



B2. Battle one: First TCC Group Mission

São Paulo

2020

Integrantes

INFO D

Pedro Oliveira Sousa

Nº 38

Sumário

Integrantes	2
Módulos	4
Especificações	5
USE CASE FUNCIONARIO	5

UC-Acesso.001:.....	5
UC-Acesso.002:.....	6
UC-Acesso.003:.....	6
UC-Acesso.004:.....	6
USE CASE CORTE.....	6
UC-Corte.001:.....	6
UC-Corte.002:.....	7
UC-Corte.003:.....	7
USE CASE ESTOQUE.....	7
UC-Estoque.001:.....	7
UC-Estoque.002:.....	8
UC-Consulta.003:.....	8
UC-Remover(acabou).004:.....	8
UC-Remover(danificado).005:.....	8
USE CASE CLIENTE	8
UC-Cliente.001:	8
UC-Cliente.002:	9
UC-Cliente.003:	9
UC-Cliente.004:	9
UC-Cliente.005:	9
USE CASE PAGAMENTO.....	9
UC-Pagamento.001:	9
UC-Pagamento.002:	9
UC-Pagamento.003:	10
Protótipos.....	10
PROTP-001.CadastrarFuncionario	10
PROTP-002.Login	11
PROTP-003.Esqueci a Senha.....	12
004.Alterar a senha	13
PROTP-005.Area de Funcionário	14
001.Cadastra Corte.....	16
002.Corte.Consulta de corte	17
003.Corte.Remover Corte	18
001.Estoque.Inserir no estoque	19
PROT-002.Estoque.Consultar Estoque	20
003.Estoque.Consultar para analise de Estoque.....	21

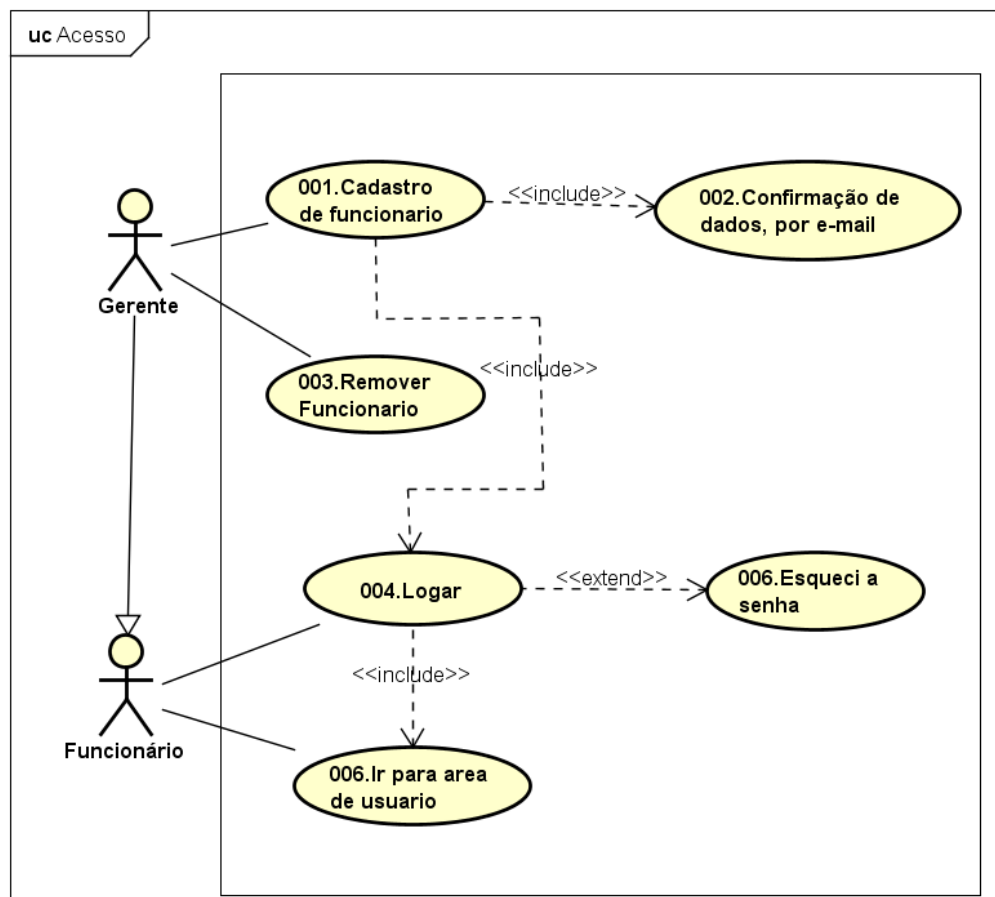
004.Estoque.Remover(acabou).....	22
005.Estoque.Remover(danificada).....	23
001.Cliente.Cadastrar Cliente	24
PROTP-002.Cliente.ConsultarDadosDoCliente.....	25
003.Cliente.ConsultarDadosDoCliente.....	26
003.Cliente.ConsultaHorariosDosClientes	27
001.Pagamento.Pagar	28
MER	30
Modelagem	31
Script	32

Módulos

1. Funcionário
2. Corte
3. Estoque
4. Cliente
5. Pagamento

Especificações

USE CASE FUNCIONARIO



UC-Acesso.001: Os dados necessários para o cadastro são Nome completo, Número dos documentos RG ou CPF, foto 3x4, Currículo, número de telefone, endereço de e-mail, endereço residencial, cep, data de nascimento, pdf com o contrato assinado pelo gerente e contratado. Se não haver registro de usuários com mesmo número de documento e e-mail o sistema abrirá um pagina onde será possível criar um usuário e senha. Senha deve ter 8 caracteres apenas números e letras, não sequenciais

