

UNIP LIMEIRA – UNIVERSIDADE PAULISTA

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CAIO YAGO VILELA

F00JED7

ENRIGO MARIANO CASSIANO

F00JGH4

GUILHERME L. C. F. CORTEZ

N480EA9

MAYCON LEONARDO VAZ DE LIMA

D898IB8

WELINGTON DOS SANTOS SALES

N401176

DIAGRAMA ENTIDADE RELACIONAL

MERCADO INTELIGENTE

LIMEIRA – SÃO PAULO

2020

CAIO YAGO VILELA
ENRIGO MARIANO CASSIANO
GUILHERME L. C. F. CORTEZ
MAYCON LEONARDO VAZ DE LIMA
WELINGTON DOS SANTOS SALES

F00JED7
F00JGH4
N480EA9
D898IB8
N401176

DIAGRAMA ENTIDADE RELACIONAL

MERCADO INTELIGENTE

Relatório de Atividade Prática
Supervisionada (APS) para avaliação no 3º
Semestre letivo do curso de Ciência da
Computação apresentado à UNIP Limeira –
Universidade Paulista.

Orientador (es): Danilo Pereira

LIMEIRA – SÃO PAULO

2020

SUMÁRIO

1.	RESUMO	4
2.	INTRODUÇÃO	4
3.	OBJETIVO.....	5
4.	GENERALIZAÇÃO/ ESPECIALIZAÇÃO	5
5.	NORMALIZAÇÃO.....	7
6.	FORMAS NORMAIS.....	7
6.1.	Primeira Forma Normal	8
6.2.	Segunda forma Normal	9
6.3.	Terceira Forma Normal	10
7.	NOTAÇÃO	11
8.	CHAVES.....	12
8.1.	Chave Primária / Primary Key (PK).....	12
8.2.	Chave Estrangeira / Foreign Key (FK).....	13
9.	CARDINALIDADE	13
10.	O PROJETO E COMO FOI FEITO	15
11.	BIBLIOGRAFIA.....	17

1. RESUMO

O presente trabalho apresenta de forma completa e organizada de um Diagrama Entidade Relacional (DER), no qual é identificado as Tabelas (Entidades) e seus atributos, para coletar dados dos usuários ou qualquer sistema que venha ser criado definidos pelos critérios adotados nesse trabalho e também uma forma de melhorar a implementação junto ao SGBD. O DER foi criado através de requisitos, pensados com base em um mercado inteligente, isso significa que, utilizamos o DER para termos uma base e ideia de como e o que faríamos para criação e implementação de um mercado inteligente, onde os clientes irão em busca de um produto que será facilmente identificado, e esse Diagrama Entidade Relacional mostra especificadamente cada passo que seria indispensável para o desenvolvimento do mercado inteligente.

2. INTRODUÇÃO

O modelo relacional seria um modelo representativo, o qual foi adequado para ser uma base para o uso aplicado nos (SGBD) – Sistema Gerenciador de Banco de Dados, o qual baseia-se de todos os dados coletados serão armazenados em tabelas que posteriormente serão usadas para relacionar-se com demais tabelas, onde será permitido a extração de informações para tomadas de decisões. Esse conceito foi criado em 1970 por Edgar Frank Codd. Basicamente o Modelo de Relacional, é baseado em outros dois conceitos: os conceitos de entidade e relação. Uma entidade é caracterizada por dados recolhidos em sua identificação designados na tabela. O modelo relacional entre outros benefícios para os SGBD, permite que seu Desenvolvedor crie um modelo lógico consistente das informações que irão ser armazenadas. O modelo relacional permite ao projetista criar um modelo lógico consistente da informação a ser armazenada. Este modelo lógico pode ser refinado através de um processo de normalização. Um banco de dados construído puramente baseado no modelo relacional estará inteiramente normalizado. O plano de acesso, outras implementações e detalhes de operação são tratados pelo sistema DBMS, e não devem ser refletidos no modelo lógico. Isto se contrapõe à prática comum para

DBMS's SQL nos quais o ajuste de desempenho frequentemente requer mudanças no modelo lógico.

