

PROJETO DE DESENVOLVIMENTO DE SISTEMA

MÓDULO PARA GERENCIAMENTO DE CLIENTES

WENDEL RODRIGUES FERREIRA BRITO, 25306¹

ALINE APARECIDA VICENTE DE SOUZA, 25604

ANDRÉ RICARDO CAPELETO, 25396

JEFERSON LUIZ BUTINHÃO DE OLIVEIRA, 25373

JOÃO GABRIEL SABATINI, 25471

RODNEY DE ANDRADE MARTINS, 25305

TIFFANY CARVALHO DAS NEVES, 25591

RESUMO

O projeto de desenvolvimento de sistema aborda a construção, de um modulo para gerenciamento de clientes, que busca automatizar um processo de uma empresa voltada ao ramo da confecção de embalagens de papelão Kraft, através da informatização de operações como inserção, remoção, listagem, consulta e edição de clientes, que antes eram feitas de modo manual, através do fichamento de formulários de cadastro físicos. Os pontos de abordagem deste documento, são focados em diversos campos da análise de sistemas, onde apresenta diagramas que descrevem não apenas o funcionamento lógico mas também a sua organização, bem como seus sub-produtos, um banco de dados e um 'software' escrito em linguagem Python. Portanto, ao final deste documento, será possível entender e distinguir as etapas de criação de um sistema computacional voltado á automação empresarial, bem como observar de modo detalhado as etapas e seus sub-produtos, além de alguns pensamentos que compõe a lógica por trás disso.

Palavras-chave ADS. Análise de Sistemas. Banco de Dados. Engenharia de Software. Programação Avançada. Orientação a Objetos. Projeto de Sistema. Python.

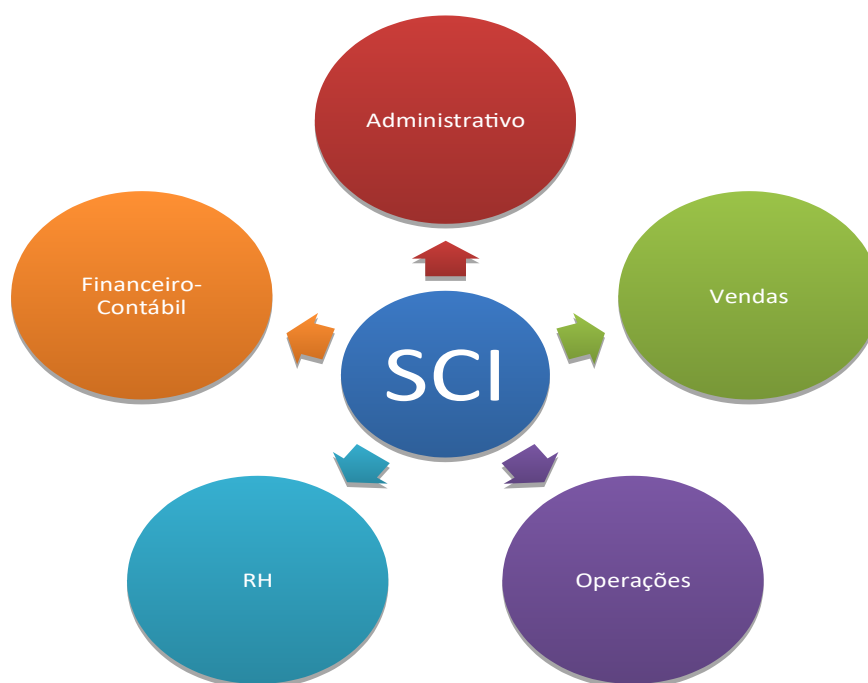
Campinas, SP

2020

1 INTRODUÇÃO

A crise do corona-virus (COVID-19), em 2020, levou a empresa **ACME Corp.** a considerar a informatização de seus principais processos administrativos. A ACME é uma pequena indústria do ramo de confecção de embalagens de papelão Kraft. Após analisar algumas soluções de mercado, que cobriam apenas parcialmente as suas necessidades de negócio, estrategicamente, a Direção decidiu por contratar uma consultoria de desenvolvimento de sistemas para proceder com a análise e desenvolvimento do **Sistema Corporativo Integrado (SCI)**, que nada mais é que um *Enterprise Resource Planning System (ERP)*. Este sistema é composto por diferentes módulos, como mostrado na Figura 1.

Figura 1 – Composição do Sistema Corporativo Integrado (SCI)



A sua equipe, neste trabalho, irá analisar, definir e desenvolver apenas as funções básicas para o funcionamento do **Módulo para Gerenciamento de Clientes** pertencente ao subsistema de VENDAS.

1.1 Tema Geral

Análise e Desenvolvimento de Sistemas: ANÁLISE DE SISTEMAS E PROGRAMAÇÃO AVANÇADA.

2 PROJETO DE DESENVOLVIMENTO DE SISTEMA

2.1 Estudo de Caso

O **Módulo para Gerenciamento de Clientes** é a base para o posterior desenvolvimento do subsistema de **Vendas**. Nele serão registrados os dados dos Clientes necessários para a emissão dos Pedidos de Venda, Emissão de Nota Fiscal Eletrônica (NF-e), eventuais cobranças e dos demais documentos legais.

2.1.1 Situação Atual

No momento, o registro dos Clientes é realizado **manualmente** na ficha vista na Figura 2. Não é gerado código único de identificação. O fichário (ou arquivo) é organizado em ordem crescente de razão social. Os dados do Cliente são obtidos a partir do recebimento de cópia do **Comprovante de Inscrição e de Situação Cadastral**. Pelo menos um número de telefone fixo deve ser registrado.

Todos os dados são **obrigatórios**, exceto Inscrição Estadual, Inscrição Municipal, Complemento, Telefone Celular, URL.

Atenção! Eventuais dúvidas devem ser sanadas com os contatos nas áreas de Vendas e Tecnologia, indicados para o seu projeto.

2.1.2 Situação Futura (desejável)

O módulo deve contemplar as seguintes funções.

