

Relatório do Projeto de Bases de Dados

Base de Dados de um Centro de Emprego

Análise de Requisitos	2
Diagrama Entidade Relação	3
Modelo Relacional	4
Normalização	5
SQL DDL	5
SQL Programming	7
Índices	7
Triggers	7
Stored Procedures	7
User Defined Functions	8
Execução do Projeto	9

Gonçalo Machado nMec 98359

Vítor Dias nMec 98396

Análise de Requisitos

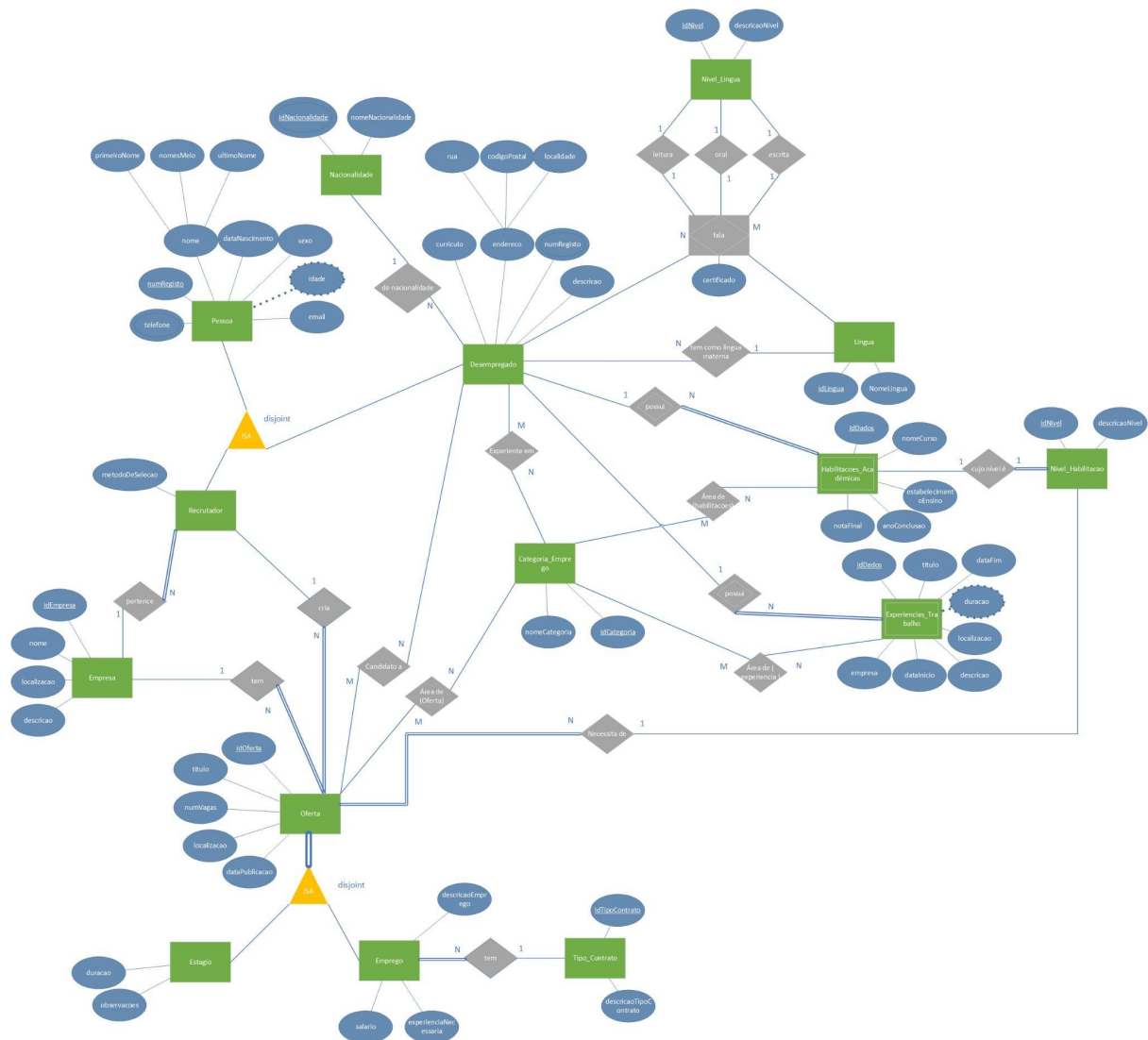
Este projeto tem como objetivo criar uma base de dados para um Centro de Emprego, assim como uma interface que permite a interação com a base de dados.