Os blocos básicos do modelo relacional são o domínio, ou tipo de dado. Uma tupla é um conjunto de atributos que são ordenados em pares de domínio e valor. Uma relvar (variável relacional) é um conjunto de pares ordenados de domínio e nome que serve como um cabeçalho para uma relação. Uma relação é um conjunto desordenado de tuplas. Apesar destes conceitos matemáticos, eles correspondem basicamente aos conceitos tradicionais dos bancos de dados. Uma relação é similar ao conceito de tabela e uma tupla é similar ao conceito de linha. O princípio básico do modelo relacional é o princípio da informação: toda informação é representada por valores em relações (relvars). Assim, as relvars não são relacionadas umas às outras no momento do projeto. Entretanto, os projetistas utilizam o mesmo domínio em vários relvars, e se um atributo é dependente de outro, esta dependência é garantida através da integridade referencial.

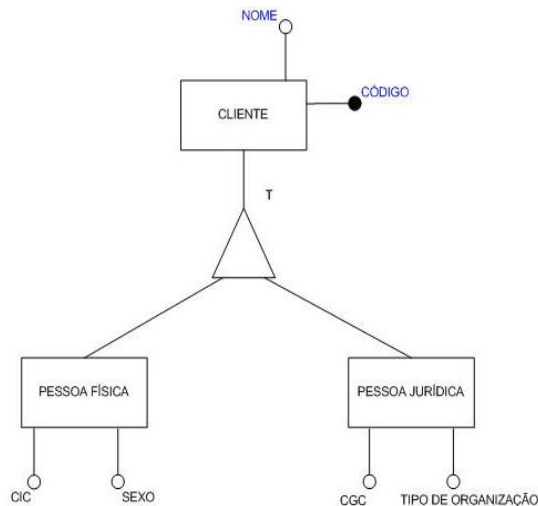
3. OBJETIVO

Pesquisar e desenvolver um Diagrama Entidade Relacional, utilizando as técnicas de desenvolvimento para o DER, expondo suas Entidades e Atributos em um modelo de Diagrama, no qual será perceptível ao profissional que irá desenvolver o SGBD as diretrizes que o banco irá seguir na questão de coletar dados e armazená-los, tornando-os relacionais para extração de informações que serão fornecidas ao usuário. Através do DER o desenvolvedor pode aplicar os conceitos de forma facilitada e normalizada.

Os conceitos destacados no DER mostraram o relacionamento que as tabelas deverão ter ao criar um Banco de Dados, se por algum acaso o desenvolvedor achar necessário alterar algum relacionamento ou criar outra tabela o DER, também pode ser refinado caso necessário, se assim for a preferência do desenvolvedor.

4. GENERALIZAÇÃO/ ESPECIALIZAÇÃO

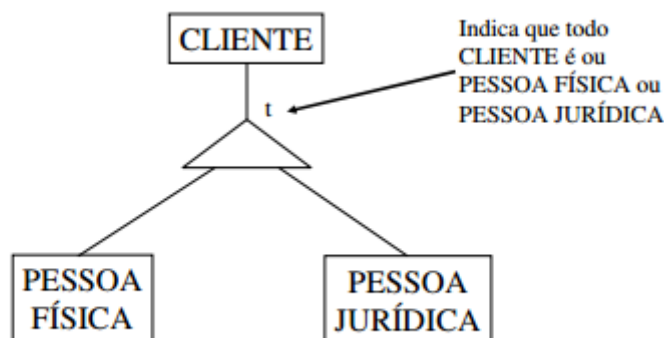
São para simbolizar objetos utilizados no mundo real com atributos iguais podendo ser representados de forma hierárquica. Como o exemplo abaixo em que uma empresa tenha clientes que são pessoas e clientes empresas.



