

# **RELATÓRIO TRABALHO PRÁTICO FINAL**

## **BASE DE DADOS 2019/2020**

Rui Miguel Silva Oliveira 89216

Pedro Miguel Miranda Gonçalves 88859

## ANÁLISE DE REQUISITOS

No âmbito da disciplina de Base de Dados, foi proposto criar um sistema para gestão de centros comerciais. Assim, pensou-se em criar uma interface onde, de uma maneira simples mas funcional, se pudesse interagir com vários centros comerciais e suas respetivas entidades.

Para controlar toda esta informação, pretende-se a partir de formulários gráficos em Windows Forms (C#) com ligação a uma base de dados criar o sistema acima descrito.

Deste modo, o utilizador poderá acrescentar, editar, remover e procurar entidades e seus constituintes, de uma maneira dinâmica, podendo assim interagir com toda a informação de vários centros comerciais.

O **centro** é caracterizado por **departamentos**. Estes são geridos pelo respetivo **responsável do departamento**.

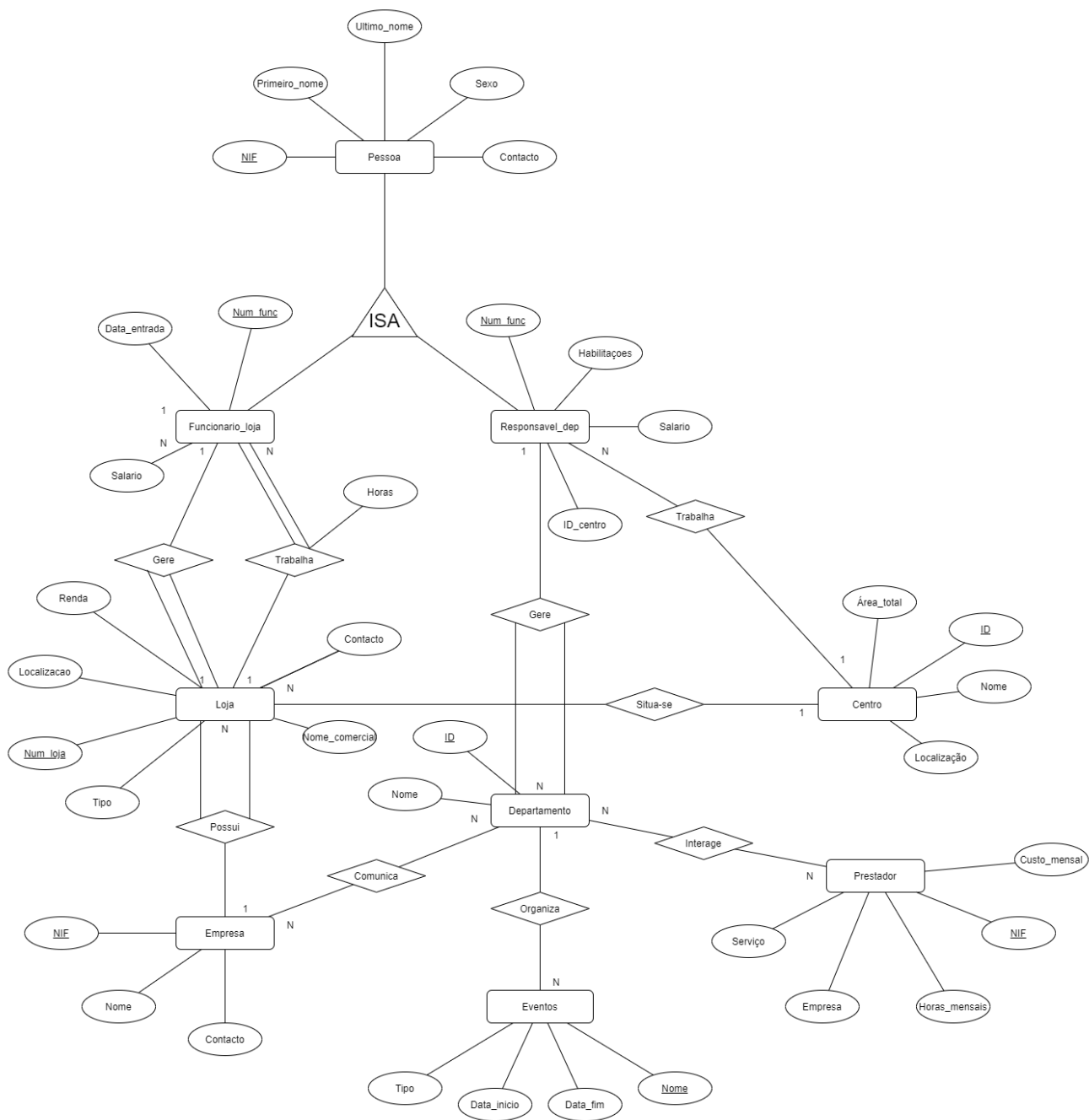
Cada **departamento** está encarregue de gerir diferentes atividades.