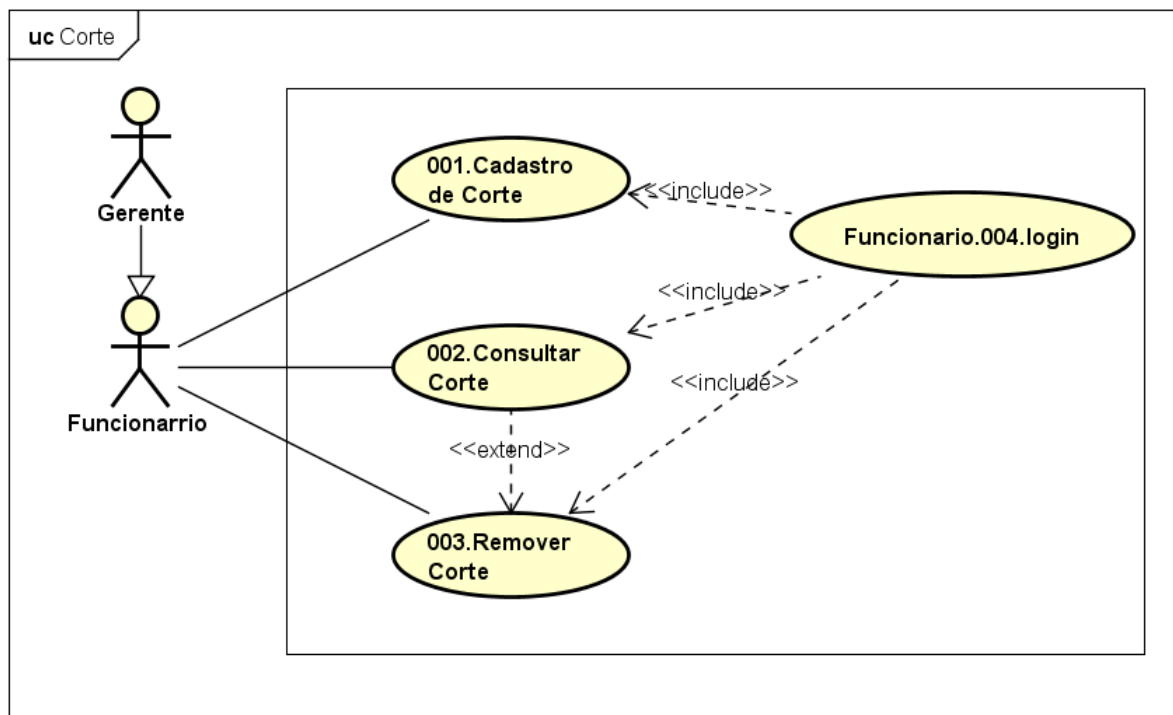
(sequência de 3 ou mais números), sendo que no mínimo deve conter 2 números e 2 letras e usuário não deve possuir um nome já cadastrado no sistema. Sistema enviará um link de confirmação para finalizar o seu cadastro no e-mail cadastrado e contratado deve então digitar o código após a validação usuário deve estar devidamente cadastrado no sistema.

UC-Acesso.002: Funcionário insere o seu login e senha, e após o sistema confirmar seus dados será encaminhado a página "Área do Usuário";

UC-Acesso.003: função de esqueci a senha onde usuário digitara seu CPF e será enviado para uma página de recadastrar a senha, link será válido por 5h;

UC-Acesso.004: Neste trecho será possível o funcionário consultar todos os seus dados. Dia a trabalhar, faltas, total de clientes no dia, mês, ou ano, atualizar documento, alterar residência (esta impacta no vale transporte), nome, e demais itens do cadastro do funcionário, além de possibilitar a o agendamento de férias. Na área de funcionários do gerente ele terá uma aba chamada gestão do estabelecimento onde ele poderá: consultar funcionários escalados para trabalhar no dia, e os que já deram entrada no ponto, o horário de entrada e saída de cada funcionário, estoque;