Funções

1. Inserir Cliente
2. Editar (modificar/alterar) Cliente
3. Excluir Cliente
4. Consultar Cliente
5. Listar Clientes

Validação

O módulo deve realizar a validação de **todos** os dados, em especial² dos seguintes:

- CNPJ
- Inscrição Estadual
- Inscrição Municipal

² Pesquisar na Internet de que maneira estas validações são realizadas e **implementá-las**.

Figura 2 – Formulário para Cadastro de Cliente

1 RAZÃO SOCIAL			
2 NOME FANTASIA		3 CNPJ	
4 INSCRIÇÃO ESTADUAL		5 INSCRIÇÃO MUNICIPAL	
ENDEREÇO			
6 LOGRADOURO			
7 COMPLEMENTO		8 BAIRRO	
9 MUNICÍPIO		10 UF	11 CEP
12 TELEFONE(S)		13 CELULAR	
			<input type="checkbox"/> WhatsApp?
14 E-MAIL			
15 URL			

Extraído a partir de ACME Corp. (2020)

2.2 Análise de Requisitos

2.2.1 Requisitos Não Funcionais

2.2.1.1 Adaptabilidade (Fornecida pela Área de Tecnologia)

2.2.1.1.1 Aplicativo cliente deve rodar em

- Windows (Necessidade)
- Linux (Desejo)
- OSX (Desejo)

2.2.1.2 Confiabilidade (Fornecida pela Área de Vendas)

- O banco de dados deve garantir a confiabilidade das informações do cliente. (Necessidade)
- Todos os dados são obrigatórios exceto inscrição estadual, inscrição municipal, complemento, telefone celular e URL. (Necessidade)
- Os dados devem ser validados previamente antes de serem enviados ao banco. (Necessidade)

2.2.1.3 Disponibilidade (Fornecida pela Área de Vendas)

- O sistema deve manter estável durante o horário comercial. (Necessidade)

2.2.1.4 Eficiência (Fornecida pela Área de Tecnologia)

- O sistema deve atender a demanda de acessos daqueles utilizando a aplicação. (Necessidade)
- O sistema deve responder o mais rápido possível as requisições de informações. (Necessidade)

2.2.1.5 Flexibilidade (Fornecida pela Área de Vendas)

- O sistema poderá customizar os campos que serão listados. (Desejo)

2.2.1.6 Usabilidade (Fornecida pela Área de Tecnologia)

- Os usuários do sistema deverão apresentar carga anterior de conhecimento com interfaces de usuário. (Desejo)

2.2.2 Requisitos de Infraestrutura (Fornecida pela Área de Tecnologia)

2.2.2.1 Operação e administração

- Duas pessoas especializadas em servidores Linux (Debian ou CentOS) (Necessidade)

2.2.2.2 Requisitos de Hardware

2.2.2.2.1 Servidor de Banco de Dados (Necessidade)

2.2.2.2.1.1 Modo de Hospedagem

- Virtual Private Server (Desejo)

2.2.2.2.1.2 Processador

- Xeon Gold 5112 @ 2.40Hz (Desejo)

2.2.2.2.1.3 Sistema de Gerenciamento de Banco de Dados

- MySQL (Desejo)

2.2.2.2.1.4 Sistema Operacional

- Debian 10 ou CentOS (Necessidade)

2.2.2.2.1.5 Read-access memory

- 2 Gigabytes (Necessidade)

2.2.2.2.1.6 Armazenamento

- 20 Gigabytes (Preferencialmente Solid State Drive)

2.2.2.2.1.7 Configurações gerais do Sistema Operacional

- Acesso através de chaves RSA pelo SSH (Necessidade)
- Login remoto através de senha e nome de usuário desativado (Necessidade)
- Regra do crontab para reiniciar o servidor toda meia noite (Desejo)

2.2.2.2.1.8 Rede

- IP Fixo (Necessidade)
- Domínio (Desejo)

2.2.2.3 Interface Humano Computador (Aplicação Cliente)

2.2.2.3.1 Tipo de computador

- Desktop (Necessidade)

2.2.2.3.2 Software interpretador

- Python 3.x (Necessidade)

2.2.2.3.3 Sistema Operacional

- Windows (Necessidade)
- Linux (Desejo)
- OSX (Desejo)

2.2.2.3.4 Read-access memory

- Mínimo: 4 Gigabytes (Necessidade)
- Recomendado: 6 Gigabytes (Desejo)

2.2.2.3.5 Processador

- Mínimo: Intel Core 2 Duo (Necessidade)
- Recomendado: Intel Core 2 Quad (Desejo)

2.2.2.3.6 Armazenamento

- 40 Gigabytes ou maior (Preferencialmente Solid State Drive)

2.2.2.3.7 Rede

- Acesso à internet (Necessidade)

2.2.3 Requisitos de Software

2.2.3.1 Contingência (Fornecida pela Área de Tecnologia) (Necessidade)

1. O sistema não responde a requisições
2. Reiniciar o banco de dados e testar, caso erro continuar
3. Reiniciar o sistema operacional, caso erro continuar
4. Acionar os desenvolvedores, com pedido de correção de BUG

2.2.3.2 Funcionais (Fornecida pela Área de Vendas)

2.2.3.2.1 Cliente

- Editar (Necessidade)

- Excluir (Necessidade)
- Consultar (Necessidade)
- Inserir (Necessidade)
- Listar (Necessidade)
- Apresentar o resultado da busca dos clientes na lista em ordem crescente ou decrescente por razão social ou CNPJ. (Desejo)

2.2.3.3 Operacionais

2.2.3.3.1 Volume (Fornecida pela Área de Tecnologia)

- Cada requisição de consulta ao banco de dados deve utilizar cerca de 3 kilobytes de rede (Desejo)

2.2.3.3.2 Frequência (Fornecida pela Área de Vendas)

- Utilizado de forma constante no horário comercial

2.2.3.3.3 Disponibilidade (Fornecida pela Área de Vendas)

- Deve estar disponível no horário comercial (Necessidade)

