



**PROJETO DE DISCIPLINA:
PROJETO DE SISTEMAS WEB**

Wanderson Pereira da Silva Lira

**SERRA TALHADA, PE
02/06/2019**

Wanderson Pereira da Silva Lira

Projeto desenvolvido para efetivação da [1ª ou 2ª ou 3ª] etapa apresentado para avaliação na Disciplina Projeto de Sistemas WEB ministrada pelo Prof. Ygor Amaral Barbosa Leite de Sena, período letivo 2019.1.

01. DEFINIÇÃO DA PROBLEMÁTICA (DOMÍNIO DA APLICAÇÃO)

A empresa fornece serviços de manutenção domiciliar/empresarial, do tipo faz-tudo, desde limpezas a manutenções, reformas, concertos, entre outros. Entretanto, têm o diferencial de os prestadores de serviços serem terceirizados, os mesmos utilizam da empresa apenas para os indexar e direcioná-los aos clientes da sua região. Os clientes por outro lado têm um acesso rápido e fácil, as pessoas que vão resolver seu problema com eficiência, caso isso não ocorra também os é possibilitado classificar o serviço prestado de acordo com o seu grau de satisfação, assim é dado aos clientes uma visão do quão bom o prestador de serviço é em desempenhar suas funções, levando também à liberdade de optar por quem fará o serviço baseado nesse parâmetro.

A empresa busca através do desenvolvimento do software uma solução computacional que possa ser utilizada para solicitação de serviços de manutenção, seja domiciliar ou empresarial. Para isso se faz necessário a persistência de dados referentes aos tipos de serviços que a empresa presta, como também os respectivos funcionários que podem exercê-los, esses os quais são tratados como Micro Empreendedores Individuais(MEI), ou seja pessoas terceirizadas. Cada MEI deve definir todos os tipos de serviços que pode fornecer , como também o valor que cobra para exercê-lo, Assim o Cliente pode buscar o prestador de serviço que está mais em conta na sua região. O MEI também deve definir as localidades que pode atender. Também se faz necessário armazenar o Cliente com informações básicas, pessoais e de contato para qualquer inconveniente, avisos, ou utilização desses dados para por ex. marketing direcionado. O cliente ao logar têm a possibilidade de selecionar os serviços que deseja requisitar como também data e horário para que cada um ocorra.

02. DEFINIÇÃO DE REQUISITOS

2.1. Requisitos Funcionais:

Identificação:	RF01 – Manter Endereço
Descrição:	<i>Deve ser possibilitada a manutenção(cadastro, exclusão, busca, e edição) do endereço de clientes.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF02 – Manter Contato
Descrição:	<i>Deve ser possibilitada a manutenção(cadastro, exclusão, busca, e edição) dos contatos do Clientes.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF03 – Manter Serviço
Descrição:	<i>Deve ser possibilitada a manutenção(cadastro, exclusão, busca, e edição) dos serviços prestados pela empresa.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF04 – Manter Usuário
Descrição:	<i>Deve ser possibilitada a manutenção(cadastro, exclusão, busca, e edição) dos Usuários do Sistema.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF05 – Manter Cidade
Descrição:	<i>Deve ser possibilitada a manutenção(cadastro, exclusão, busca, e edição) das Cidades os MEIs poderão se cadastrar como sendo sua área de troca(trade area, localidade onde presta seus serviços)</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF06 – Manter Administradores
Descrição:	<i>Deve ser possibilitada a manutenção(cadastro, exclusão, busca, e edição) dos administradores do sistema.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF07 – Manter Cliente
Descrição:	<i>Deve ser possibilitada a manutenção(cadastro, exclusão, busca, e edição) de clientes.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF08 – Manter MEI
Descrição:	<i>eve ser possibilitada a manutenção(cadastro, exclusão, busca, e edição) de MEIs.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF09 – Manter Serviços MEI
Descrição:	<i>Deve ser possibilitada a manutenção(cadastro, exclusão, busca, e edição) dos Serviços que são prestados pelo MEI.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF10 – Manter Solicitação
-----------------------	----------------------------------