Antes de realizarmos o desenho conceptual, fizemos alguma pesquisa sobre como um centro de emprego funciona e que entidades seriam utilizadas.

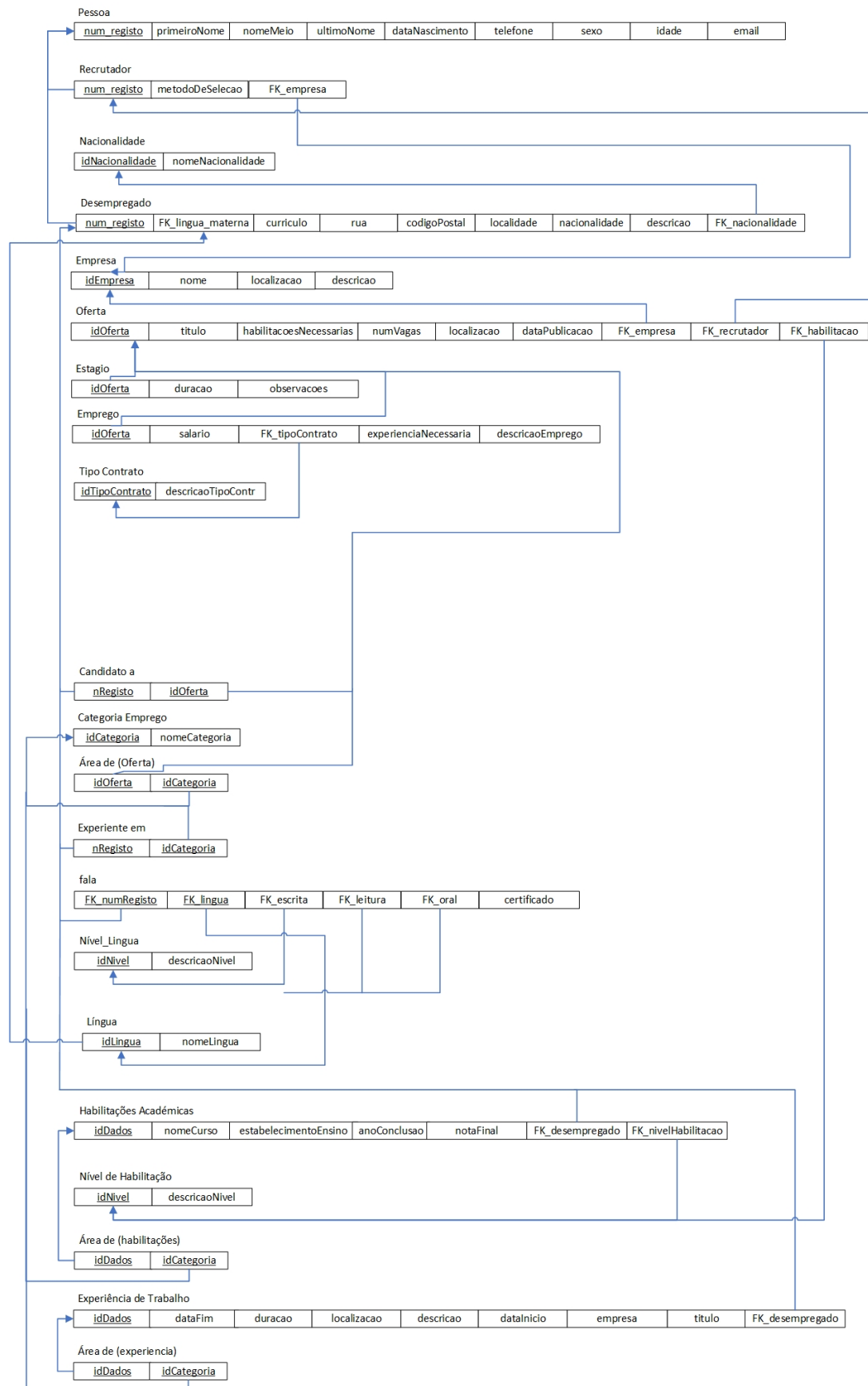
Após a pesquisa, chegámos à seguinte análise de requisitos:

