



UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS – PICOS
BACHARELADO EM SISTEMAS DE INFORMAÇÃO
PROFESSOR: FRANCISCO DAS CHAGAS IMPERES FILHO

PROJETO FINAL DA DISCIPLINA BANCO DE DADOS I

Componentes:

Luís Eduardo, Kennedy Camilo,
Vanderlei, Victor Macêdo,
Ismael Lima

Setembro de 2022

1. Informações sobre o projeto

Temática: Comércio

Título: Distopia solutar: delivery

Requisitos e principais características da aplicação:

É perceptível, em muitos brasis brasileiros, a carência de um sistema de delivery que satisfaça as necessidades do cliente, isso porque, as empresas não fazem questão de equiparar as necessidades do local onde a empresa está instalada, e nem muito menos de prover entregas rápidas e eficientes, resultando na perda de estoque e, por conseguinte, na perda de clientes. Com o objetivo de promover um sistema que apresente eficácia, esse projeto visa auxiliar as compras e vendas de uma empresa, desde o encaminhar do produto, bem como à entrega. Mantendo, assim, o controle do estoque de mercadorias, de clientes, bem como do caixa, armazenando suas informações e realizando processos lógicos de verificação, garantindo a eficácia em todos os procedimentos de serviço para com o cliente.

1.1 Principais entidades:

- Empresa de fast-food
- Cliente
- Funcionário
- Venda
- Entrega.

Os atributos com seus respectivos tipos de dados podem ser visualizados na próxima seção. 1.

Ata de reunião inicial com o(s) cliente(s)

Ata de Reunião de Levantamento de Requisitos		
Projeto distopia solutar: delivery		
Data:	Horário:	Local:
13/09/2022	13:00 – 14:00	Reunião On-line
Participantes do Cliente:	Bruno de Sousa	
Participantes do Desenvolvedor:	Victor Macêdo Luís Eduardo	
Tópicos discutidos		
Nº	Assunto	Descrição

1.	Apresentação inicial	<ol style="list-style-type: none"> Foram apresentados os conceitos iniciais da aplicação, tendo o seguinte objetivo: produzir um sistema local para profissionais da área comercial - funcionários (as) - capaz de prover um ambiente para facilitar e agilizar as informações buscadas. A proposta inicial é adquirir dados de determinada empresa para monitorar a movimentação (entrada e saída de lanches). Existe a perspectiva de montar uma base de dados com os produtos cadastrados.
----	----------------------	--

2.	Seções iniciais da ERSw	<ol style="list-style-type: none"> Todo usuário do sistema de delivery deverá possuir credenciais para prover atualizações na base de dados (login de acesso: nome do usuário e senha). O sistema terá como perfis de acesso os seguintes usuários: funcionários e gerente(s), podendo ser adicionado uma maior variedade futuramente. Os usuários com acesso autorizados podem acessar o banco de dados para adicionar ou remover itens do mesmo, além de atualizar os dados que já estão nos bancos de dados. O sistema de delivery deve proporcionar cadastramento dos dados de todos os envolvidos com a movimentação dos produtos de estoque (compra e venda). Alguns dados serão obrigatórios, como por exemplo: nome, CPF, endereço, contato (e-mail ou número do telefone). Os dados adicionados no banco de dados podem ser modificados, mas não é recomendado, logo ao cadastrar os dados deve-se ter muito cuidado ao informar dados incorretos.
----	-------------------------	---

4. Definição de requisitos

Versões revisadas anteriores

Responsáveis	Versão	Data	Descrição
--------------	--------	------	-----------

Luís Eduardo, Victor Macêdo, Kennedy Camilo, Ismael Lima e Vanderlei	1.0	13/09/2022	Finalização da versão inicial.
--	-----	------------	--------------------------------

Especificação dos Requisitos do
Projeto de Banco de Dados 1.0
Sumário

1. INTRODUÇÃO

- 1.1. Nome do projeto 6
- 1.2. Escopo do projeto 6
- 1.3. Limites do projeto 6
- 1.4. De iniciais e siglas 6

2. DEFINIÇÃO DOS REQUISITOS 6

- 2.1. Requisitos Funcionais 7
- 2.2. Requisitos Não-Funcionais 8

1. Introdução

- 1.1. Nome do projeto

Distopia solutar: delivery

- 1.2. Escopo do projeto

Uma maneira de estabelecer conexão com a localização do cliente, de modo que facilite o encaminhar dos produtos – mais rápido e sem que o cliente transite até o estabelecimento em análise –, haja vista que a empresa monitore e controle o gerenciamento do estoque de forma eficaz, seguindo, paralelamente, o fluxo de vendas daquele determinado ambiente

1.3. Limites do projeto

Com o objetivo de promover um sistema que apresente eficácia, esse projeto visa auxiliar as compras e vendas de uma empresa, desde o encaminhar do produto, bem como à entrega. Mantendo, assim, o controle do estoque de mercadorias, de clientes, bem como do caixa, armazenando suas informações e realizando processos lógicos de verificação, garantindo a eficácia em todos os procedimentos de serviço para com o cliente.