USE CASE CORTE

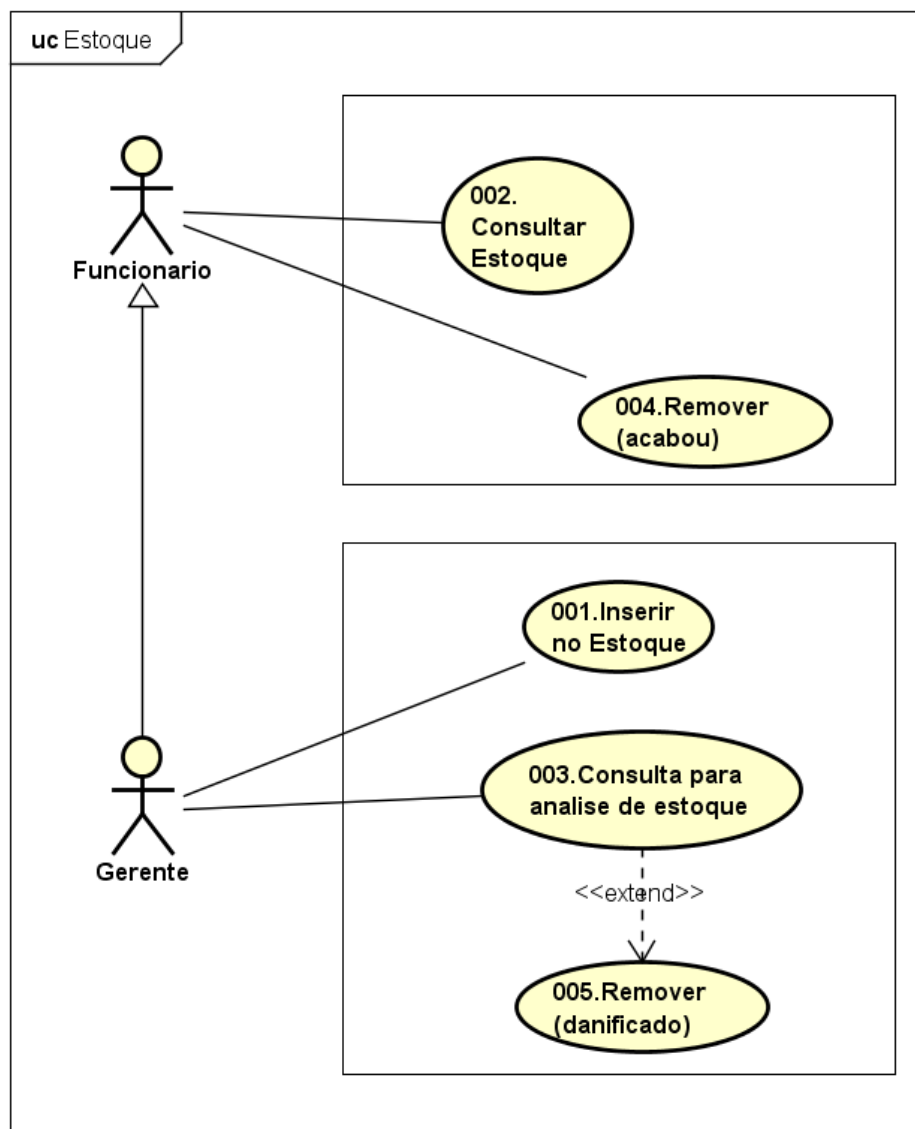


UC-Corte.001: Para cadastra um corte o funcionário de ver inserir os seguintes dados: valor, materiais de preparo, tipos de corte, tempo médio de preparo, fotos, dicas de como fazer, descrições, estilo; sistema deve verificar se o corte já foi cadastrado no sistema pelo nome.

UC-Corte.002: Os funcionários devem poder consultar todos os cortes cadastrados no sistema. Funcionário deve poder filtrar por: data de publicação, tendências, tempo de preparo, nome do corte, avaliação do corte, funcionário que o publicou e estilo;

UC-Corte.003: Deve-se poder remover um corte cadastrado, desde que o usuário seja o mesmo que o cadastrou ou o gerente, essa remoção pode ser a partir de uma busca.

USE CASE ESTOQUE



UC-Estoque.001: Apenas o gerente poderá inserir produtos comprados para a estabelecimento no sistema. Deve sempre ser acompanhado da nota fiscal, quantidade, marca, empresa vendedora, preço Unit., preço total, data de fabricação, data de vencimento, nome do produto.

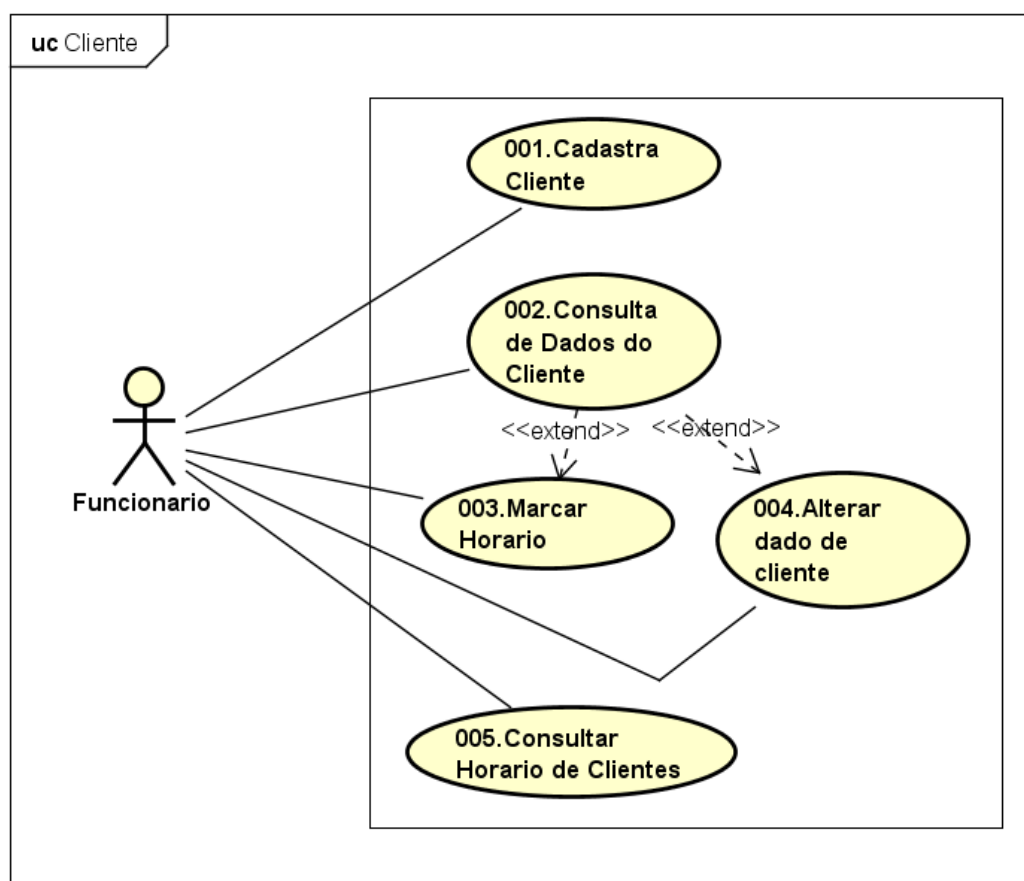
UC-Estoque.002: Nesta área qualquer funcionário tem acesso onde será mostrado a os dados cadastrados e a quantidade sempre atualizada de acordo com as remoções de itens do sistema. Poderá realizar filtros para facilitar a identificação de um produto. Em caso de não identificar os campos inseridos no filtro informar que produto não foi encontrado. Podendo a partir da consulta remover itens.

UC-Consulta.003: Campo específico para gerente onde poderá analisar o estoque e consultar empresas parceira que vendem esse produto, ver quais produtos estão acabando, tempo de ressuprimento, estoque mínimo, estoque máximo.

UC-Remover(acabou).004: Quando um produto acabar os funcionários deve fazer a consulta-lo no sistema, idêntica o produto, dizer a quantidade e assim remove-las do sistema.