2.2.3.3.4 Performance (Fornecida pela Área de Tecnologia)

- Deve atender a requisições dos usuários com baixo tempo de resposta. (Necessidade)
- O sistema não deve perder nenhuma requisição solicitada pelo usuário. (Necessidade)

2.2.3.3.5 Localização Física (Fornecida pela Área de Tecnologia)

- O Software Banco de dados deve ser instalado em uma VPS na nuvem (Desejo)
- O Software Banco de dados deve ser ao menos instalado no servidor da empresa (Necessidade)
- O Software Cliente deve ser instalado nos computadores da empresa (Necessidade)

2.2.3.4 Segurança (Fornecida pela Área de Tecnologia)

- Somente o cliente oficial deverá e poderá requisitar dados do Servidor Banco de Dados. (Necessidade)
- O software só pode ser instalado nos computadores da empresa. (Necessidade)

2.2.3.5 Técnicos (Fornecida pela Área de Tecnologia)

2.2.3.5.1 Arquitetura

- Client-Database (Desejo)

2.2.3.5.2 Linguagem de programação

- Python (Necessidade)

2.2.3.5.3 Tipo do banco de dados

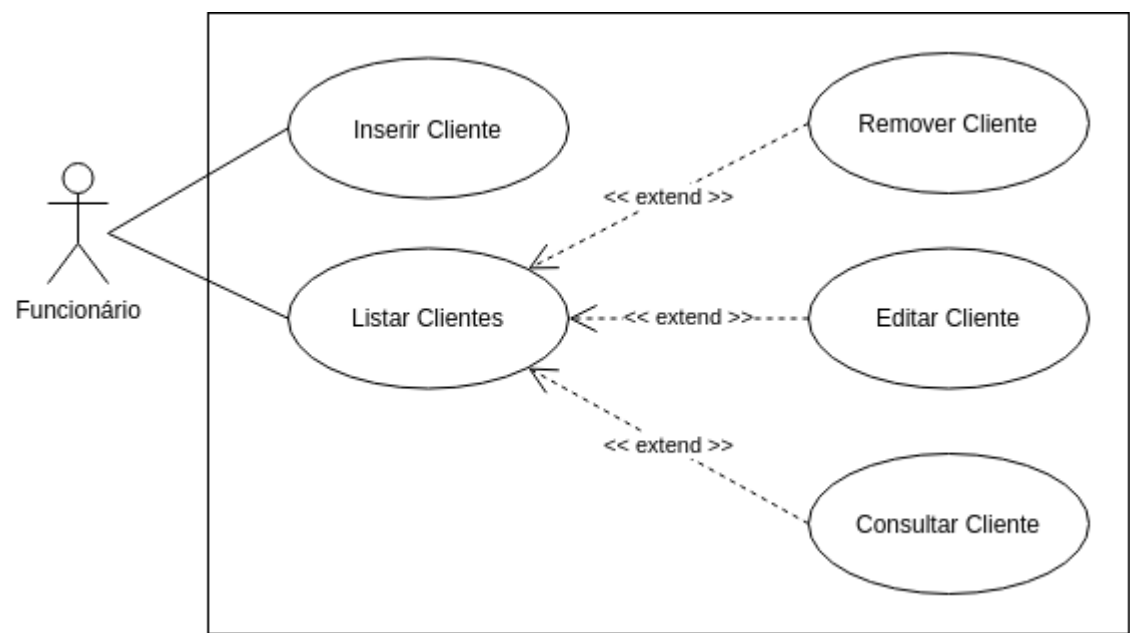
- Relacional/SQL (Necessidade)

2.2.4 UML: Casos de Uso

2.2.4.1 Atores

ATOR	DEFINIÇÃO
Funcionário	O funcionário da empresa, responsável pela área de vendas.

2.2.4.2 Diagrama



2.2.4.3 Casos

Nome	Consultar Cliente
Caso Geral	
Ator Principal	Funcionário
Atores Secundários	
Resumo	Descreve as etapas de Consulta aos dados de um Cliente
Pré-condições	Ter efetuado o caso de uso Listar Clientes, para selecionar o cliente desejado
Pós-condições	
Ações do Ator	Ações do Sistema
1. O ator inicia pressionando “Consultar Cliente” após marcar qual cliente deseja;	
	2. O sistema oferece a interface de consulta de clientes com as informações do cliente selecionado;
Restrições/Validações	1. O cliente precisa existir no banco.
	2. Os dados do cliente não poderão ser alterados neste caso.

Nome	Editar Cliente
Caso Geral	
Ator Principal	Funcionário
Atores Secundários	
Resumo	Descreve as etapas da Edição dos dados de um Cliente
Pré-condições	Ter efetuado o caso de uso Listar Clientes, para selecionar o cliente desejado
Pós-condições	
Ações do Ator	Ações do Sistema
1. O ator pressiona editar cliente, após marcar qual cliente deseja;	
	2. O sistema apresenta o layout com as informações a serem alteradas;
3. O ator edita as informações que quer alterar;	
4. O ator seleciona o botão "salvar";	
	5. O sistema salva as informações do cliente no depósito de dados;
Restrições/Validações	1. O cliente precisa existir no banco.
	2. Os campos CNPJ, Inscrição Estadual e Inscrição Municipal devem ser validados conforme algoritmo, enquanto os outros campos sofrem validação básica de tamanho máximo de caracteres e pontuação.

Nome	Remover Cliente
Caso Geral	
Ator Principal	Funcionário
Atores Secundários	
Resumo	Descreve as etapas da Remoção dos dados de um Cliente
Pré-condições	Ter efetuado o caso de uso Listar Clientes, para selecionar o cliente desejado
Pós-condições	
Ações do Ator	Ações do Sistema
1. O ator pressiona remover cliente, após marcar qual cliente deseja;	
	2. O sistema questiona se o Funcionário realmente deseja remover o cliente;
3. O ator seleciona a opção “sim”, realmente desejo;	
	4. O sistema exclui as informações do cliente do depósito de dados;
Restrições/Validações	1. O cliente precisa existir no banco.