1.4. Definições e siglas

Nr.	Sigla	Definição
1	D-ER	Diagrama Entidade-Relacionamento
2	MySQL	Banco de dados relacional de código livre, bastante popular e utilizado por muitas empresas no mundo.
3	SGBD	Sistema Gerenciador de Banco de Dados.
4	WorkBench	Ferramenta que permite criar D-ER para bases de dados MySQL.
5	SQL	Structure Query Language - Linguagem de Consulta Estruturada.
6	ERSw	Especificação de Requisitos de Software

2. Definição dos requisitos

2.1. Requisitos Funcionais

ID	Requisito	Descrição
RF1	Clientes	Cadastro e gestão dos dados pessoais dos respectivos clientes.
RF2	Fornecedores	Cadastro e gestão dos dados dos respectivos fornecedores.

RF3	Funcionários	Cadastro e gestão dos dados dos respectivos funcionários.
RF4	Produtos	Cadastros e gestão dos dados dos produtos com todas as suas informações.
RF5	Endereço	Localização do cliente, funcionário, para, respectivamente, realizar a entrega ao cliente e entender a localização do funcionário, durante o a realização da entrega.

2.2. Requisitos não funcionais

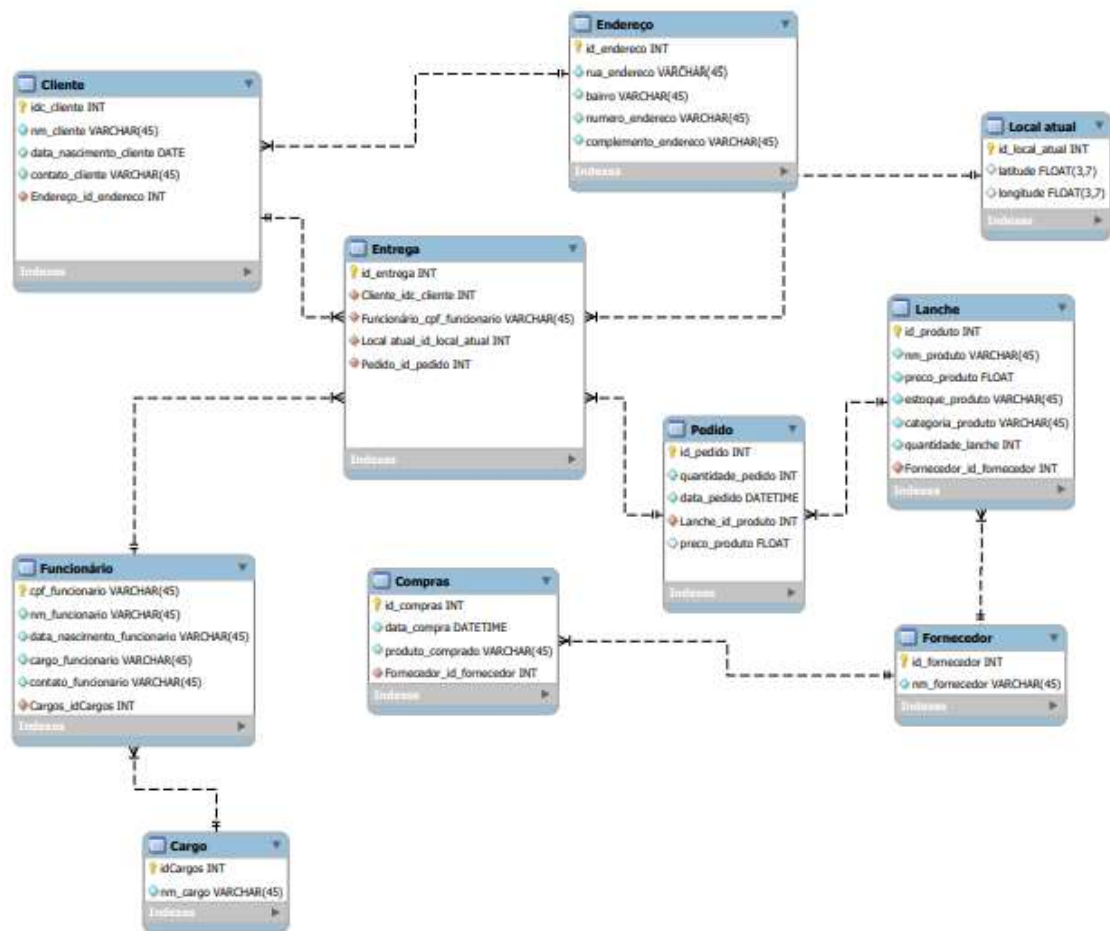
ID	Requisito	Descrição
RNF1	Ambiente	O projeto vai usar o ambiente gráfico WorkBench (versão 8.0.28.), para criar as estruturas internas, atualizações e consultas à base de dados.
RNF2	Linguagem	O projeto vai utilizar SQL para criar o esquema interno e prover atualizações na base de dados.
RNF3	Banco de dados	O projeto será desenvolvido utilizando o SGBD MySQL (versão 8.0.28.).
RNF4	Desempenho	O projeto deve ser construído de forma a facilitar o acesso de múltiplos usuários simultâneos.