O departamento de **Marketing** é responsável pela gestão dos **eventos** (Natal, Páscoa...).

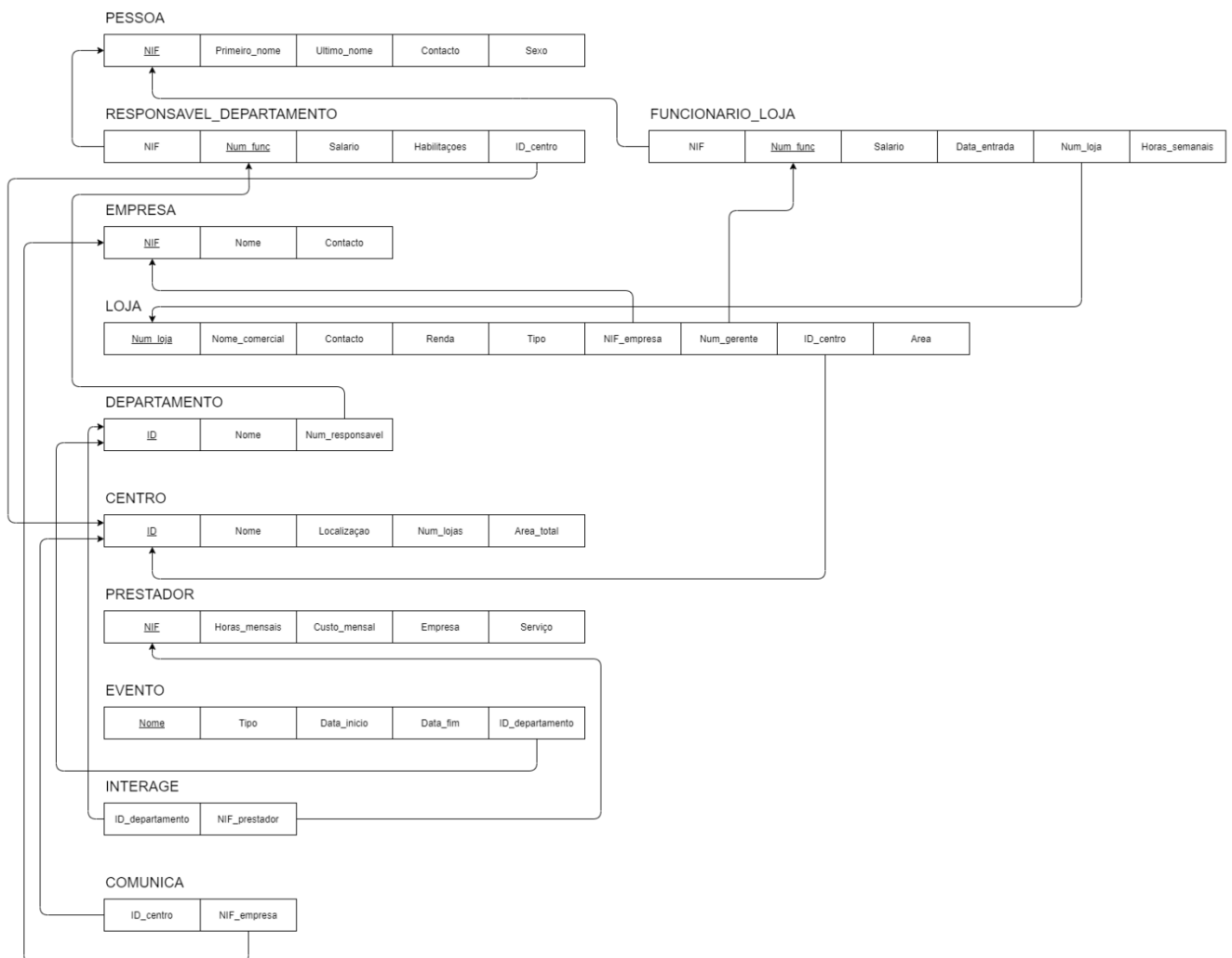
O departamento de **Operações** lida com os **prestadores** (água, eletricidade, gás etc. do centro e das respetivas lojas).