Nome	Listar Cliente
Caso Geral	
Ator Principal	Funcionário
Atores Secundários	
Resumo	Descreve as etapas para Listar os Clientes disponíveis a partir do depósito de dados
Pré-condições	
Pós-condições	
Ações do Ator	Ações do Sistema
1. O ator inicia o caso de uso selecionando "Listar Cliente";	
	2. O sistema oferece a interface com a lista de clientes;
3. O ator seleciona o filtro de CNPJ ou Razão Social;	
	4. O sistema exibe a lista de clientes conforme o filtro selecionado pelo ator;
Restrições/Validações	

Nome	Inserir Cliente
Caso Geral	
Ator Principal	Funcionário
Atores Secundários	
Resumo	Descreve as etapas para Inserção de um Cliente no depósito de dados
Pré-condições	
Pós-condições	
Ações do Ator	Ações do Sistema
1. O ator inicia o caso de uso selecionando “Inserir Cliente”;	
	2. O sistema oferece a interface para inserção de clientes;
3. O ator realiza a inserção de informações necessárias para o “cadastro do cliente”;	
4. O ator confirma o cadastro do cliente e pressiona o botão “Salvar”;	
	5. O sistema valida se as informações inseridas estão de acordo com os critérios necessários;
	6. O sistema insere as informações do novo cliente no depósito de dados.
	7. O sistema informa que o cadastro foi realizado com sucesso;
Restrições/Validações	1. Os campos obrigatórios devem ser informados.
	2. Os dados inseridos devem ser válidos e no formato correto.
	3. Os campos CNPJ, Inscrição Estadual e Inscrição Municipal devem ser validados conforme algoritmo, enquanto os outros campos sofrem validação básica de tamanho máximo de caracteres e pontuação.

2.3 Dicionário de Dados (DD)

Em ordem ALFABÉTICA CRESCENTE por item

Quadro 1 – Dicionário de Dados

SEQ.	ITEM	DEFINIÇÃO	OBSERVAÇÕES
01	_booleano	[False True]	
02	_caractere	[_digito _letra _simbolo]	
03	_digito	[0 1 2 3 4 5 6 7 8 9]	
04	_id	{_digito}	AUTO INCREMENT
05	_letra	[a-z A-Z]	
06	_simbolo	[! @ # \$ % ^ & * () _ +]	
07	_numero	{_digito}	
08	_texto	{_caractere}	
09	bairro_cliente	_texto	
10	cep_cliente	_numero	

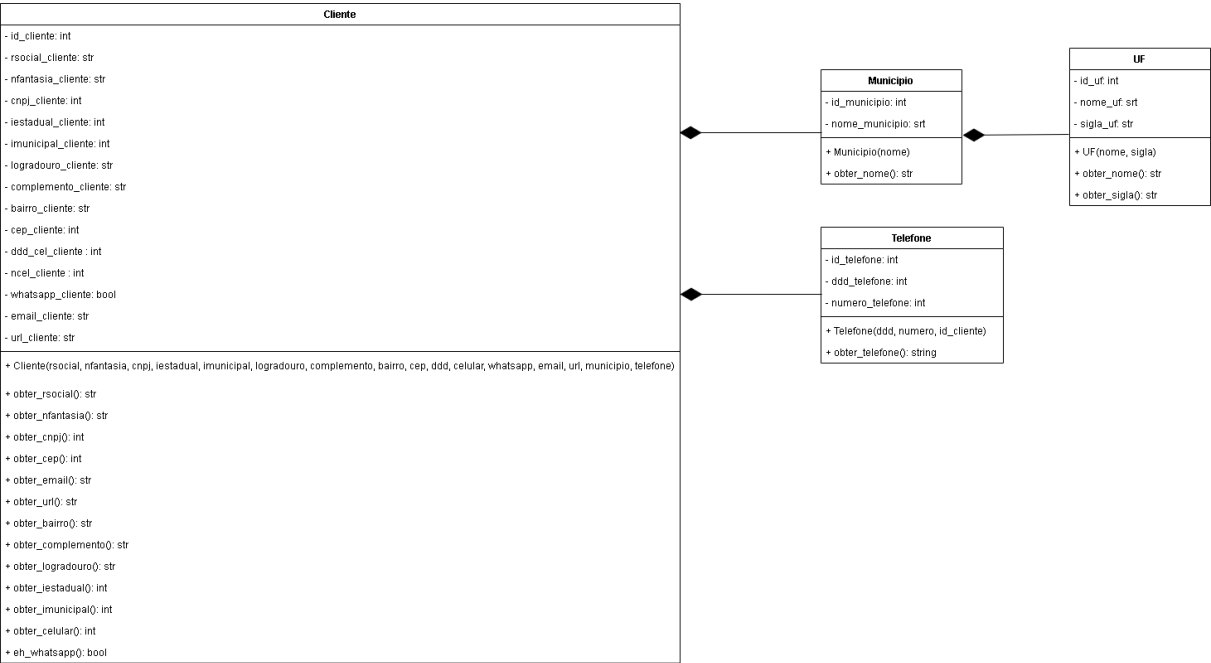
11	CLIENTE	{dado_cliente}	Depósito de Dados
12	cliente_id_telefone	id_cliente	FK(CLIENTE)
13	cnpj_cliente	_numero	
14	complemento_cliente	_texto	Opcional
15	dado_cliente	id_cliente +	Estrutura de Dados
		rsocial_cliente +	
		nfantasia_cliente +	
		cnpj_cliente +	
		iestadual_cliente +	
		imunicipal_cliente +	
		logradouro_cliente +	
		complemento_cliente +	
		bairro_cliente +	
		cep_cliente +	
		ddd_cel_cliente +	
		ncel_cliente +	
		whatsapp_cliente +	
		email_cliente +	
		url_cliente +	
		municipio_id_cliente	