UC-Remover(danificado).005: Quando um produto é danificado deve fazer a consulta da matéria, idêntica o produto, dizer a quantidade, motivo da remoção (o defeito) e assim remove-las do sistema.

USE CASE CLIENTE



UC-Cliente.001: Para cadastro de cliente deve haver os seguintes inputs: Nome, CPF, Telefone ou Celular, E-mail. Após preencher esses dados e clicar no botão o cliente

deverá informar o código que chegou em seu e-mail ou mensagem de texto para o número para confirmar o código;

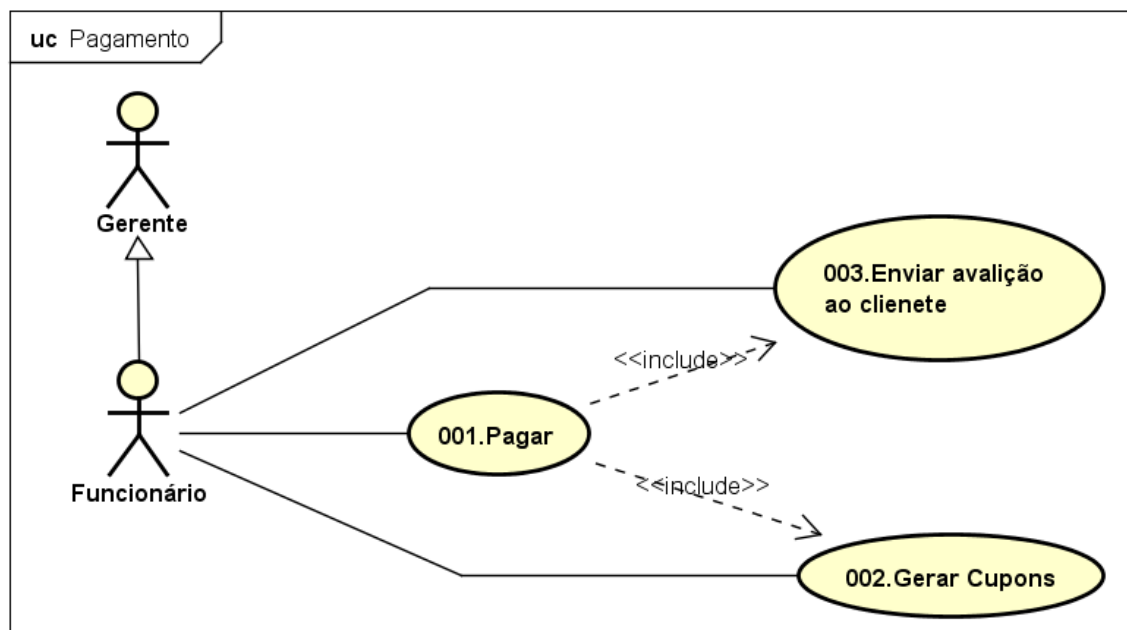
UC-Cliente.002: Consulta e feita pelo nome e cpf e então é mostrado as seguintes informações: horários marcados, nome, número de telefone, e-mail. A partir da consulta deve-se poder marcar horário para cliente e alterara dados do cliente.

UC-Cliente.003: Funcionário deve poder marca no sistema o agendamento do horário que um cliente será atendido, caso necessário deve-se poder alterar e apagar o agendamento. Dados necessário para o agendamento: Nome do Cliente, CPF do cliente, data e horário de atendimento (sistema deve verificar se horário está disponível no sistema antes de efetuas o agendamento).

UC-Cliente.004: Funcionários devem ter permissão para alterar dados cadastrais de um cliente (dados cadastrais UC-Cliente.001).

UC-Cliente.005: Funcionários poderá consultar todos os clientes que ele atendera o dia e os horários em que serão atendidos.

USE CASE PAGAMENTO



UC-Pagamento.001: O funcionário deve consultar o cliente por seu CPF e informar os serviços prestados naquele dia, informar o método de pagamento do cliente (crédito, débito, mercado pago, dinheiro). Cliente deve receber um e-mail com anexo de todas as informações (Valor pago, serviço prestados, nome do funcionário que o atendeu e pedido de avaliação)

UC-Pagamento.002: Após o pagamento o cliente receberá um E-mail onde poderá avaliar de 1 a 10 os seguintes aspectos corte, estabelecimento e atendimento do funcionário. Apenas o gerente poderá ver essas respostas.

UC-Pagamento.003: Cupom deve ser criado vez sim, vez não sedo que não há limite de cupons por clientes, não são acumulativos podemos o cliente ter apenas um cupom por vez, o descende deve ser entre 5% e 15% do valor do pagamento do serviço prestado.

Protótipos

PROTP-001.CadastrarFuncionario

Novos Dados

Cadastrar Funcionario

Nome:

Email:

RG:

RG ou CPF:

Cel:

Endereço

Adicionar Currículo: File

Data de Nascimento: / /

Senha

Foto do funcionario:

Cadastrar

Alert

Digite o codigo que recebeu em seu e-mail ou msg:

Button

Evento: EV-001.CadastrodeFuncionario.Click

Entrada: Nome, E-mail, RG ou CPF, Cel, Endereço, Adicionar Currículo, Data de nascimento, Senha;

Saída: pop-up (Alert de verificação)

Evento: EV-001.CadastrodeFuncionario.Click

Entrada: Código de validação;

Saída: Redireciona para [[Área do funcionário(Gerente)]];

PROTP-002.Login

The diagram illustrates a web browser window with a login form. The browser's address bar shows 'https://'. The page content includes a large square with an 'X' inside, followed by the text 'Login:'. Below this are two input fields: the first is labeled 'Usuário' and the second is labeled 'Senha' with the placeholder text 'Senha Usuário'. A 'Login' button is positioned below the password field. At the bottom of the form, there is a blue hyperlink that reads 'Esqueceu a senha?'. The browser window has a grey header bar with navigation icons (back, forward, stop, home) and a search icon.