RNF5	Segurança	O projeto deve restringir o acesso de usuários por meio de senhas. Deve-se ter um controle no registro de senha, de forma a impedir o uso de senhas consideradas fáceis.
RNF6	Backup e Restauração de dados	O backup e a recuperação da base de dados serão automatizados através de rotinas internas providas pelo SGC.
RNF7	Usabilidade	Nesse estágio do projeto somente a equipe de desenvolvimento terá privilégios para criar e interagir com a base de dados.

5. Diagrama Entidade-Relacionamento (Exemplo)

Ferramenta utilizada para criação do diagrama:

- MySQL Workbench
 - Versão 8.0.28 (64 bits)



-- MySQL Script generated by MySQL Workbench
-- Fri Sep 16 10:47:09 2022
-- Model: New Model Version: 1.0
-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;  
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0;  
SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZER  
O_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

-- Schema mydb

```
DROP SCHEMA IF EXISTS `mydb` ;
```

-- Schema mydb

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;  
USE `mydb` ;
```

-- Table `mydb`.`Endereco`

```
DROP TABLE IF EXISTS `mydb`.`Endereco` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Endereco` (  
  `id_endereco` INT NOT NULL AUTO_INCREMENT,  
  `rua_endereco` VARCHAR(45) NOT NULL,  
  `bairro` VARCHAR(45) NOT NULL,  
  `numero_endereco` VARCHAR(45) NOT NULL,  
  `complemento_endereco` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id_endereco`))  
ENGINE = InnoDB;
```

-- Table `mydb`.`Cliente`

```
DROP TABLE IF EXISTS `mydb`.`Cliente` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Cliente` (  
  `idc_cliente` INT NOT NULL AUTO_INCREMENT,  
  `nm_cliente` VARCHAR(45) NOT NULL,  
  `data_nascimento_cliente` DATE NOT NULL,  
  `contato_cliente` VARCHAR(45) NOT NULL,  
  `Endereco_id_endereco` INT NOT NULL,  
  PRIMARY KEY (`idc_cliente`),  
  INDEX `fk_Cliente_Endereco1_idx` (`Endereco_id_endereco` ASC) VISIBLE,  
  CONSTRAINT `fk_Cliente_Endereco1`  
    FOREIGN KEY (`Endereco_id_endereco`)  
    REFERENCES `mydb`.`Endereco` (`id_endereco`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `mydb`.`Cargo`

DROP TABLE IF EXISTS `mydb`.`Cargo` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Cargo` (
 `idCargos` INT NOT NULL,
 `nm_cargo` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`idCargos`))
ENGINE = InnoDB;

-- Table `mydb`.`Funcionário`

DROP TABLE IF EXISTS `mydb`.`Funcionário` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Funcionário` (
 `cpf_funcionario` VARCHAR(45) NOT NULL,
 `nm_funcionario` VARCHAR(45) NOT NULL,
 `data_nascimento_funcionario` VARCHAR(45) NOT NULL,
 `cargo_funcionario` VARCHAR(45) NOT NULL,
 `contato_funcionario` VARCHAR(45) NOT NULL,
 `Cargos_idCargos` INT NOT NULL,
 PRIMARY KEY (`cpf_funcionario`),
 INDEX `fk_Funcionário_Cargos1_idx` (`Cargos_idCargos` ASC) VISIBLE,
 CONSTRAINT `fk_Funcionário_Cargos1`
 FOREIGN KEY (`Cargos_idCargos`)
 REFERENCES `mydb`.`Cargo` (`idCargos`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- Table `mydb`.`Local atual`

DROP TABLE IF EXISTS `mydb`.`Local_atual` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Local_atual` (
 `id_local_atual` INT NOT NULL AUTO_INCREMENT,
 `latitude` decimal(12,7) NULL,
 `longitude` decimal(12,7) NULL,
 PRIMARY KEY (`id_local_atual`))
ENGINE = InnoDB;

