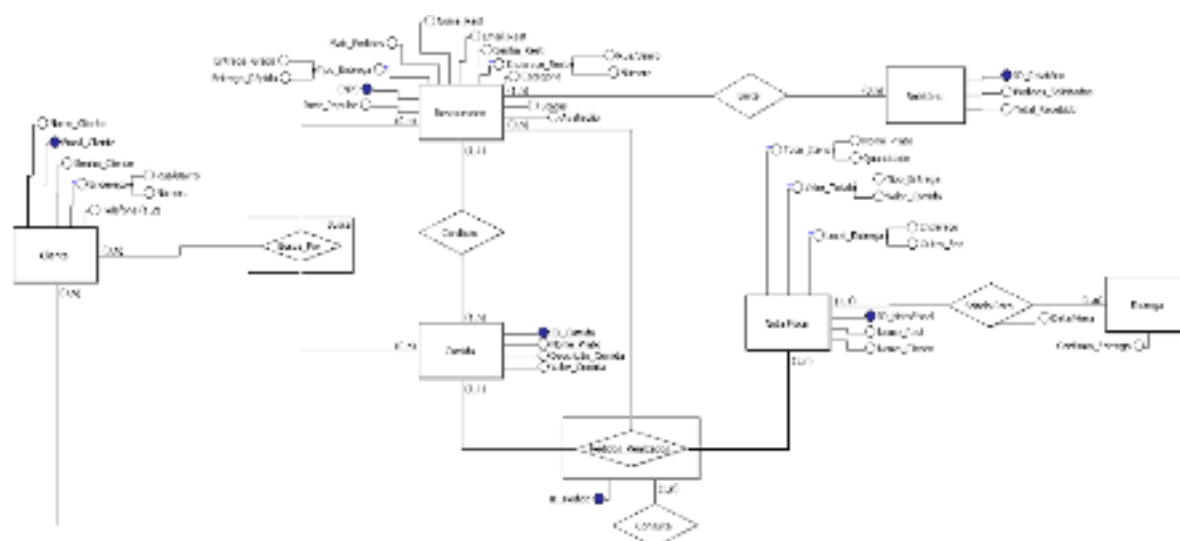


## **1. Modelagem Conceitual: Diagrama ER - Entidade Relacionamento**

Na construção desse modelo não é levado em conta o banco de dados em si, mas sim os requisitos do sistema, ou seja, qual sua funcionalidade e quais estruturas serão criadas para armazenar os dados do usuário.



**Figura 1: Diagrama Entidade-Relacionamento**

Neste nosso modelo, temos como entidades: **Cliente**, **Restaurante**, **Relatório**, **Comida**, **Nota Fiscal** e **Entrega**. Como a busca pode ser feita tanto através da Comida quanto do Restaurante, a entidade **Busca\_Por** esta como uma *entidade associativa*. Isso acontece quando o relacionamento com cardinalidade n:n entre duas entidades precisa se relacionar com uma outra entidade, isto é, Cliente além de se relacionar com Restaurante, também se relaciona com Comida. O mesmo acontece com **Pedidos\_Realizados**; essa entidade precisa se relacionar com a entidade Restaurante.

Nota Fiscal está como uma entidade fraca, pois ela é uma entidade dependente, ou seja, só existe Nota Fiscal se houver pelo menos um pedido realizado, isto é, pelo menos uma comida pedida.

Em relação ao atributos das entidades, temos por exemplo, em Cliente, atributos simples, atributo identificador como e-mail, atributos compostos como endereço que possui



rua/bairro e número, e atributo multivalorado como telefone com espaço para dois números. Em Restaurante também temos atributos simples, atributos compostos como endereço e Tipo de Entrega que pode ser rápida ou grátis, e atributo identificador que é o CNPJ do lugar. As outras entidades seguem a mesma sequência.

## **2. Diagrama de Tabelas: Modelo Lógico**

Para este modelo poder ser construído, é necessário que a modelagem dos dados realizada no modelo conceitual esteja pronta, sendo assim, acaba por ser um modelo um pouco mais limitado. Aqui são implementados recursos como definição de chaves primárias e estrangeiras, integridade referencial, adequação ao padrão de nomenclatura, normalização, entre outras.

As entidades com relacionamentos n:n estão com tabelas próprias devido a esses tipos de relacionamentos permitirem essa criação. Os atributos compostos e multivalorados foram colocados em tabelas diferentes da tabela da entidade à qual pertencem. Essa decisão foi tomada para evitar tabelas grandes, que é o que esses atributos iriam causar na tabela da entidade pertencente, caso fossem postos nela.