FONTE:

<https://sites.google.com/site/uniplibancodedados1/aulas/aula-9---generalizacao-e-especializacao>

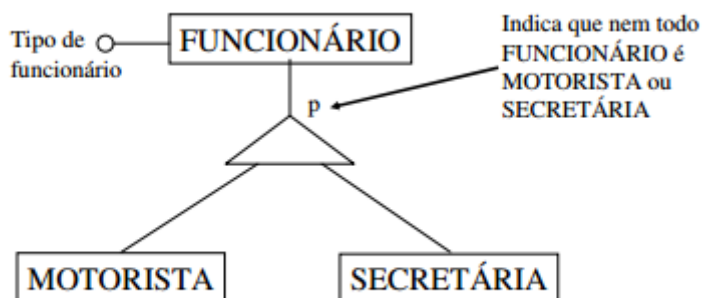
Contam com dois tipos, o Total(T) em que todo uso de uma entidade genérica sempre tem sua resposta em uma das entidades especializadas.



FONTE:

<https://sites.google.com/site/uniplibancodedados1/aulas/aula-9---generalizacao-e-especializacao>

E o outro tipo é o Parcial(P), e que nem todo uso de uma entidade genérica resulta com a resposta de uma entidade especializada.



FONTE:

<https://sites.google.com/site/uniplibancodedados1/aulas/aula-9---generalizacao-e-especializacao>

Seu uso é aconselhável quando um atributo pode ser aplicável a mais de uma entidade em um DER. Criando uma entidade com os atributos comuns das entidades especializadas. Utilize a Generalização para criação de uma entidade especializada caso seja necessário ou estará apenas inserindo informação desnecessária ao DER.

5. NORMALIZAÇÃO

Normalização é o processo de modelar o banco de dados projetando a forma como as informações serão armazenadas a fim de eliminar, ou pelo menos minimizar, a redundância no banco. Tal procedimento é feito a partir da identificação de uma anomalia em uma relação, decompondo-as em relações melhor estruturadas.

Normalmente precisamos remover uma ou mais colunas da tabela, dependendo da anomalia identificada e criar uma segunda tabela, obviamente com suas próprias chaves primárias e relacionarmos a primeira com a segunda para assim tentarmos evitar a redundância de informações.

Um banco de dados dentro dos padrões de normalização reduz o trabalho de manutenção e ajuda a evitar o desperdício do espaço de armazenamento. Se tivermos cadastrado no banco um cliente e tivermos o seu telefone registrado em mais de uma tabela, havendo uma alteração no seu número de telefone, teremos que fazer essa atualização em cada tabela.

A tarefa se torna muito mais eficiente se tivermos seu telefone registrado em apenas uma tabela.

6. FORMAS NORMAIS

Temos conjuntos de regras para determinar com qual forma normal o banco é compatível. Primeiramente, precisamos verificar se encontramos compatibilidade com a primeira forma normal. Caso esteja tudo conforme, analisamos se a segunda forma normal se encaixa e assim sucessivamente.

É importante lembrar que existem 6 formas normais e que para uma relação atender as exigências de uma forma normal, se faz necessário que esta obedeça às

regras da forma normal anterior. A primeira forma normal é exceção pois não existe uma forma normal anterior a primeira.

6.1. Primeira Forma Normal

Uma relação está na primeira forma normal quando todos os atributos contêm apenas um valor correspondente, singular e não existem grupos de atributos repetidos, não admite repetições ou campos que tenham mais que um valor.

O procedimento inicial é identificar a chave primária da tabela. Após, devemos reconhecer o grupo repetitivo e removê-lo da entidade. Em seguida, criamos uma nova tabela com a chave primária da tabela anterior e o grupo repetitivo.