Evento: EV-001.Login.Click

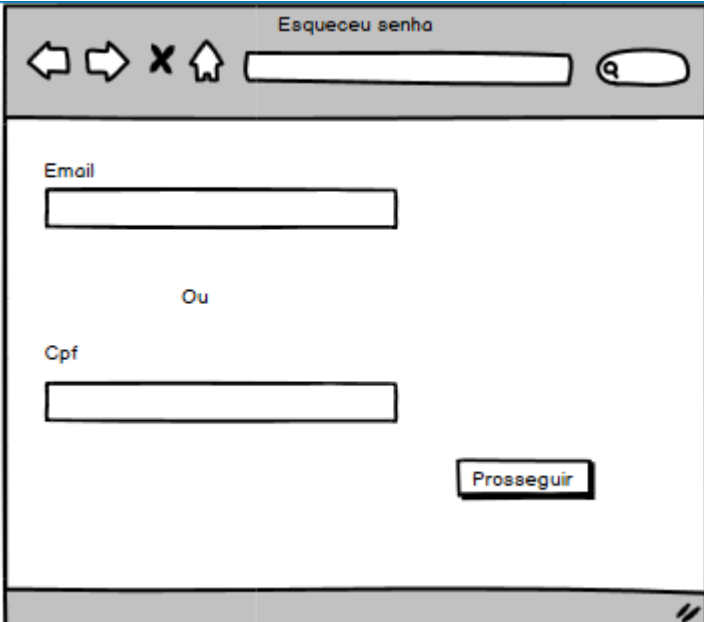
Entrada: e-mail e senha

Saída: Redireciona para [[Área do funcionário]];

Evento: EV-001.EsqueceuaSenha.Click

Saída: Redireciona para [[Esqueci a Senha]];

PROTP-003.Esqueci a Senha



Esqueceu senha

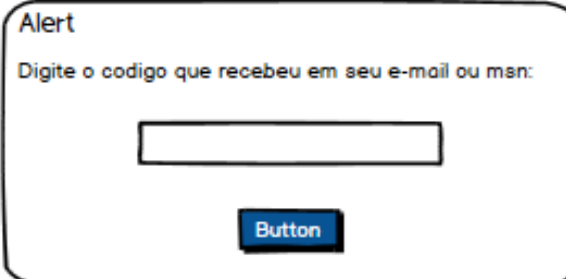
← → × ↶

Email

Ou

Cpf

Prosseguir



Alert

Digite o código que recebeu em seu e-mail ou msg:

Button

Evento: EV-001.Prosseguir.Click

Entrada: e-mail ou cpf

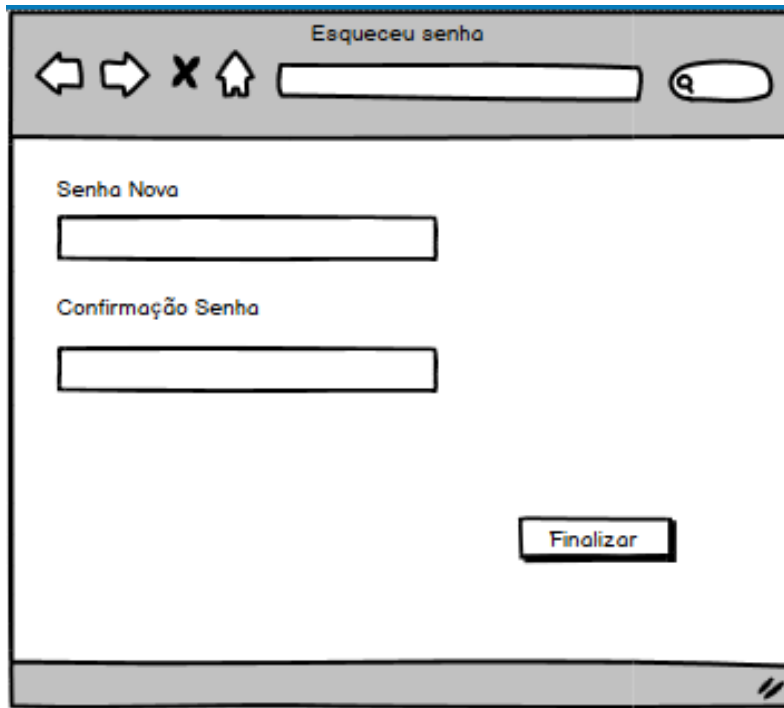
Saída: *pop-up (Alert de verificação)*

Evento: EV-002.CadastrodeFuncionario.Click

Entrada: Código de validação;

Saída: *Redireciona para [[Alterar senha]];*

004.Alterar a senha



A diagrama de uma interface web para a página "Esqueceu senha". No topo, há uma barra cinza com ícones de navegação (setas, X, casa) e uma barra de pesquisa. O formulário principal contém dois campos de entrada: "Senha Nova" e "Confirmação Senha", ambos representados por retângulos brancos. Abaixo dos campos, há um botão "Finalizar" com um contorno preto. A interface é encimada por uma barra cinza com o título "Esqueceu senha" e o rodapé por uma barra cinza com um ícone de engrenagem.

Alert

Senha alterada com sucesso

[Ir para Login](#)

Evento: EV-001.Finalizar.Click

Entrada: nova senha, confirmação da senha(nova)

Saída: *pop-up (Alert de verificação)*

Evento: EV-002.Ir para login.Click

Saída: *Redireciona para [[Login]];*

PROTP-005.Area de Funcionário

← → × ↩

Area funcionario

🔍

DadosFaltas do MêsConsultar CortesCadastro de CortesFérias

Cadastrar Cliente

Consultar Cliente

Remover corte

Pagamento

Gerar Nota Fiscal

Consultar Vale alimentação

Consultar Vale Transporte

Reunião Funcionários-15/06

Informar Motivo da Minha Falta

☐ A fazer

☐ A fazer

Bem vindo Siclano!