16	dado_municipio	id_municipio + nome_municipio + uf_id_municipio	Estrutura de Dados
17	dado_telefone	id_telefone + ddd_telefone + numero_telefone + cliente_id_telefone	Estrutura de Dados
18	dado_uf	id_uf + nome_uf + sigla_uf	Estrutura de Dados
19	ddd	_numero	
20	ddd_cel_cliente	ddd	
21	ddd_telefone	ddd	
22	whatsapp_cliente	_booleano	
23	email_cliente	_texto	
24	id_cliente	_id	PK(CLIENTE)
25	id_municipio	_id	PK(MUNICIPIO)

26	id_telefone	_id	PK(TELEFONE)
27	id_uf	_id	PK(UF)
28	iestadual	_numero	Opcional
29	iestadual_cliente	iestadual	Formato: 999.999.999.999
30	imunicipal	_numero	Opcional
31	imunicipal_cliente	imunicipal	Formato: 9.999.999-9
32	logradouro_cliente	_texto	
33	nome	_texto	
34	nome_municipio	nome	
35	nome_uf	nome	
36	nfantasia_cliente	nome	
37	ncel_cliente	_numero	Formato: (99) 9 9999-9999
38	numero_telefone	_numero	Formato: (99) 9999-9999
39	MUNICIPIO	{dado_municipio}	Depósito de Dados

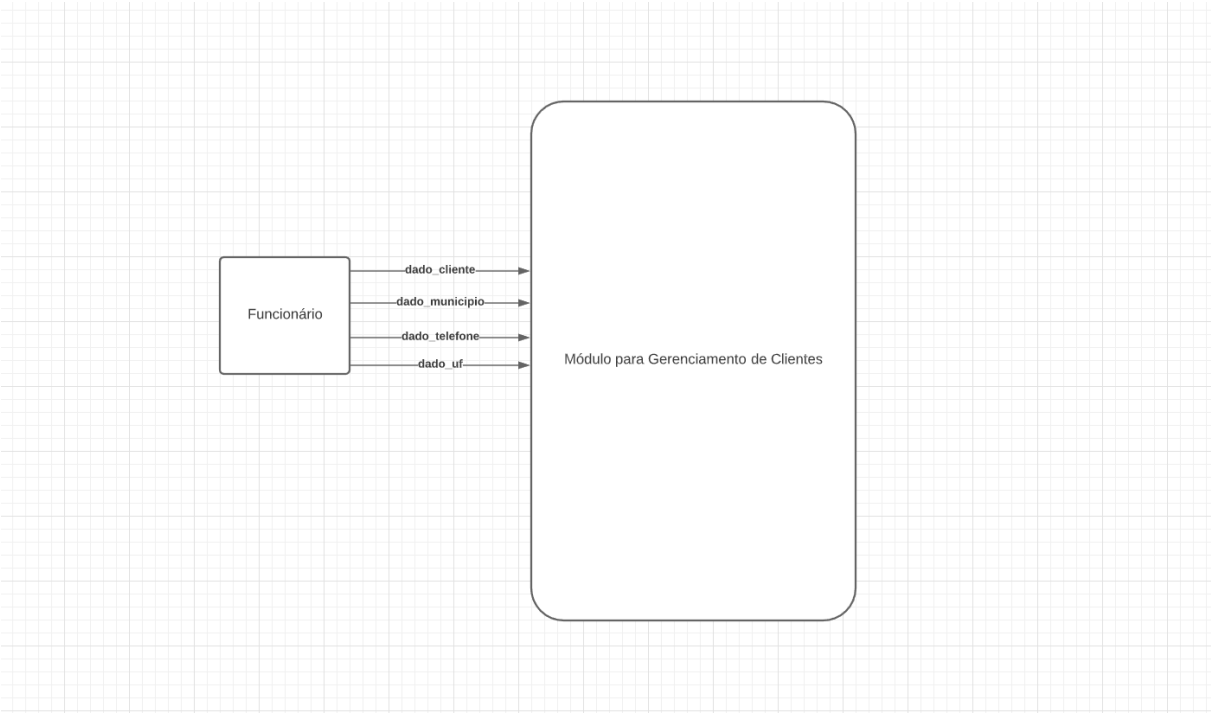
40	municipio_id_cliente	id_municipio	FK(MUNICIPIO)
41	rsocial_cliente	nome	
42	sigla_uf	{_letra}2	UK
43	TELEFONE	{dado_telefone}	Depósito de Dados
44	UF	{dado_uf}	Depósito de Dados
45	uf_id_municipio	id_uf	FK(UF)
46	url_cliente	_texto	Opcional