Código	Nome	Endereço	Telefone
1001	Diego Machado	Rua Tal 321 Porto	5312345678 5398765432
1002	Fulano de Tal	Avenida Tal 71 Centro	5187654321 5143215678

(Tabela não está na primeira forma normal)

Analisando o exemplo acima, podemos observar dois problemas: Temos uma pessoa com dois números de telefone e um endereço com diferentes valores, a rua e o bairro. A fim de normalizar, teremos que colocar cada informação em uma coluna diferente e criar uma nova tabela relacionando a pessoa a seus números de contato.

Código	Nome	Endereço	Bairro
1001	Diego Machado	Rua Tal 321	Porto
1002	Fulano de Tal	Avenida Tal 71	Centro

(Tabela está na primeira forma normal)

Dessa forma, como mostrado na tabela acima, temos uma tabela na primeira forma normal evitando assim repetições e campos com múltiplos valores, conforme observamos na tabela abaixo.

Código	Telefone
1001	5312345678
1001	5398765432
1002	5112345678
1002	5187654321

(Nova tabela criada para evitar campos com mais de um valor)

6.2. Segunda forma Normal

Uma tabela está na segunda forma normal se ela atende a todos os requisitos da primeira forma normal e se os registros na tabela, que não são chaves, dependam da chave primária em sua totalidade e não apenas parte dela. A segunda forma normal trabalha com essas irregularidades e previne que haja redundância no banco de dados.

Para isso, devemos localizar os valores que dependem parcialmente da chave primária e criar tabelas separadas para conjuntos de valores que se aplicam a vários registros e relacionar estas tabelas com uma chave estrangeira.

cd_locacao	cd_filme	titulo_filme	devolucao	cd_cliente
1010	201	The Matrix	2011-10-12	743
1011	302	O Grito	2011-12-10	549
1012	201	The Matrix	2011-12-30	362

(Tabela não está na segunda forma normal)

Podemos observar que a tabela acima apresenta uma coluna responsável por armazenar o título do filme, onde este foi alugado e está associado a um número de locação. Porém, ele também está associado a um código, tornando-o então um valor que não é totalmente dependente da chave primária da tabela.

cd_filme	titulo_filme
201	The Matrix
302	O Grito

(Tabela criada para armazenar os filmes)

Se em algum momento tivemos que alterar o título de um filme, teríamos que procurar e alterar os valores em cada tupla (linha) da tabela. Isso demandaria um trabalho e tempo desnecessário. Porém, ao criarmos uma tabela e vinculamos elas com o recurso da chave estrangeira, tornamos o nosso banco mais organizado e ágil para as futuras consultas e manutenções que podem vir a ser necessárias.

cd_locacao	cd_filme	devolucao	cd_cliente
1010	201	2011-10-12	743
1011	302	2011-12-10	549
1012	201	2011-12-30	362

(Tabela na segunda forma normal)

6.3. Terceira Forma Normal

Se analisarmos uma tupla e não encontrarmos um atributo não chave dependente de outro atributo não chave, podemos dizer que a entidade em questão está na terceira forma normal - contanto que esta não vá de encontro as especificações da primeira e da segunda forma normal.

Como procedimento principal para configurar uma entidade que atenda as regras da terceira forma normal, nós identificamos os campos que não dependem da chave primária e dependem de um outro campo não chave. Após, separamos eles para criar uma outra tabela distinta, se necessário.

placa	modelo	qtd_kmetro	cod_fab	nome_fab
qwe1234	Modelo1	867	3004	fabricante1
asd456	Modelo2	928	3005	fabricante2

(Tabela não está na terceira forma normal)

No exemplo acima temos uma entidade que lista os carros cadastrados, bem como o modelo, a quantidade de quilômetros rodados, o código do fabricante e o nome do fabricante. Observamos que “nome_fab” se dá em função de “cod_fab”. Para

adequarmos esta tabela de acordo com os padrões da terceira forma normal, devemos remover a coluna do nome do fabricante.

placa	modelo	qtd_kmetro	cod_fab
qwe1234	Modelo1	867	3004
asd456	Modelo2	928	3005

(Tabela na terceira forma normal)

A coluna que removemos deve ser colocada em uma nova tabela, relacionando corretamente o nome do fabricante com o seu código. Abaixo, podemos observar como ficaria esta nova entidade.

cod_fab	nome_fab
3004	fabricante1
3005	fabricante2

(Tabela criada para armazenar o nome do fabricante)

7. NOTAÇÃO

De acordo com o dicionário Michaelis: Notação:

Ato ou efeito de notar.