[Sair](#)

Clientes do Dia

Terça-15 cortes

Quarta-20 cortes

Quinta-25 cortes

Sexta-29 cortes

Sábado-33 cortes

Domingo-30 cortes

Some text

Evento: EV-001.Sair.Click

Entrada: nova senha, confirmação da senha(nova)

Saída: Redirecionado para [[002.Login]]

Área Gerente

← → ✕ 🏠


Dados
Estoque
Consultar Estoque
Férias

Reunião Funcionários-15/06
Planos para retomada-19/06
Reunião Gerentes-20/06

☐ A fazer

☐ A fazer

☐ A fazer



Bem Vindo Fulano!

Sair

Cientes do dia

Terca-15 cortes
Quarta-20 cortes
Quinta-25 cortes
Sexta-29 cortes
Sábado-33 cortes
Domingo-30 cortes

Cadastrar Funcionários
Dados do funcionários
Faltas
Faltas do funcionários
Gestão do Estabelecimento
Consultar Vale Transporte
Consultar Vale Refeição

O diagrama ilustra a interface de usuário para o cadastro de cortes, dividida em duas versões: a tela inicial e a tela após a interação.

Tela Inicial (Esquerda):

- Barra de Navegação:** Contém ícones de navegação e uma barra de endereço com o texto "https://".
- Formulário:** Possui campos de entrada para "valor:", "materiais de preparo:", "tempo:" e "fotos:" (com três caixas de imagem).
- Descrições:** Uma seção com o rótulo "descrições:" e um campo de texto para inserir descrições.
- Dicas:** Uma seção com o rótulo "dicas:" e um campo de texto para inserir dicas.
- Botão:** Um botão "consultar cortes" localizado na parte inferior.

Tela após a Interação (Direita):

- Alertas:** Duas janelas de alerta são exibidas:
 - Alerta de Sucesso: "Cadastro efetuado com sucesso" com o botão "Fechar".
 - Alerta de Erro: "Erro ao efetuar o cadastro" com os botões "Tentar Novamente" e "Fechar".
- Botão:** O botão "consultar cortes" é substituído por "Cadastrar Corte".

Entrada: Valor, Material de preparo, tempo, imagens, descrição, dicas;

Saída: Fechar - Redirecionamento para *[[área de Consulta de Cortes]]*;

Entrada: Valor, Material de preparo, tempo, imagens, descrição, dicas;

Saída: Tentar Novamente – Reabre tela *[[Cadastrar Corte]]*;

002.Corte.Consulta de corte

The image displays two screenshots of a web application interface for managing haircuts.

Left Screenshot (consultas de cortes):

- Header: consultas de cortes
- Buttons: filtrar, nome do corte, tempo, / / (calendar icon), nome do funcionário que publicou, tendencias, estilo (dropdown), avaliação do corte: (5 stars), cortes cadastrados (dropdown), consultar

Right Screenshot (cadastro de cortes):

- Header: cadastro de cortes
- URL: https://
- Details:
 - valor: 120
 - materias de preparo: (text)
 - tempo: 45min
- Section: fotos (3 placeholder boxes, right arrow icon)
- Section: descrições: (text)
- Section: dicas: (text)

Evento: EV-001.Consultar Cortes. Click

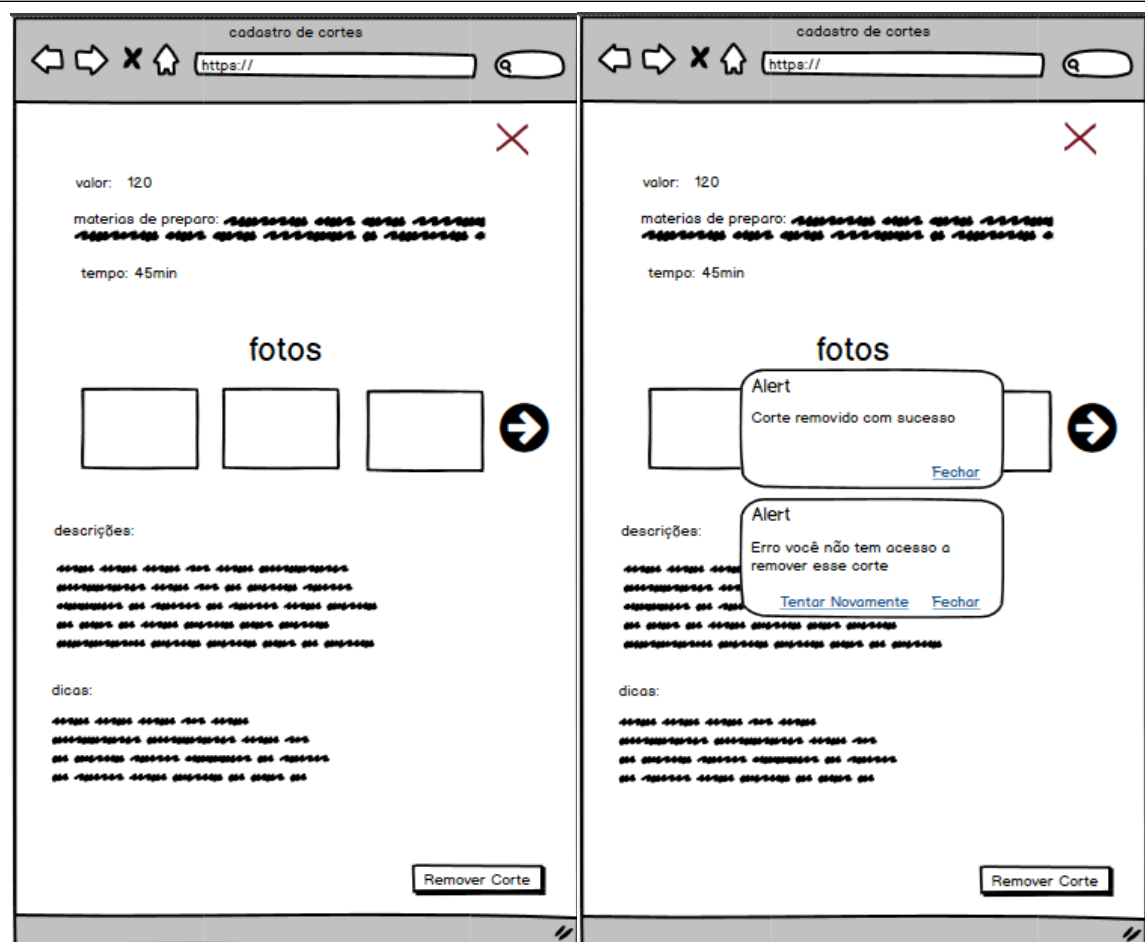
Entrada: nome do corte, tempo, data de publicação, funcionários que a publicou, estilo, avaliação;

