, 400, 404, 500
/usuarios/{id}	DELETE	Remove usuário	—	—	204, 404, 500
/cursos	CRUD	Gestão dos cursos	JSON	Curso/Lista	200, 201, 204, 400, 404, 500
/empresas	CRUD	Gestão de empresas	JSON	Empresa/Lista	200, 201, 204, 400, 404, 409, 500
/recomendacoes?usuarioId=X	GET	Lista recomendações com filtro	—	Lista de recomendações	200, 500
/recomendacoes/gerar?usuarioId=X	POST	Gera recomendações via IA	—	Lista criada	201, 400, 404, 503, 500
/cluster-profile (IA Flask)	POST	Consome IA externa	JSON com features	{ cluster, cursosRecomendados[] }	200, 400, 500

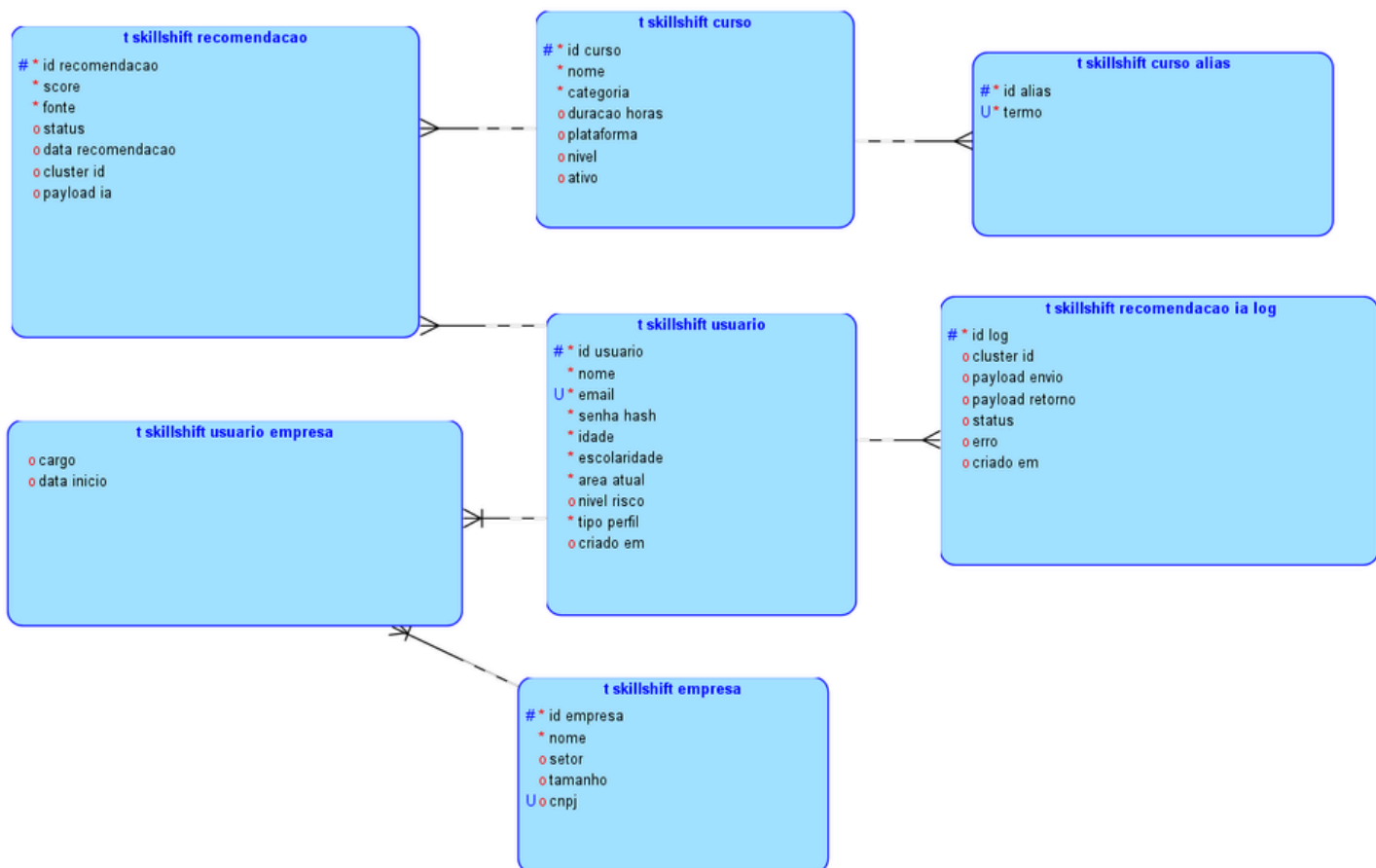
6. Diagrama de Classes



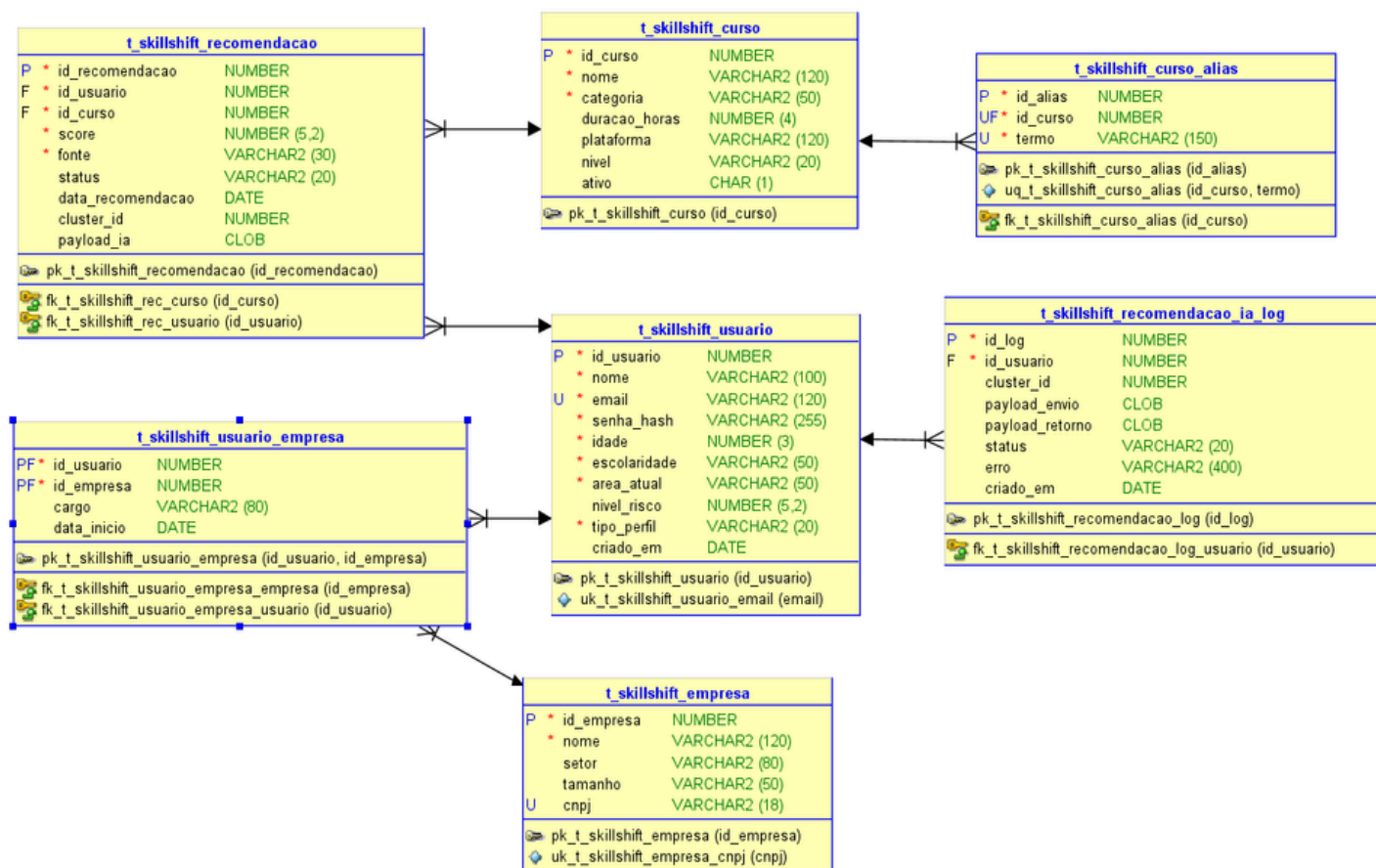
7. Modelo Entidade-Relacionamento (MER)