Sistema de representação ou designação convencional.

Conjunto de sinais com que se faz essa representação ou designação.

Em outras palavras, em relação a projetos de bancos de dados, a notação é a forma utilizada para representar determinado modelo. Algumas notações são as preferidas pelos arquitetos de dados. Dentre elas, citarei três principais que podem ser utilizadas no próprio

8. CHAVES

8.1. Chave Primária / Primary Key (PK)

Este tipo de chave, refere-se aos conjuntos de um ou mais campos, cujos valores, considerando a combinação de valores de todos os campos da tupla (registro), nunca se repetem e que podem ser usadas como um índice para os demais campos da tabela do banco de dados. Em chaves primárias, não pode haver valores nulos nem repetição de tuplas.

Quando a chave é simples ela é formada por um único campo da tabela, sendo que este campo não pode ter dois valores ou mais registros de mesmo valor, e não pode conter registro nulo.

Quando a chave é composta, ou seja, formada por mais de um campo, os valores podem se repetir, mas não a combinação desses valores. Exemplo: a tabela 'Livros_Autores' tem como chave primária (cod_livro, cod_autor). Podem existir nessa tabela os registros: (5, 9), (5, 10), (4, 9), (9, 5). Mas não podem existir dois registros (5, 9).

Podemos criar definir as chaves primárias logo após a declaração dos campos:

```
CREATE TABLE nome_tabela
(
    campo1 NOT NULL,
    campo2 NOT NULL,
    campoX ,
    PRIMARY KEY (campo1, campo2)
);
```

8.2. Chave Estrangeira / Foreign Key (FK)

Este outro tipo de chave é utilizado para criar os relacionamentos entre as tabelas. Imagine que você queira cadastrar vários produtos que sejam de uma determinada categoria. Toda vez que você preencher os dados do produto, precisaremos indicar a chave primária da tabela categoria que seja da categoria que o nosso produto pertencerá. Ou seja, quando inserirmos um registro na tabela de produtos com o “id_categoria”, essa chave primária da tabela “categorias” representará uma chave estrangeira (FK) dentro da tabela de produtos. É uma chave que vem de fora, de outra tabela.

Com essa chave estrangeira, podemos facilitar as consultas e fazer cruzamento de dados através destas referências, o que poderia gerar uma consulta que iria pegar o nome do produto e o nome da categoria que ele pertence para exibirmos em uma listagem dos dados.

Perceba que a chave estrangeira não será única dentro da tabela de produtos, já que podemos ter vários produtos de uma categoria. Já no caso da chave primária, sempre será e deverá ser única.

9. CARDINALIDADE

Em modelagem de dados a cardinalidade é um dos princípios fundamentais sobre relacionamento de um banco de dados relacional. Nela são definidos os graus de relação entre duas entidades ou tabelas.

No modelo relacional, podemos ter os seguintes níveis de relacionamento: 1:N, N:N, 1:1.

Por exemplo, considere um banco de dados desenhado para manter informações relativas a um hospital. Esse banco de dados poderá ter várias tabelas como:

- Tabela médico onde constará informações sobre o médico profissional;

- Tabela paciente onde constará dados relativos aos assuntos médico e sobre o tratamento do paciente;
- Tabela departamento onde será tratado as informações relativas as divisões departamentais do hospital.