Saída: Redirecionamento para *[[Tela de informações do corte]]*

Evento: EV-002.Fechar. Click

Saída: Redirecionamento para *[[retorna para tela de consulta de corte]]*

003.Corte.Remover Corte



Evento: EV-001.RemoverCorte. Click

Saída: Redirecionamento para *[[retorna para tela de consulta de corte]]*

[001.Estoque.Inserir no estoque](#)

A Web Page

https:// ad

Inserir produto no estoque

NF-e:

Quantidade:

Marca:

Preço Unit:

Preço total:

Fabricação:

Vencimento:

Produto:

Evento: EV-001.CadastroCorte. Click

Entrada: Nota Fiscal, Quantidade de itens, Marca, Preço Unit., Preço Total, Fabricação, Vencimento e Produto

Saída: Redireciona para [[área de consultar Estoque]]

PROT-002.Estoque.Consultar Estoque


A Web Page


https://



Consultar estoque

Marca: Ou Inicial:

Produto:

Data cadastrado: 

Data vencimento: 

Qtd em estoque:  

Média de valor:

A Web Page

https://

Dados da Consulta

Codigo	Name	Qtd^v	abrir	remover
1	Giacomo Guilizzoni Founder & CEO	40	a link	Remover
2	Marco Botton Tuttofare	38	a link	Remover
3	Mariah MacLachlan Better Half	41	a link	Remover
4	Valerie Liberty Head Chef	30	a link	Remover

Evento: EV-001.Buscar. Click

Entrada: Marca, Produto, Iniciais, Data cadastro, Data de Vencimento, Qtd em estoque ou Unidade de Medida;

Saída: Redireciona para [Dados de Consulta]

003.Estoque.Consultar para analise de Estoque

A Web Page

← → ✕ 🏠 🔍

Gerente Consultar

Marca específica

Qtd em estoque: ▲ ▼

estoque acaba em: 📅

Recompor estoque 📅

Agendar recomposição de estoque.

Alert

Agendamento de reposição
de material ao estoque feita

[fechar](#)

Evento: EV-001.AgendarReposiçãodeEstoque. Click

Entrada: Marca, Qtd;

Saída: Alert[[Agendamento de reposição de material ao estoque feita]]

004.Estoque.Remove(acabou)

A Web Page

Remover (Funcionario)

Qtde de produto: 0

Filtrar

Resultado da Filtragem:

ComboBox

Produto 1
Produto 2 -
Produto 10

Deletar do estoque

Filtrar

Marca:

Produto:

Data cadastrado: / /

Data vencimento: / /

Média de valor: A

Evento: EV-001.Filtrar. Click

Entrada: Marca, Produto, Data cadastro, Data vencimento, média de valor;

Saída: Redireciona para *[[consultar estoque]]*

Evento: EV-002.Deletar do Estoque.Click;

Entrada: Qtd, Escolher – Resultado de Filtragem;

005.Estoque.Remover(danificada)

A Web Page

← → × 🏠 🔍

Gerente (Remove)

ID produto danificado:

Motivo da remoção:

Quantidade:

Alert

Material removido do sistema

[fechar](#)

Alert

Erro ao remover item do sistema

[Tentar Novamente](#)

[fechar](#)

Evento: EV-001.Efetuar remoção. Click

Entrada: Código do produto, Motivo de remoção, quantidade;

Saída: *Aleta de confirmação*

Evento: EV-002.fecha.Click;

Saída: Reabrir pagina de remoção do estoque[[005.Estoque.Remover(danificada)]]

001.Cliente.Cadastrar Cliente

Cadastro de cliente

← → ✕ 🏠 🔍

Nome:

CPF:

Telefone ou Celular:

E-mail:

Validar cadastro

Coloque o código recebido para validar o cadastro:

Evento: EV-001CadastraCliente.Click

Entrada: nome, cpf, telefone, e-mail

Saída: *Aleta de validação;*

Evento: EV-002.ConcluirCadastro.Click;

Entrada: Código;

Saída: Redirecionar para pagina[[002.Cliente.ConsultadeDadosdoCliente]]

PROTP-002.Cliente.ConsultarDadosDoCliente

The image displays two sequential screenshots of a web browser interface. The top screenshot shows the initial state of the 'Consulta de dados do cliente' page. It features a title bar with navigation icons and a search bar. The main content area contains a form with two input fields: 'Nome completo:' and 'CPF:'. Below these fields is a green button labeled 'Pesquisar cliente'. The bottom screenshot shows the same browser window after the search button is clicked. A modal window titled 'Informações do cliente' is displayed, containing a list of client details: 'Número de celular:', 'E-mail:', 'Nome completo:', 'Alterar E-mail:', 'Alterar número de celular:', and 'Horario Marcado:'. Each detail is followed by a series of asterisks representing redacted information. A red 'X' icon is visible in the top right corner of the modal window.