Modelo Lógico



Modelo Relacional



8. Scripts SQL e Dados

Localizados em src/db/ddl/:

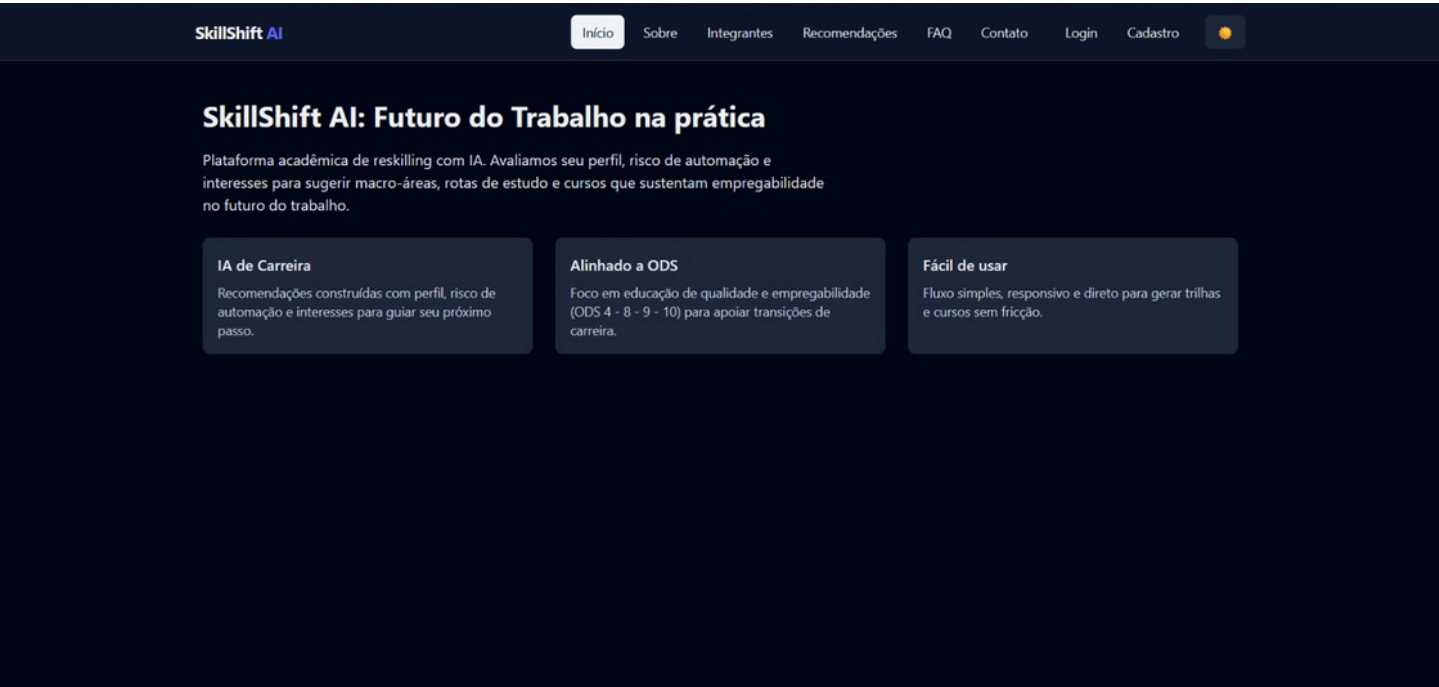
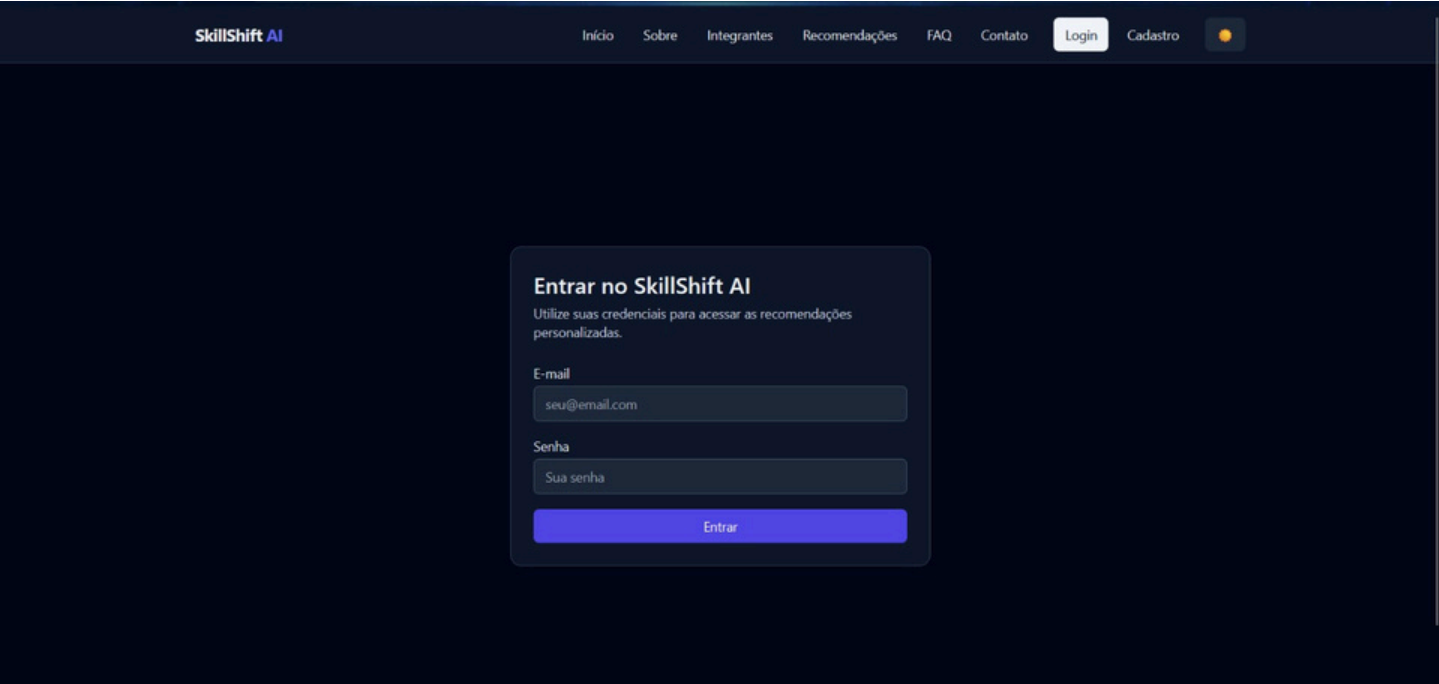
- create_tables.sql: recria todas as tabelas t_skillshift_* (PK, FK, UK, CK, cluster_id, payload_ia).
- drop_tables.sql: drop ordenado para reset.
- carga_dados.sql + blocos carga_dados_0X_*: massa de teste (16 usuários, >100 cursos, empresas, recomendações, aliases).
- ../dql/consultas.sql: consultas analíticas para BI.

Estruturas complementares:

- t_skillshift_curso_alias: mapear nomes retornados pela IA.
- t_skillshift_recomendacao_ia_log: log de payloads e status do serviço IA.

Execute drop_tables.sql → create_tables.sql → carga_dados.sql antes de subir a API para garantir aderência entre banco e código.

9. Prints da Solução



Meu perfil

Dados da sua conta e histórico de recomendações.

Sair

Dados básicos

Nome: teste

E-mail: teste@fiap.com

Alterar senha

Informe a senha atual e defina uma nova senha.

Senha atual

Nova senha

Confirmar nova senha

Atualizar senha

Últimas recomendações

Histórico salvo das suas consultas recentes.

23/11/2025, 15:24:46

Macro-área: Human

- Trilha: Fundamentos de programação
- Curso: Python para Iniciantes
- Curso: Lógica de Programação