O departamento **Comercial** é responsável pela gestão das empresas que por sua vez possuem as **lojas** do centro compostas por diversos **funcionários**.

## DIAGRAMA ENTIDADE RELAÇÃO



## ESQUEMA RELACIONAL



## SQL DDL ASSOCIADOS À DEFINIÇÃO DA ESTRUTURA DA BD EM SQL SERVER

Em conformidade com as entidades e seus respectivos atributos descritos no DER e ER criámos as diversas tabelas com as respetivas chaves primárias e estrangeiras.

Um caso particular resulta da inserção do “funcionario\_loja” antes da “loja”.

Visto que o funcionário necessita do número da loja, temos de adicionar um *constraint* apenas para introduzir o número de loja nos funcionários, alterando assim a tabela após a sua criação.

Todas as tabelas estão disponíveis no ficheiro SQL\_DDL.sql.

## SQL DML ASSOCIADAS A CADA FORMULÁRIO GRÁFICO

No que toca aos dados introduzidos na base de dados, todos estes foram introduzidos com critério e estão disponíveis no ficheiro SQL\_DML.sql.

## NORMALIZAÇÃO

Todas as relações foram normalizadas até à BCNF. Estas foram criadas já com o objetivo de reduzir a redundância de dados, aumentar a integridade de dados e o desempenho.

Deste modo, só a entidade departamento é que mais tarde teve de ser passada de 3FN para BCNF visto que não existia necessidade de existir o ID do centro, já que poderia ser obtido através do responsável do departamento.

## TRIGGERS

O projeto possui dois *triggers* (trgDelDepartamento e trgLojaAlterada) que se encontram no ficheiro Triggers.sql.

O primeiro tem como objetivo alertar o utilizador que não pode remover nenhum dos três departamentos principais (Marketing, Operações e Comercial). Mais tarde chegámos à conclusão de que era mais eficaz utilizar um *stored procedure* para esse fim.

Com o segundo pretende-se indicar ao utilizador que não pode colocar um funcionário em *part-time* como gerente da loja.

## STORED PROCEDURES

Criamos vários SP com a finalidade de obter os dados de cada entidade que estão na base de dados. Também criámos alguns com o objetivo de remover várias entidades, entre elas o spDelCentro que permite remover um centro comercial bem como todas as entidades que dele dependem.

Todos os SP estão todos disponíveis no ficheiro StoredProcedures.sql.

## UDF

No caso de necessitarmos de um valor de retorno optámos por criar funções de modo a que estas devolvam o valor pretendido.

Por exemplo, a UDF `fnGetNextNumLoja` permite gerar dinamicamente o número da próxima loja a ser inserida no centro. A maior parte das funções tem como principal objetivo facilitar a interação com a interface tornando-a mais dinâmica.

Todas as UDF estão disponíveis no ficheiro `UDFs.sql`.

## FUNCIONALIDADES

Esta disponível um vídeo (`VideoDemostrador.mp4`) com o propósito de mostrar como se interage com a interface e demonstrar as suas funcionalidades.

## NOTAS

Nem toda a interação com a base de dados é feita a partir de SP, UDF e *Triggers*, os comandos mais simples foram declarados no próprio ficheiro (`.cs`).