- Uma Pessoa é constituída por um nome, uma data de nascimento, um género, um email, uma idade, um número de registo e um ou mais telefones
- Uma Pessoa pode ser ou um Recrutador ou um Desempregado
- Um Recrutador também é caracterizado por um método de seleção (texto a descrever a abordagem do recrutador das entrevistas)
- Um Desempregado também é caracterizado por uma língua materna, um currículo, uma nacionalidade, uma descrição, um endereço (composto por rua, código postal e localidade)
- Um Desempregado tem 0 ou mais Habilitações Linguísticas, 0 ou mais Experiências de Trabalho, 0 ou mais Categorias de Emprego e 1 ou mais Habilitações Académicas
- Uma Habilitação Linguística é caracterizada por uma língua, um certificado, um nível de oralidade, um nível de escrita, um nível de leitura e 0 ou mais Categorias de Emprego
- Uma Habilitação Académica é composta por nível de Habilitação, o nome do curso, estabelecimento de ensino, ano de conclusão, nota final e 0 ou mais Categorias de Emprego
- Uma Experiência de Trabalho é caracterizada por uma empresa, data início, data fim, duração, localização, descrição e 0 ou mais Categorias de Emprego
- Uma Categoria de Emprego é constituída por o id da categoria e o nome da categoria
- Uma Empresa é caracterizada por um id, um nome, uma localização e uma descrição
- Uma Empresa pode ter 0 ou mais Recrutadores, e um Recrutador tem que pertencer a uma Empresa
- Uma Oferta é caracterizada por um id, um título, as habilitações necessárias, o número de vagas, a localização, a data de publicação, 0 ou mais Categorias de Emprego e pode ter no máximo 1 Teste
- Uma Oferta pode ser ou um Estágio ou um Emprego
- Um Estágio tem ainda uma duração e observações
- Um Emprego tem ainda um salário, um tipo de contrato, a experiência necessária e uma descrição do emprego
- Um Recrutador pode criar 0 ou mais Ofertas
- Uma Empresa pode ter 0 ou mais Ofertas
- Um Desempregado pode se candidatar a 0 ou mais Ofertas e uma Oferta pode ter 0 ou mais Desempregados a candidatarem-se

Diagrama Entidade Relação



Modelo Relacional



Normalização

Foi realizada a normalização de todas as tabelas e depois de algumas alterações, todas as tabelas ficaram na forma BCNF

SQL DDL

Quanto à criação da Base de Dados em SQL, o primeiro passo foi criar um schema “projeto” para podermos agrupar todas as tabelas e outros elementos e separar de outras tabelas existentes.

De seguida, e utilizando o modelo relacional, criámos as tabelas com todos os atributos e os respetivos tipos. Os tipos utilizados foram :

- int - Para id's;
- smallint - Para números e para id's de tabelas que sabíamos que não iriam ter muitos tuplos, e portanto poucos id's;
- varbinary()- Para ficheiros;
- varchar() , nvarchar(), nchar(), char() - Para texto;
- date - Para datas;

Foram utilizados também Constraints durante a criação das tabelas, tais como:

- Primary key - Para definir as chaves primárias;
- Foreign key - Para definir as chaves estrangeiras;
- Unique - Para atributos que, não sendo a chave primária, eram chaves candidatos e por isso seriam únicos;
- Check - Para verificar se os atributos inseridos estariam no formato correto;

Ainda foi utilizado Identity para os id's dos tuplos serem criados automaticamente, assim como atributos calculados automaticamente a partir de outros atributos, como a idade a partir da data de nascimento, ou a duração de algo a partir da sua data de início e de fim.