Como a relação entre Cliente, Restaurante e Comida é do tipo n:n, o relacionamento Busca\_Por, nesta conversão para o modelo lógico, possui uma tabela própria, onde nela tem os atributos identificadores das entidades relacionadas. O mesmo acontece com a relação entre Cliente e Pedidos\_Realizados, chamada de Consulta e entre a relação Restaurante e Relatório, chamada Emite.

Já o relacionamento entre Restaurante e Comida, chamado cardápio, é do tipo 1:n e, como esse relacionamento permite adição de coluna, Restaurante e Comida possuem cada um uma tabela, porém Restaurante tem seus atributos e Comida tem seus atributos mais os atributos identificadores de Restaurante. O mesmo acontece com o relacionamento entre Comida e Nota Fiscal, chamado de Pedidos\_Realizados. Além disso, como foi dito na seção acima, Nota Fiscal é uma entidade fraca, então ela possui seus atributos mais o atributo da entidade forte que nesse caso é Comida. Esse esquema é repetido na relação entre Nota Fiscal



e Entrega, chamada Saindo\_Para. Nesse caso, como é um relacionamento identificador, esse atributo fica em Nota Fiscal.

Segue abaixo o modelo lógico do modelo relacional apresentado na seção 1.

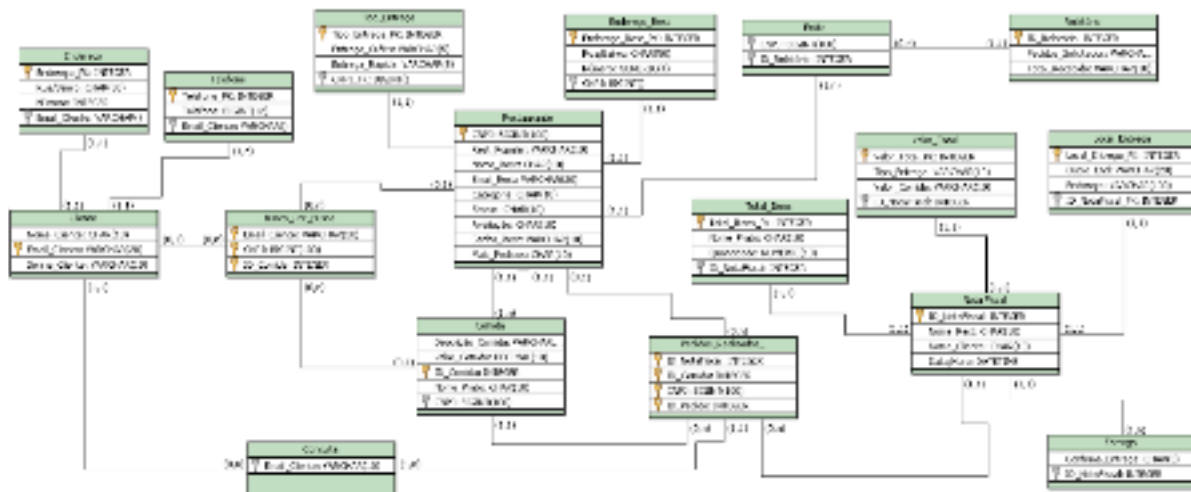


Figura 2: Modelo Lógico

### 3. Modelo Físico

O modelo físico foi gerado a partir do modelo lógico. Nele, podem ser realizadas as consultas em SQL.



```

-- Geração de Modelo Físico
-- Sql ANST 2003 - brModelo.

CREATE TABLE Busca_Por_Busca (
  Email_Cliente VARCHAR(20),
  CNPJ BIGINT(100),
  ID_Comida INTEGER,
  PRIMARY KEY(Email_Cliente,CNPJ,ID_Comida)
)

CREATE TABLE Tipo_Entrega (
  Tipo_Entrega_PK INTEGER PRIMARY KEY,
  Entrega_Grátis VARCHAR(5),
  Entrega_Rápida VARCHAR(5),
  CNPJ_FK BIGINT()
)

CREATE TABLE Telefone (
  Telefone_PK INTEGER PRIMARY KEY,
  Telefone BIGINT(12),
  Email_Cliente VARCHAR()
)

CREATE TABLE Emite (
  CNPJ BIGINT(100),
  ID_Relatório INTEGER
)

CREATE TABLE Relatório (
  ID_Relatório INTEGER PRIMARY KEY,
  Pedidos_Solicitados VARCHAR(10),
  Total_Recebido VARCHAR(10)
)

CREATE TABLE Endereço_Rest (
  Endereço_Rest_PK INTEGER PRIMARY KEY,
  Rua/Bairro CHAR(50),
  Número NUMERIC(4),
  CNPJ BIGINT()
)

CREATE TABLE Entrega (
  Confirma_Entrega CHAR(3),
  ID_NotaFiscal INTEGER
)

CREATE TABLE Local_Entrega (
  Local_Entrega_PK INTEGER PRIMARY KEY,
  Outro_End VARCHAR(20),
  Endereço VARCHAR(100),
  ID_NotaFiscal_FK INTEGER
)