Descrição:	<i>Deve ser possibilitada a manutenção(cadastro, exclusão, busca, e edição) das Solicitações de serviços dos clientes.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF11 – Acesso ao Sistema
Descrição:	<i>O sistema deve possibilitar entrada(login) e saída(logout) de usuários.</i>
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	RF12 – Buscar Serviços Disponíveis por Região
Descrição:	<i>O Sistema deve possibilitar a busca por serviços que estão sendo prestados por região, os quais consistem em uma sub-lista de serviços que a empresa presta, constituída pelos serviços que são prestados pelos MEI que atendem na respectiva região.</i>
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	RF13: Somar Valores de Serviços
Descrição:	<i>o sistema pode somar o valor de todos os serviços para expressar, como por exemplo, o total que uma solicitação de serviço terá, em valor.</i>
Prioridade:	() Essencial () Importante (x) Desejável

Identificação:	RF14: Buscar MEI por Região e Serviço
Descrição:	<i>O sistema deve possibilitar busca por MEIs de determinada região que prestam um serviço e os ordenar pela pontuação que têm em fazer aquele serviço, ou valor que cobra, ou ambos.</i>
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	RF015 – Solicitar Serviço
Descrição:	<i>Através do endereço (que pode ser obtido através da lista de endereços do cliente ou de outro endereço informado na hora de solicitação) informado pelo cliente para que o serviço seja feito é obtido uma lista de serviços disponíveis na região (RF12), O cliente pode selecionar cada serviço que deseja requisitar e horário para que isso ocorra. Enquanto isso ,o é</i>

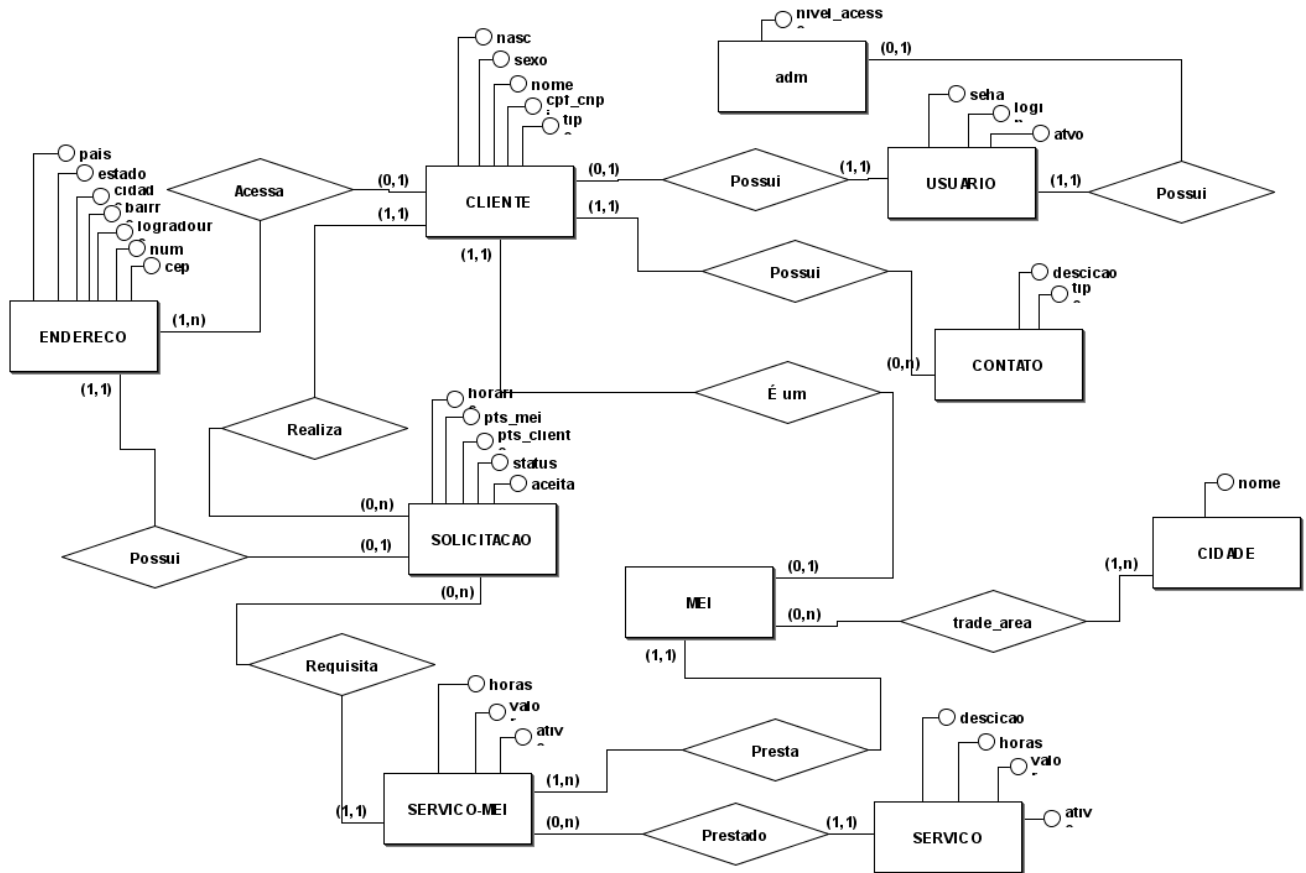
	<i>mostrado o valor padrão que o mesmo poderia pagar, que é obtido através da soma de todos os valores padrões dos serviços que escolheu (RF13), em seguida, o cliente seleciona para cada serviço o MEI que deseja que realize (RF14). Por fim, é mostrado uma lista contendo o serviço a ser feito, o MEI que o fará, valores associados a cada um e horário que ocorrerá, como também o total por todos os serviços, de acordo com os valores atualizados dos MEI.</i>
Prioridade:	(x) Essencial () Importante () Desejável