SQL DML

Os comandos de DML usados nos forms foram sobretudo comandos simples e que não necessitavam monitorização. Todos os restantes (comandos mais complexos) foram aplicados em Stored Procedures, Triggers e User Defined Functions, que serão vistos mais à frente.

Foram utilizados INSERT's, UPDATE's e DELETE's para permitir a inserção, atualização e eliminação de dados das tabelas da Base de Dados.

Foram também utilizadas query's para recolher dados da BD. Estas envolvem diversas características aplicáveis às mesmas como :

- SELECT FROM, para obter dados de uma tabela;
- WHERE, para selecionar subconjuntos de tuplos das tabelas;
- Renomeação de atributos e tabelas;
- Junção de tabelas (JOIN's), para obter todos os dados de um Recrutador, por exemplo;
- Comparação de Strings para implementar a busca por nome;
- ORDER BY, para ordenar os tuplos pela ordem que o utilizador pretende.

SQL Programming

Índices

Para termos uma melhor performance da base de dados, foram criados quatro índices:

- IxEmail_Pessoa - Para o email de uma Pessoa
- IxIdEmpresa_Oferta - Para o idEmpresa de uma Oferta
- IxIdRecrutador_Oferta - Para o idRecrutador de uma Oferta
- IxIdEmpresa_Recrutador - Para o idEmpresa de um Recrutador

Triggers

Foram criados os seguintes triggers, todos eles são triggers “Instead of” sobre a ação de *delete*:

- deleteEstagio;
- deleteEmprego;
- deleteEmpresa;
- deleteDesempregado;
- deleteExperienciaTrabalho;
- deleteRecrutador;
- deleteOferta;
- deleteHabilitacaoAcademica;

Estes triggers foram criados pois existem muitas tabelas que tem como chaves primárias chaves primárias de outras tabelas, sendo que antes de apagar um tuplo da “tabela mãe”, temos que apagar os tuplos dependentes da “tabela filha”.

Stored Procedures

Foram criados stored procedures maioritariamente para facilitar a inserção e atualização de dados nas tabelas:

- insertPessoa;
- updatePessoa;
- insertDesempregado;
- updateDesempregado ;
- insertDesempregado;
- updateRecrutador;
- insertEmpresa;
- insertExperienciaTrabalho;
- updateExperienciaTrabalho;
- insertOferta;
- updateOferta;
- insertEmprego;
- updateEmprego;
- insertEstagio;
- updateEstagio;
- insertHabilitacaoAcademica;
- updateHabilitacaoAcademica;
- insertUpdateFala;
- insertCandidatura;
- updateCandidatura;

Existem alguns Stored Procedure que possuem um Nested Stored Procedure, ou seja, dentro deles é executado outro Stored Procedure. Um exemplo é o *insertDesempregado* que dentro dele executa o SP *insertPessoa*.

User Defined Functions

De forma a executar query's mais complexas de recolha de dados foram usadas as seguintes UDF's:

- `RecolherCandidatosDeUmaOferta` - recolher os candidatos que se registaram a uma determinada oferta que foi selecionada;
- `RecolherCandidaturasDeDesempregado` - recolher todas as candidaturas que um determinado utilizador realizou;
- `RecolherEmpregosDeEmpresa` - recolher todos os empregos que uma empresa tem registados;
- `RecolherEmpregosdeRecrutador` - recolher todos os empregos que um recrutador registou;
- `RecolherEstagiosDeEmpresa` - recolher todos os estágios que uma empresa tem registados;
- `RecolherEstagiosdeRecrutador` - recolher todos os estágios que um recrutador registou;
- `RecolherInfoDeOferta` - recolher informações detalhadas de uma oferta;
- `RecolherRecrutadores` - recolher informações detalhadas de um recrutador.

Estas UDF's contêm Subconsultas correlacionadas e diversos JOIN's, sendo então usadas para as executar consoante os dados passados pelos formulários.

Execução do Projeto

Para conectar o programa a uma base de dados é necessário alterar a classe "Connection", na instrução `SqlConnection` da linha 24.