2.4 UML: Diagrama de Classes

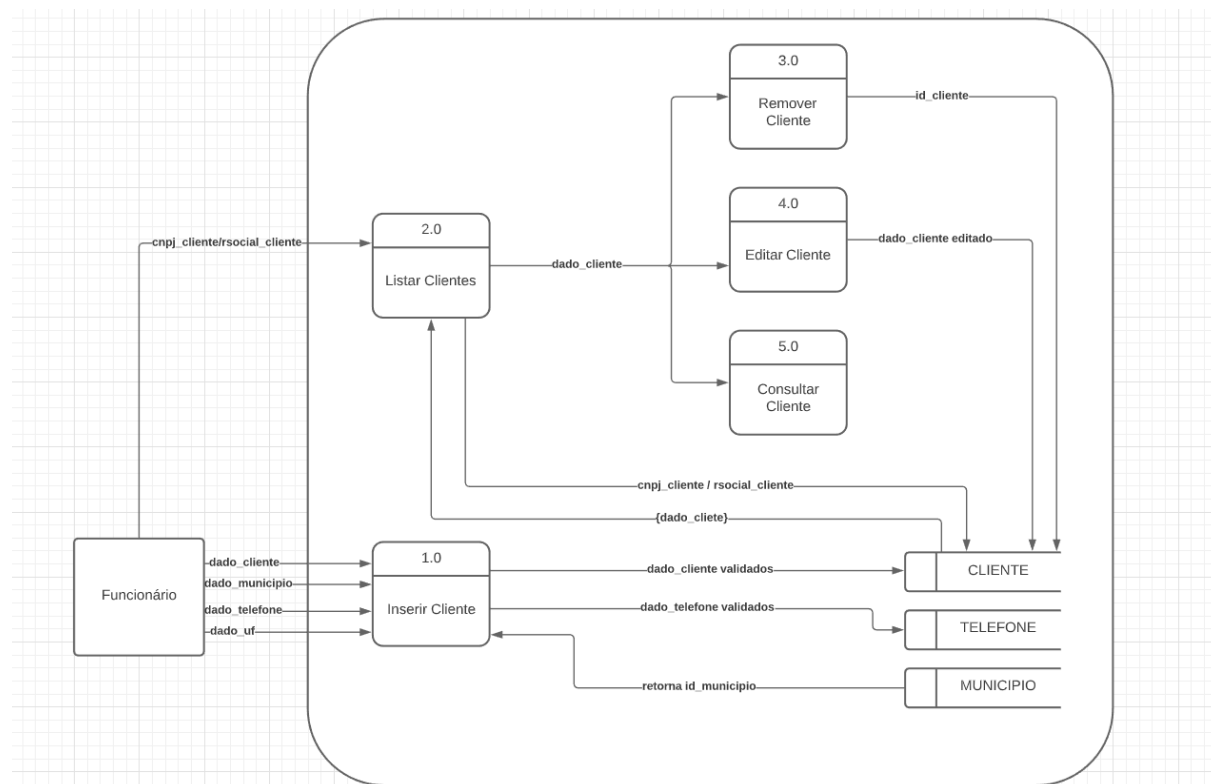


2.5 Diagrama de Fluxo de Dados (DFD)

2.5.1 Nível 0



2.5.2 Nivel 1



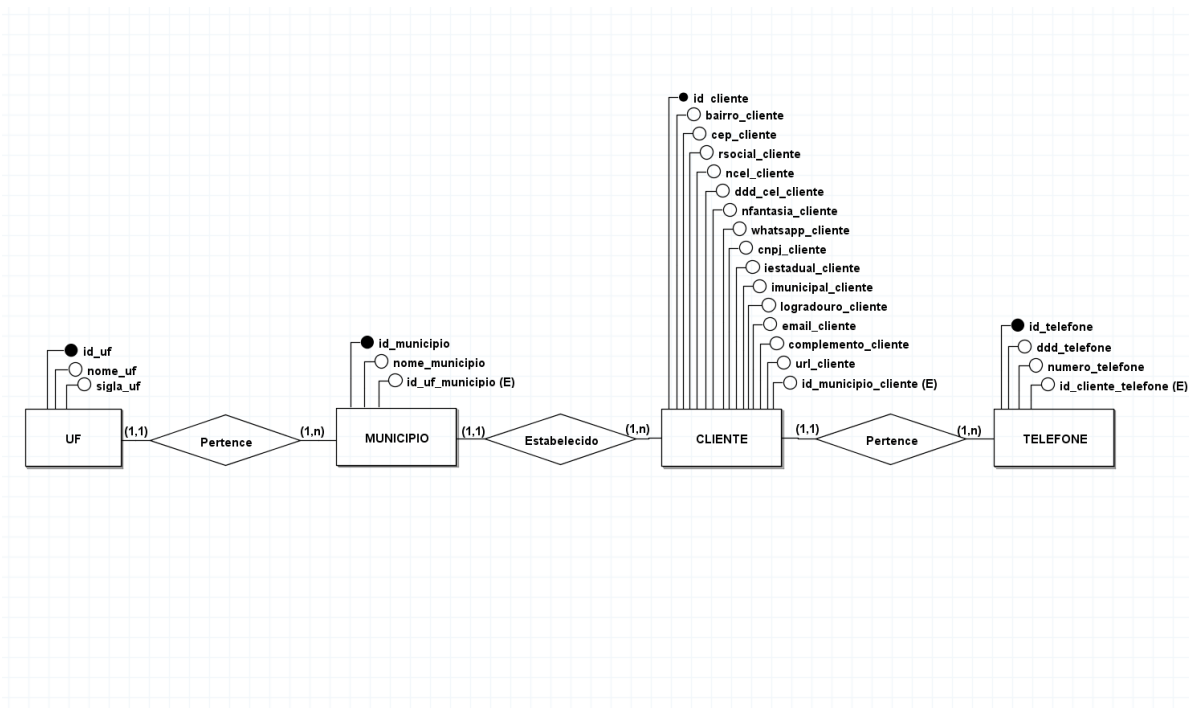
2.5.3 Nivel 2 Inserir Cliente

2.5.4 Nivel 2 Editar Cliente

3 PROJETO DE BANCO DE DADOS

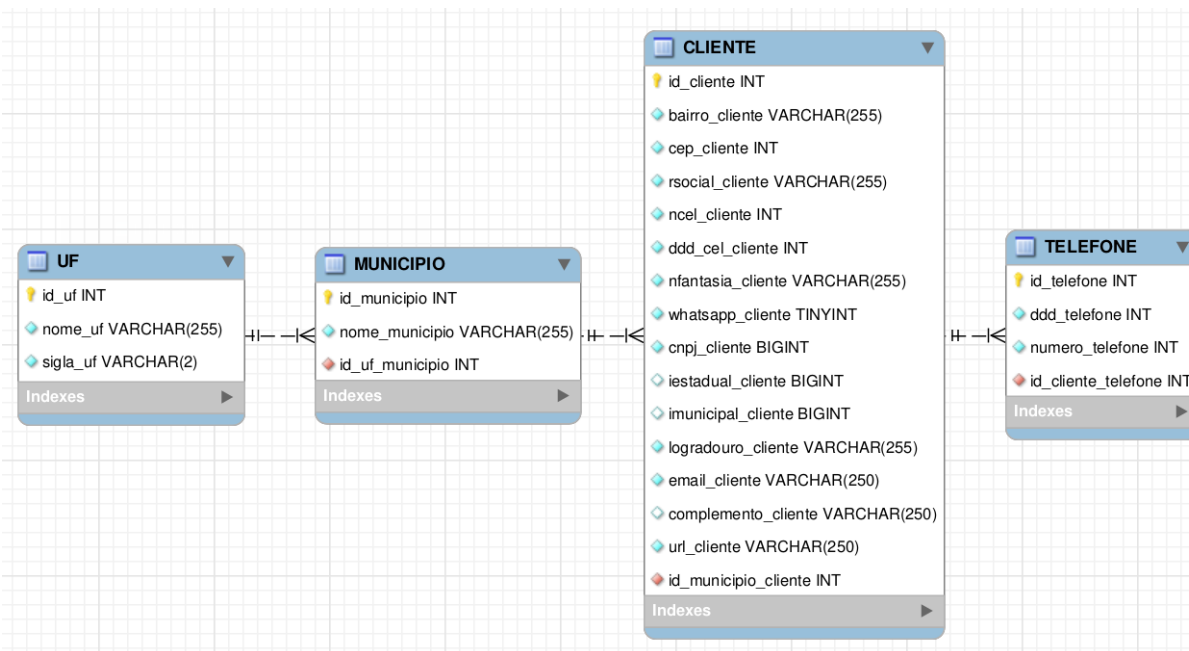
3.1 Projeto Conceitual

DER na Notação Peter CHEN



3.2 Projeto Lógico

DER na Notação Pé-de-Galinha



3.3 Projeto Físico

```
/* Create sci_db Database */
```

```
CREATE DATABASE `sci_db`;
```

```
/* Create UF Table */
```

```
CREATE TABLE IF NOT EXISTS `sci_db`.`UF` (  
  `id_uf` INT NOT NULL AUTO_INCREMENT,  
  `nome_uf` VARCHAR(255) NOT NULL,  
  `sigla_uf` VARCHAR(2) NOT NULL,  
  PRIMARY KEY (`id_uf`),  
  UNIQUE INDEX `sigla_uf_UNIQUE` (`sigla_uf` ASC) VISIBLE)  
ENGINE = InnoDB;
```

```
/* Create MUNICIPIO Table */
```

```
CREATE TABLE IF NOT EXISTS `sci_db`.`MUNICIPIO` (  
  `id_municipio` INT NOT NULL AUTO_INCREMENT,  
  `nome_municipio` VARCHAR(255) NOT NULL,  
  `id_uf_municipio` INT NOT NULL,  
  PRIMARY KEY (`id_municipio`),  
  INDEX `fk_MUNICIPIO_1_idx` (`id_uf_municipio` ASC) VISIBLE,  
  CONSTRAINT `fk_MUNICIPIO_1`  
    FOREIGN KEY (`id_uf_municipio`)  
    REFERENCES `sci_db`.`UF` (`id_uf`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
/* Create CLIENTE Table */
```

```
CREATE TABLE IF NOT EXISTS `sci_db`.`CLIENTE` (  
  `id_cliente` INT NOT NULL AUTO_INCREMENT,  
  `bairro_cliente` VARCHAR(255) NOT NULL,
```

```

`cep_cliente` INT NOT NULL,
`rsocial_cliente` VARCHAR(255) NOT NULL,
`ncel_cliente` INT NOT NULL,
`ddd_cel_cliente` INT NOT NULL,
`nfantasia_cliente` VARCHAR(255) NOT NULL,
`whatsapp_cliente` TINYINT NOT NULL,
`cnpj_cliente` BIGINT NOT NULL,
`iestadual_cliente` BIGINT NULL,
`imunicipal_cliente` BIGINT NULL,
`logradouro_cliente` VARCHAR(255) NOT NULL,
`email_cliente` VARCHAR(250) NOT NULL,
`complemento_cliente` VARCHAR(250) NULL,
`url_cliente` VARCHAR(250) NOT NULL,
`id_municipio_cliente` INT NOT NULL,
PRIMARY KEY (`id_cliente`),
INDEX `fk_CLIENTE_1_idx` (`id_municipio_cliente` ASC) VISIBLE,
CONSTRAINT `fk_CLIENTE_1`
  FOREIGN KEY (`id_municipio_cliente`)
    REFERENCES `sci_db`.`MUNICIPIO` (`id_municipio`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