Identificação:	RF016 – Confirmar Solicitação
Descrição:	<i>O MEI não terá acesso a todas as solicitações de serviço em sua cidade, mas sim a aquelas direcionadas especificamente a ele, com isso, terá a liberdade de aceitar ou não uma solicitação de serviço direcionada. Portanto, dependendo da resposta a solicitação passará para seu respectivo estado (<u>aceita/recusada</u>).</i>
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	RF017 – Cancelar Solicitação
Descrição:	<i>Tanto o Cliente como o MEI podem a qualquer momento, se a solicitação ainda não foi <u>concluída</u>, cancelá-la. Entretanto, caso seja o MEI a <u>cancelar</u>, essa solicitação deve passar para o estado de <u>recusada</u>, assim outro Cliente poderá (caso queira) selecionar outra pessoa para fazer o serviço, como também pontuar o MEI de acordo com seu grau de satisfação.</i>
Prioridade:	() Essencial (x) Importante () Desejável

Identificação:	RF018 – Concluir Solicitação / Finalizar Serviço
Descrição:	<i>O sistema deve possibilitar a conclusão de uma solicitação. Nesse momento tanto o Cliente, como o MEI, pontuam um ao outro, a fim de definir o grau de satisfação. Para o Cliente esse grau se refere ao quanto gostou do serviço do MEI, e para o MEI o quanto o Cliente favoreceu o bom “andar” do serviço. Por fim, a solicitação deve ser definida como concluída.</i>
Prioridade:	(x) Essencial () Importante () Desejável

03. DIAGRAMA ENTIDADE-RELACIONAMENTO



04. MAPEAMENTO DO ESQUEMA CONCEITUAL PARA O ESQUEMA LÓGICO-RELACIONAL

Tabela	Atributos
USUARIO	(<u>id</u> , ativo, login, senha);
SERVICO	(<u>id</u> , ativo, valor, horas, descricao);
CIDADE	(<u>id</u> , nome);
ENDERECO	(<u>id</u> , cep, num, logradouro, bairro, cidade, estado, pais, <i>usuario_id</i>);
CONTATO	(<u>id</u> , tipo, descricao, <i>usuario_id</i>);
ADM	(<u>id</u> , gral_acesso, <i>usuario_id</i>);
CLIENTE	(<u>id</u> , tipo, cpf_cnpj, nome, nasc, sexo, <i>usuario_id</i>);
MEI	(<u>id</u> , <i>cliente_id</i>);
MEI_CIDADE	(<u>id</u> , <i>mei_id</i> , <i>cidade_id</i>);
SERVICO_MEI	(<u>id</u> , ativo, valor, horas, <i>servico_id</i> , <i>mei_id</i>);
SOLICITACAO	(<u>id</u> , aceita, descricao, pts_cliente, pts_mei, horario, <i>cliente_id</i> , <i>servico_mei_id</i> , <i>endereco_id</i>);