Neste modelo teremos o seguinte cenário:

- Existirá o relacionamento vários-para-vários (N:N) entre os registros da tabela médico Oppenheimer e os registro da tabela paciente, um médico atende diversos pacientes, assim como um paciente pode ser atendido por diversos médicos;
- Existirá o relacionamento um-para-vários (1:N) no relacionamento entre a tabela departamento em relação a tabela de médicos, pois um médico, poderá trabalhar em somente um departamento do hospital, contudo, um departamento poderá ter vários médicos.
- Já o relacionamento um-para-um (1:1) será usado nos casos onde o registro de uma tabela só poderá ter uma associação com um registro de outra tabela. No nosso caso, isso caberia na relação entre um quarto de apartamento e um paciente. Pois um paciente só poderá estar em um determinado apartamento, e cada apartamento só poderá abrigar um determinado paciente (partindo do princípio de quartos individuais).

Uma das principais funções de cardinalidade, é manter a integridade do banco de dados, em associação com as regras de negócio, não permitindo que essas regras sejam quebradas causando anomalias no SGBD, dados repetidos ou fora de normalização. Essas associações são ligadas através de chaves (chave estrangeira e chave primária) que são registro de indexação que não se repetem e que podem ser usadas como um índice para os demais campos da tabela do banco de dados. Em chaves primárias, não pode haver valores nulos nem repetição.

Em modelos de dados complexos, o relacionamento poderá ocorrer centenas de vezes, envolvendo dezenas de tabelas. O renomado cientista da computação, C.J. Date, criou um método sistemático para essa organização dos modelos de banco de dados. Esse modelo é o que conhecemos como normalização de dados. A normalização de dados permite-nos entre outras coisas, evitar anomalias em

comando de delete e updates. Na vida real o processo de normalização de banco de dados consiste em dividir uma grande tabela com diversas colunas em tabelas menores.