-- Table `mydb`.`Fornecedor`

DROP TABLE IF EXISTS `mydb`.`Fornecedor` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Fornecedor` (
 `id_fornecedor` INT NOT NULL AUTO_INCREMENT,
 `nm_fornecedor` VARCHAR(45) NOT NULL,
 PRIMARY KEY (`id_fornecedor`))
ENGINE = InnoDB;

-- Table `mydb`.`Lanche`

DROP TABLE IF EXISTS `mydb`.`Lanche` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Lanche` (
 `id_produto` INT NOT NULL AUTO_INCREMENT,
 `nm_produto` VARCHAR(45) NOT NULL,
 `preco_produto` FLOAT NOT NULL,
 `estoque_produto` VARCHAR(45) NOT NULL,
 `categoria_produto` VARCHAR(45) NOT NULL,
 `quantidade_lanche` INT NOT NULL,
 `Fornecedor_id_fornecedor` INT NOT NULL,
 PRIMARY KEY (`id_produto`),
 INDEX `fk_Lanche_Fornecedor1_idx` (`Fornecedor_id_fornecedor` ASC) VISIBLE,
 CONSTRAINT `fk_Lanche_Fornecedor1`
 FOREIGN KEY (`Fornecedor_id_fornecedor`)
 REFERENCES `mydb`.`Fornecedor` (`id_fornecedor`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- Table `mydb`.`Pedido`

DROP TABLE IF EXISTS `mydb`.`Pedido` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Pedido` (
 `id_pedido` INT NOT NULL AUTO_INCREMENT,
 `quantidade_pedido` INT NOT NULL,
 `data_pedido` DATETIME NOT NULL,
 `Lanche_id_produto` INT NOT NULL,
 `preco_produto` FLOAT NULL,
 PRIMARY KEY (`id_pedido`),
 INDEX `fk_Pedido_Lanche1_idx` (`Lanche_id_produto` ASC) VISIBLE,
 CONSTRAINT `fk_Pedido_Lanche1`
 FOREIGN KEY (`Lanche_id_produto`)
 REFERENCES `mydb`.`Lanche` (`id_produto`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- Table `mydb`.`Entrega`

DROP TABLE IF EXISTS `mydb`.`Entrega` ;

CREATE TABLE IF NOT EXISTS `mydb`.`Entrega` (
 `id_entrega` INT NOT NULL AUTO_INCREMENT,
 `Cliente_idc_cliente` INT NOT NULL,
 `Funcionário_cpf_funcionario` VARCHAR(45) NOT NULL,
 `Local_atual_id_local_atual` INT NOT NULL,
 `Pedido_id_pedido` INT NOT NULL,
 PRIMARY KEY (`id_entrega`),
 INDEX `fk_Entrega_Cliente1_idx` (`Cliente_idc_cliente` ASC) VISIBLE,
 INDEX `fk_Entrega_Funcionário1_idx` (`Funcionário_cpf_funcionario` ASC) VISIBLE,
 INDEX `fk_Entrega_Local_atual1_idx` (`Local_atual_id_local_atual` ASC) VISIBLE,
 INDEX `fk_Entrega_Pedido1_idx` (`Pedido_id_pedido` ASC) VISIBLE,
 CONSTRAINT `fk_Entrega_Cliente1`

```

FOREIGN KEY (`Cliente_idc_cliente`)
REFERENCES `mydb`.`Cliente` (`idc_cliente`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Entrega_Funcionario1`
FOREIGN KEY (`Funcionario_cpf_funcionario`)
REFERENCES `mydb`.`Funcionario` (`cpf_funcionario`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Entrega_Local_atual1`
FOREIGN KEY (`Local_atual_id_local_atual`)
REFERENCES `mydb`.`Local_atual` (`id_local_atual`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Entrega_Pedido1`
FOREIGN KEY (`Pedido_id_pedido`)
REFERENCES `mydb`.`Pedido` (`id_pedido`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`Compras`
-----

```

```

DROP TABLE IF EXISTS `mydb`.`Compras` ;

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`Compras` (
  `id_compras` INT NOT NULL AUTO_INCREMENT,
  `data_compra` DATETIME NOT NULL,
  `produto_comprado` VARCHAR(45) NOT NULL,
  `Fornecedor_id_fornecedor` INT NOT NULL,
  PRIMARY KEY (`id_compras`),
  INDEX `fk_Compras_Fornecedor1_idx` (`Fornecedor_id_fornecedor` ASC) VISIBLE,
  CONSTRAINT `fk_Compras_Fornecedor1`
    FOREIGN KEY (`Fornecedor_id_fornecedor`)
    REFERENCES `mydb`.`Fornecedor` (`id_fornecedor`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```