5.1 DDL

CRIAÇÃO DE BANCO

```
CREATE DATABASE `fdt` /*!40100 DEFAULT CHARACTER SET utf8 */;
```

CRIAÇÃO DE TABELAS

```
CREATE TABLE `usuario` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `ativo` tinyint(1) NOT NULL DEFAULT 1,  
  `login` varchar(100) UNIQUE NOT NULL,  
  `senha` varchar(32) NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
CREATE TABLE `servico` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `ativo` tinyint(1) NOT NULL DEFAULT 1,  
  `valor` float NOT NULL,  
  `horas` int(11) NOT NULL,  
  `descricao` varchar(255) NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
CREATE TABLE `cidade` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `nome` varchar(100) NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
CREATE TABLE `endereco` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `cep` varchar(20) NOT NULL,  
  `num` varchar(20) NOT NULL,  
  `logradouro` varchar(255) NOT NULL,  
  `bairro` varchar(100) NOT NULL,  
  `cidade` varchar(255) NOT NULL,  
  `estado` varchar(100) NOT NULL,  
  `pais` varchar(45) NOT NULL,  
  `usuario_id` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`usuario_id`) REFERENCES `usuario` (`id`)  
);  
  
CREATE TABLE `fdt`.`contato` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `tipo` VARCHAR(20) NOT NULL,  
  `descricao` VARCHAR(255) NULL,  
  `usuario_id` INT NOT NULL,  
  PRIMARY KEY (`id`),
```



```

        FOREIGN KEY (`usuario_id`) REFERENCES `fdt`.`usuario` (`id`)
    );

CREATE TABLE `adm` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `gral_acesso` varchar(20) NOT NULL,
    `usuario_id` int(11) UNIQUE NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`usuario_id`) REFERENCES `usuario` (`id`)
);

CREATE TABLE `fdt`.`cliente` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `tipo` VARCHAR(20) NOT NULL,
    `cpf_cnpj` VARCHAR(45) NOT NULL,
    `nome` VARCHAR(255) NOT NULL,
    `nasc` DATE NULL,
    `sexo` CHAR(1) NULL,
    `usuario_id` INT UNIQUE NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`usuario_id`) REFERENCES `fdt`.`usuario` (`id`)
);

CREATE TABLE `fdt`.`mei` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `cliente_id` INT UNIQUE NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`cliente_id`) REFERENCES `fdt`.`cliente` (`id`)
);

CREATE TABLE `fdt`.`mei_cidade` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `mei_id` INT NOT NULL,
    `cidade_id` INT NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`mei_id`) REFERENCES `fdt`.`mei` (`id`),
    FOREIGN KEY (`cidade_id`) REFERENCES `fdt`.`cidade` (`id`)
);

CREATE TABLE `fdt`.`servico_mei` (
    `id` INT NOT NULL AUTO_INCREMENT,
    `ativo` TINYINT(1) NOT NULL DEFAULT 1,
    `valor` FLOAT NOT NULL,
    `horas` INT NOT NULL,
    `mei_id` INT NOT NULL,
    `servico_id` INT NOT NULL,
    PRIMARY KEY (`id`),
    FOREIGN KEY (`mei_id`) REFERENCES `fdt`.`mei` (`id`),
    FOREIGN KEY (`servico_id`) REFERENCES `fdt`.`servico` (`id`)
);

CREATE TABLE `fdt`.`solicitacao` (

```

```

`id` INT NOT NULL AUTO_INCREMENT,
`aceita` TINYINT(1) NULL DEFAULT 0,
`estado` VARCHAR(20) NOT NULL,
`pts_cliente` FLOAT NULL,
`pts_mei` FLOAT NULL ,
`horario` TIMESTAMP NOT NULL,
`cliente_id` INT NOT NULL,
`servico_mei_id` INT NOT NULL,
`endereco_id` INT NOT NULL,
PRIMARY KEY (`id`),
FOREIGN KEY (`cliente_id`) REFERENCES `fdt`.`cliente` (`id`),
FOREIGN KEY (`servico_mei_id`) REFERENCES `fdt`.`servico_mei` (`id`),
FOREIGN KEY (`endereco_id`) REFERENCES `fdt`.`endereco` (`id`)
);

```

5.2. DML

Criação de primeiro administrador

```

INSERT INTO `fdt`.`usuario` (`login`, `senha`)
VALUES('admin', 'admin');
INSERT INTO `fdt`.`adm`(`gral_acesso`, `usuario_id`)
VALUES('alto', (select id from usuario where login = 'admin'));

```