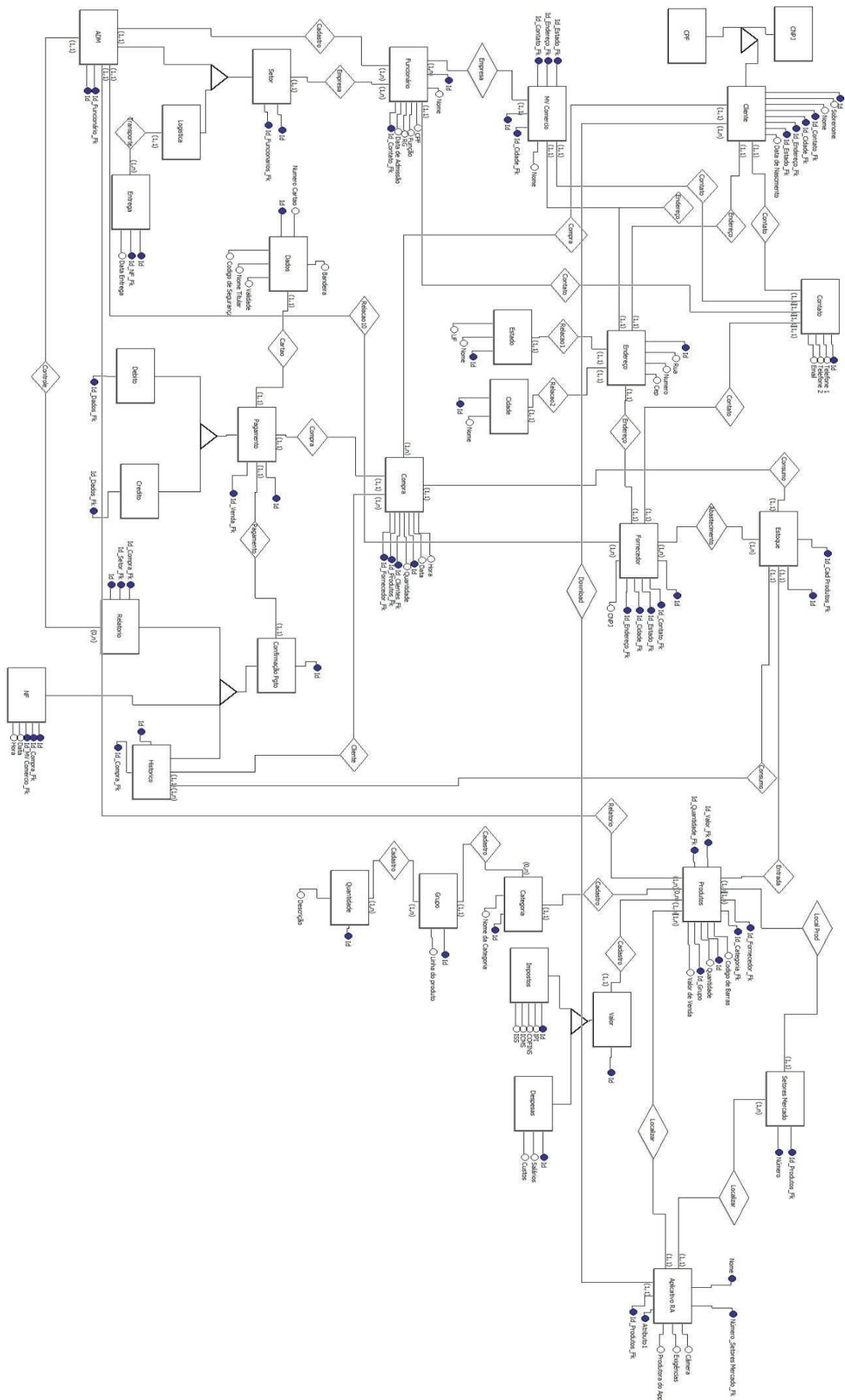
10. O PROJETO E COMO FOI FEITO

O projeto foi definido pelos conceitos que definimos como intrínsecos a um mercado inteligente. Visamos buscar o que seria necessário para a efetivação de tal mercado, que oferecesse facilidades que fossem executadas de forma eficiente.

Para a base do DER, focamos em características sempre presentes em um mercado, como estoque, fornecedor, produtos, setores, definindo tais entidades. Juntamente, analisamos os processos existentes em um comércio desse tipo, definindo as características de entidades, como Nota Fiscal e um Relatório para pagamento, o setor que um funcionário está atrelado, a logística empregada a esse setor, ligados à Administração, que gerencia todos os setores do mercado.

Um diferencial nesse projeto é a existência de um aplicativo do mercado, ideia essa que partiu do princípio de que tal mercado possui um conceito de oferecer uma experiência superior ao cliente, que experimentará de uma eficiência maior em relação a mercados comuns, através de tal aplicativo, onde o cliente pode efetuar suas compras, assim como conferir o que há em estoque para ser adquirido.

Sendo tais metas definidas, elaboramos o projeto, determinando todas as entidades necessárias para tanto, cuidando para que o DER possuísse a objetividade necessária às expectativas, resultando na efetivação de nossos conceitos aplicados no projeto.



11. BIBLIOGRAFIA

WIKIPÉDIA, A ENCICLOPÉDIA LIVRE. Wikipédia, a enciclopédia livre. **Wikipédia, a enciclopédia livre**. Disponível em: <https://pt.wikipedia.org/wiki/Modelo_relacional>. Acesso em: 18 maio 2020.

MICROSOFT. Descrição dos conceitos de normalização banco de dados básicos. 2007. Disponível em: <<http://support.microsoft.com/kb/283878/pt-br>>. Acesso em: 20 abril 2020.

CONTRIBUIDORES DA WIKIPÉDIA. Normalização de dados. Wikipédia, a enciclopédia livre. 2011. Disponível em: <http://pt.wikipedia.org/w/index.php?title=Normaliza%C3%A7%C3%A3o_de_dados&oldid=25486265>. Acesso em: 20 abril 2020.

FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

Atividades Práticas Supervisionadas (laboratórios, atividades em biblioteca, Iniciação Científica, trabalhos Individuais e em grupo, práticas de ensino e outras)

NOME:

RA:_____

CURSO:

Ciências da Computação

CAMPUS:

SEMESTRE:

TURNOS:

Noturno

[illegible]

TOTAL DE HORAS: 61 horas

FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS - A PS

Atividades Práticas Supervisionadas (laboratórios, atividades em biblioteca, iniciação Científica, trabalhos Individuais e em grupo, práticas de ensino e outras)

NOME: Enrico Mariano Cassiano

RA: F00JGH4

CURSO: Ciência da Computação

CAMPUS: Limeira

SEMESTRE: 3º

TURNOS: Noturno

			ASSINATURA	
DATA	ATIVIDADE	TOTAL DE	ALUNO	PROFESSOR
19/03/2020	Reunião para entendimento do tema	2 Horas		
20/03/2020	Debate sobre o tema em questão	4 horas		
21/03/2020	Escolha dos critérios discutidos	4 horas		
01/04/2020	Esboço do trabalho, como será feito	4 horas		
03/04/2020	Pesquisa para complementar entendimento	5 horas		
07/04/2020	Novos critérios obtidos e discutidos	5 horas		
08/04/2020	Pesquisa sobre o tema em torno do mundo	5 horas		
13/04/2020	Usando o DER	5 horas		
14/04/2020	Implementação do que foi discutido	4 horas		
15/04/2020	Finalizado DER	5 horas		
28/04/2020	Reunião de demonstração do DER	4 horas		
29/04/2020	Refinamento do DER	5 horas		
04/05/2020	Iniciando relatório	4 horas		
07/05/2020	Finalizando projeto	4 horas		
TOTAL DE HORAS: 60 Horas				

FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

Atividades Práticas Supervisionadas (laboratórios, atividades em biblioteca, Iniciação Científica, trabalhos Individuais e em grupo, práticas de ensino e outras)

NOME: Guilherme Lourenço C. F. Cortez

RA: N480EA9

CURSO: Ciência da Computação

CAMPUS: Unip-Limeira

SEMESTRE: 3º

TURNO: Noturno

DATA	ATIVIDADE	TOTAL DE HORAS	ASSINATURA	
			ALUNO	PROFESSOR
19/03/2020	Revisando o tema	4		
20/03/2020	Debatendo o tema em grupo	5		
21/03/2020	Estabelecendo Conceitos a serem aplicados no projeto	4		
01/04/2020	Elaborando o esboço do DER	4		
02/04/2020	Continuando o esboço do DER	5		
07/04/2020	Adicionando novas ideias aos conceitos do projeto	3		
08/04/2020	Pesquisando como o tema se aplica ao redor do mundo	4		
13/04/2020	Começando o DER do projeto	5		
14/04/2020	Colocando em pratica (DER) as ideias definidas em grupo	4		
15/04/2020	DER finalizado	5		
28/04/2020	Revisando a objetividade DER	3		
04/05/2020	Iniciando Relatorio	5		
06/05/2020	Finalizando Relatorio	5		
07/05/2020	Finalizando Projeto	4		

TOTAL DE HORAS: 60 Horas

FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

Atividades Práticas Supervisionadas (laboratórios, atividades em biblioteca, Iniciação Científica, trabalhos Individuais e em grupo, práticas de ensino e outras)

NOME: Maycon Leonardo Vaz de Lima

RA: D898|B8

CURSO: Ciência da Computação

CAMPUS: Unip-Limeira

SEMESTRE: 3º

TURN0: Noturno

[illegible]

FICHA DE ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

Atividades Práticas Supervisionadas (laboratórios, atividades em biblioteca, Iniciação Científica, trabalhos Individuais e em grupo, práticas de ensino e outras)

NOME:	Welington dos Santos Sales
--------------	----------------------------

RA: N401176

CURSO: Ciência da Computação

CAMPUS: Unip-Limeira

SEMESTRE: 3º

TURN: Noturno

[illegible]