/* Create TELEFONE Table */

```

```

CREATE TABLE IF NOT EXISTS `sci_db`.`TELEFONE` (
  `id_telefone` INT NOT NULL AUTO_INCREMENT,
  `ddd_telefone` INT NOT NULL,
  `numero_telefone` INT NOT NULL,
  `id_cliente_telefone` INT NOT NULL,
  PRIMARY KEY (`id_telefone`),
  INDEX `fk_TELEFONE_1_idx` (`id_cliente_telefone` ASC) VISIBLE,

```

```

CONSTRAINT `fk_TELEFONE_1`

FOREIGN KEY (`id_cliente_telefone`)

REFERENCES `sci_db`.`CLIENTE` (`id_cliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

/* Insert UF entries */

INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Amazonas", "AM");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Pará", "PA");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Mato Grosso", "MT");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Minas Gerais", "MG");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Bahia", "BA");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Mato Grosso do Sul", "MS");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Goiás", "GO");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Maranhão", "MA");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Rio Grande do Sul", "RS");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Tocantins", "TO");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Piauí", "PI");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("São Paulo", "SP");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Rondônia", "RO");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Roraima", "RR");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Paraná", "PR");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Acre", "AC");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Ceará", "CE");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Amapá", "AP");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Pernambuco", "PE");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Santa Catarina", "SC");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Paraíba", "PB");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Rio Grande do Norte", "RN");
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Espírito Santo", "ES");