CREATE TABLE Valor_Total (
  Valor_Total_PK INTEGER PRIMARY KEY,
  Tipo_Entrega VARCHAR(10),
  Valor_Comida VARCHAR(10),
  ID_NotaFiscal INTEGER
)

CREATE TABLE Total_Itens (
  Total_Itens_PK INTEGER PRIMARY KEY,
  Nome_Prato CHAR(10),
  Quantidade NUMERIC(10),
  ID_NotaFiscal INTEGER
)

CREATE TABLE Consulta (
  Email_Cliente VARCHAR(10)
)

CREATE TABLE Cliente (
  Nome_Cliente CHAR(10),
  Email_Cliente VARCHAR(20) PRIMARY KEY,
  Senha_Cliente VARCHAR(10)
)

CREATE TABLE Endereço (
  Endereço_PK INTEGER PRIMARY KEY,
  Rua/Bairro CHAR(50),
  Número INTEGER,
  Email_Cliente VARCHAR(),
  FOREIGN KEY(Email_Cliente) REFERENCES Cliente (Email_Cliente)
)

CREATE TABLE Nota_Fiscal (
  ID_NotaFiscal INTEGER PRIMARY KEY,
  Nome_Rest CHAR(10),
  Nome_Cliente CHAR(10),
  Data/Hora DATETIME
)

CREATE TABLE Comida (
  Descrição_Comida VARCHAR(10),
  Valor_Comida DECIMAL(10),
  ID_Comida INTEGER PRIMARY KEY,
  Nome_Prato CHAR(10),
  CNPJ BIGINT(100)
)

CREATE TABLE Restaurante (
  CNPJ BIGINT(100) PRIMARY KEY,
  Rest_Popular VARCHAR(10),
  Nome_Rest CHAR(10),
  Email_Rest VARCHAR(20),
  Categoria CHAR(10),
  Status CHAR(10),
  Avaliação CHAR(10),
  Senha_Rest VARCHAR(10),
  Mais_Pedidos CHAR(10)
)

CREATE TABLE Pedidos_Realizados_ (
  ID_NotaFiscal INTEGER,
  ID_Comida INTEGER,
  CNPJ BIGINT(100),
  ID_Pedido INTEGER,
  PRIMARY KEY(ID_NotaFiscal,ID_Comida,CNPJ,ID_Pedido)
)

ALTER TABLE Tipo_Entrega ADD FOREIGN KEY(CNPJ_FK) REFERENCES Restaurante (CNPJ)
ALTER TABLE Telefone ADD FOREIGN KEY(Email_Cliente) REFERENCES Cliente (Email_Cliente)
ALTER TABLE Emite ADD FOREIGN KEY(CNPJ) REFERENCES Restaurante (CNPJ)
ALTER TABLE Emite ADD FOREIGN KEY(ID_Relatório) REFERENCES Relatório (ID_Relatório)
ALTER TABLE Endereço_Rest ADD FOREIGN KEY(CNPJ) REFERENCES Restaurante (CNPJ)
ALTER TABLE Entrega ADD FOREIGN KEY(ID_NotaFiscal) REFERENCES Nota_Fiscal (ID_NotaFiscal)
ALTER TABLE Local_Entrega ADD FOREIGN KEY(ID_NotaFiscal_FK) REFERENCES Nota_Fiscal (ID_NotaFiscal)
ALTER TABLE Valor_Total ADD FOREIGN KEY(ID_NotaFiscal) REFERENCES Nota_Fiscal (ID_NotaFiscal)
ALTER TABLE Total_Itens ADD FOREIGN KEY(ID_NotaFiscal) REFERENCES Nota_Fiscal (ID_NotaFiscal)
ALTER TABLE Consulta ADD FOREIGN KEY(Email_Cliente) REFERENCES Cliente (Email_Cliente)

```

Figura 3: Modelo Físico





Para melhor visualização das imagens acima, consultar:  
[https://github.com/asleyi/Banco de Dados](https://github.com/asleyi/Banco_de_Dados)