```

```
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Rio de Janeiro", "RJ");
```

```
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Alagoas", "AL");
```

```
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Segipe", "SE");
```

```
INSERT INTO `sci_db`.`UF` (`nome_uf`, `sigla_uf`) VALUES ("Distrito Federal", "DF");
```

```
/* Insert MUNICIPIO entries */
```

```
INSERT INTO `sci_db`.`MUNICIPIO` (`nome_municipio`, `id_uf_municipio`) VALUES ("Campinas", 12);
```

```
INSERT INTO `sci_db`.`MUNICIPIO` (`nome_municipio`, `id_uf_municipio`) VALUES ("Araraquara", 12);
```

```
INSERT INTO `sci_db`.`MUNICIPIO` (`nome_municipio`, `id_uf_municipio`) VALUES ("São José do Rio Preto", 12);
```

```
INSERT INTO `sci_db`.`MUNICIPIO` (`nome_municipio`, `id_uf_municipio`) VALUES ("Ribeirão Preto", 12);
```

```
INSERT INTO `sci_db`.`MUNICIPIO` (`nome_municipio`, `id_uf_municipio`) VALUES ("São Paulo", 12);
```

```
INSERT INTO `sci_db`.`MUNICIPIO` (`nome_municipio`, `id_uf_municipio`) VALUES ("Hortolandia", 12);
```

```
INSERT INTO `sci_db`.`MUNICIPIO` (`nome_municipio`, `id_uf_municipio`) VALUES ("Valinhos", 12);
```

```
/* Insert CLIENTE entries */
```

```
INSERT INTO `sci_db`.`CLIENTE`
```

```
(`bairro_cliente`,
```

```
`cep_cliente`,
```

```
`rsocial_cliente`,
```

```
`ncel_cliente`,
```

```
`ddd_cel_cliente`,
```

```
`nfantasia_cliente`,
```

```
`whatsapp_cliente`,
```

```
`cnpj_cliente`,
```

```
`iestadual_cliente`,
```

```
`imunicipal_cliente`,
```

```
`logradouro_cliente`,
`email_cliente`,
`complemento_cliente`,
`url_cliente`,
`id_municipio_cliente`)
```

```
VALUES
```

```
("Bairro 3M",
35655972,
"3M do Brasil",
99327426,
19,
"3M",
0,
90178983000143,
799169523181,
99169523181,
"Rua 3M",
"gerente@3m.com.br",
"Recepção Norte",
"www.3m.com.br",
1);
```

```
/* Insert TELEFONE entries */
```

```
INSERT INTO `sci_db`.`TELEFONE` (`ddd_telefone`, `numero_telefone`, `id_cliente_telefone`)
VALUES (19, 32456477, 1);
```

```
INSERT INTO `sci_db`.`TELEFONE` (`ddd_telefone`, `numero_telefone`, `id_cliente_telefone`)
VALUES (19, 42726843, 1);
```

```
INSERT INTO `sci_db`.`TELEFONE` (`ddd_telefone`, `numero_telefone`, `id_cliente_telefone`)
VALUES (19, 52233496, 1);
```

4 IMPLANTAÇÃO

4.1 Definição das Tabelas no Banco de Dados



4.2 Dados das Tabelas

CLIENTE														
#	id_bairro_cli	cep_cliente	rsocial_cliente	ncel_cliente	ddd_cel	nfantasia	whatsapp	cnpi_cliente	iestadual_cliente	imunicipal_cliente	logradouro_email_cliente	complemento_c	url_cliente	id_municipio
1	Bairro 3M	35655972	3M do Brasil	99327426	19	3M	0	90178983000143	799169523181	99169523181	Rua 3M	gerente@3m.com.br	Recepção Norte	www.3m.com.br
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														

UF			
#	id_uf	nome_uf	sigla_uf
1	1	Amazonas	AM
2	2	Pará	PA
3	3	Mato Grosso	MT
4	4	Minas Gerais	MG
5	5	Bahia	BA
6	6	Mato Grosso do Sul	MS
7	7	Goiás	GO
8	8	Maranhão	MA
9	9	Rio Grande do Sul	RS
10	10	Tocantins	TO
11	11	Piauí	PI
12	12	São Paulo	SP
13	13	Rondônia	RO
14	14	Roraima	RR
15	15	Paraná	PR
16	16	Acre	AC
17	17	Ceará	CE
18	18	Amapá	AP
19	19	Pernambuco	PE
20	20	Santa Catarina	SC
21	21	Paraíba	PB
22	22	Rio Grande do N...	RN
23	23	Espírito Santo	ES
24	24	Rio de Janeiro	RJ
25	25	Alagoas	AL
26	26	Segipe	SE
27	27	Distrito Federal	DF

TELEFONE			
#	id_telefone	ddd_telefone	numero_telefone
1	1	19	32456477
2	2	19	42726843
3	3	19	52233496
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			

MUNICIPIO		
#	id_municipio	nome_municipio
1	1	Campinas
2	2	Araraquara
3	3	São José do Rio Preto
4	4	Ribeirão Preto
5	5	São Paulo
6	6	Hortolândia
7	7	Valinhos
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		

4.3 Graphical User Interface (GUI)

Cole, neste tópico, as imagens das telas (GUIs) desenvolvidas para o sistema. Posteriormente, remova estas instruções.

4.4 Versão **Funcional** Final do Aplicativo

Arquivo  python a parte.

5 REGISTRO DE REUNIÕES

SEQ.	DATA	DISCIPLINA	ASSUNTO
01	10/09/2020	AS	Requisitos/Interface Gráfica
02	18/09/2020	ESW	Requisitos não funcionais
03	21/09/2020	AS	Diagrama de Classes/Dicionário de Dados/Modelo Entidade Relacionamento
04	22/09/2020	AS	Dicionário de Dados/Modelo Entidade Relacionamento

6 CONCLUSÃO

Os objetivos foram atingidos? Quais foram os resultados obtidos? A que conclusão se chegou. Todas as questões iniciais foram respondidas? O que foi aprendido? Quais as dificuldades encontradas? **O uso de 1ª pessoa NÃO é permitido nesta seção.** Use quebra de seção entre capítulos. **Não utilize recuos nos parágrafos.** Utilize a estrutura de três parágrafos (introdução-desenvolvimento-conclusão). Em geral, uma página mas não mais do que duas. **Remova estas instruções.**

REFERÊNCIAS

ICS. Github, 2020. Disponível em: < <https://github.com/wendellbrito/ICS/tree/master/doc> >. Acesso em: 27 de set. de 2020.