Evento: EV-001.PesquisarCliente.Click

Entrada: Nome e CPF do clientes;

Saída: *Pop-upc com dados do cliente*

Evento: EV-002.fechar.Click;

Saída: Redirecionar para pagina[[002.Cliente.ConsultadeDadosdoCliente]]

003.Cliente.ConsultarDadosDoCliente

Marcar Horário

← → ✕ 🏠 🔍

Marcar horario de atendimento

Nome do cliente:

CPF:

Horario de atendimento:

JUNE 2020						
S	M	T	W	T	F	S
31	1	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
28	29	30	1	2	3	4

[Agendar](#)

Alert

Agendamento realizado com sucesso

[fechar](#)

Alert

Erro no agendamento, consulte se o cliente está correto

[Tentar novamente](#) [fechar](#)

Evento: EV-001.MarcarHorario.Click

Entrada: Nome, CPF e Horário Marcado;

Saída: Pop-up com alerta de confirmação;

Evento: EV-002.fechar.Click;

Saída: Recarregar pagina[[003.Cliente.ConsultarDadosDoCliente]]

Evento: EV-003.TentarNovamente.Click;

Saída: Recarregar pagina[[003.Cliente.ConsultarDadosDoCliente]]

003.Cliente.ConsultaHorariodosClientes

Consultar Horarios Marcados

← → × 🏠 https:// 🔍

Consultar horarios Marcados;

Nome do cliente:

CPF:

Consultar Horarios Marcados

← → × 🏠 https:// 🔍

×

JUNE 2020

S	M	T	W	T	F	S
31	1	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
28	29	30	1	2	3	4
5	6	7	8	9	10	11

Dia 8 às 14h20

001.Pagamento.Pagar

pagamento do cliente

https://

nome do cliente

CPF do cliente:

valor total:

serviços prestados

forma de pagamento:

☐ débito ☐ crédito ☐ dinheiro

pagamento concluído

Alert

Pagamento Realizado com sucesso foi enviado um e-mail para que o cliente possa avaliar nosso atendimento comunique a ele essa informação.

[fechar](#)

Alert

Falha no pagamento deseja tentar novamente?

[No](#) [Yes](#)

Evento: EV-001.Pagar.Click

Entrada: Nome e CPF do cliente, serviços prestados, forma de pagamento;

Saída: *Pop-up com alerta de confirmação do pagamento;*

Evento: EV-002.Yes.Click;

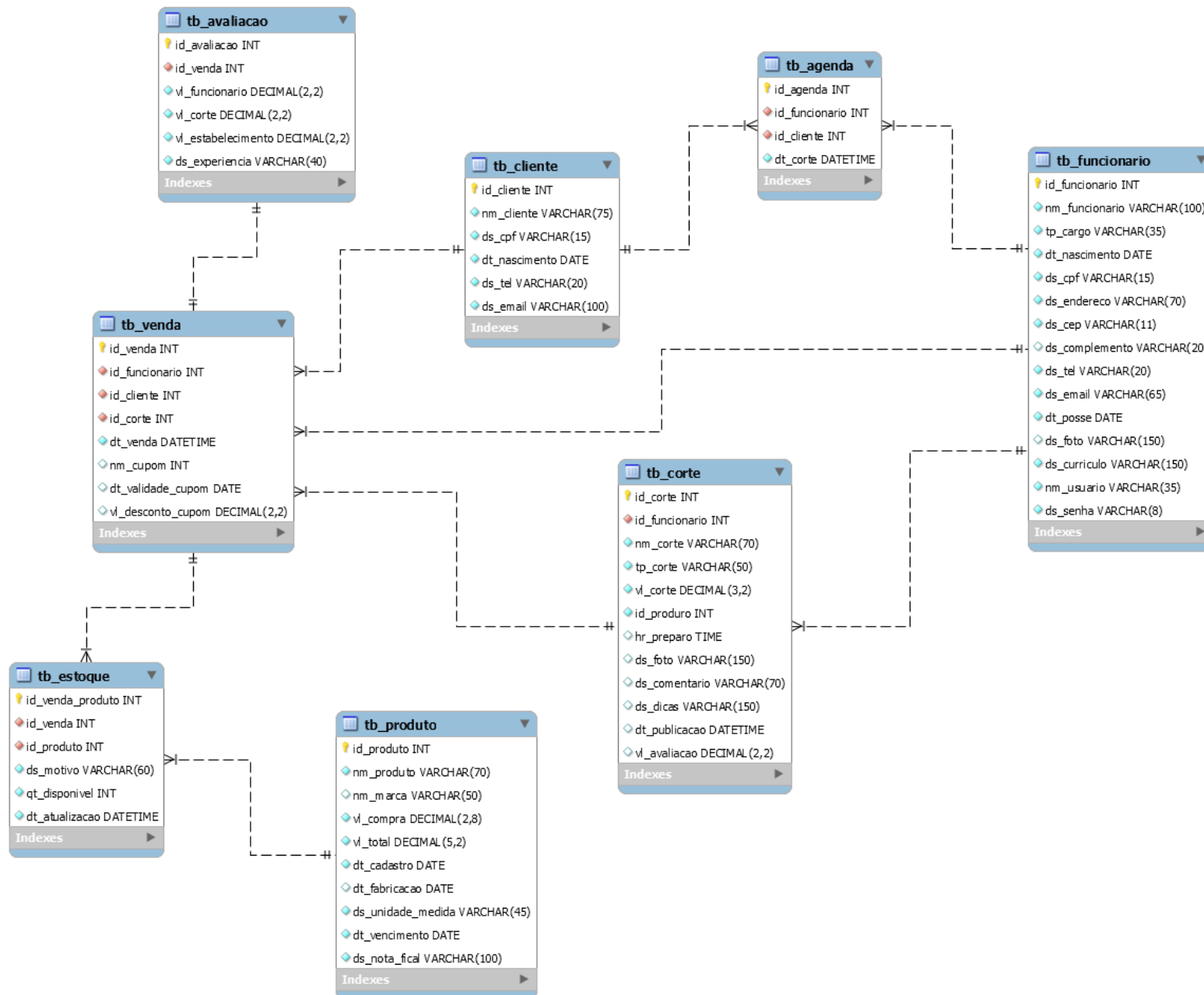
Saída: Recarrega a página *[[001.Pagamento.Pagar]]*

Evento: EV-002.No.Click;

Saída: Redirecionar para a página *[[003.Cliente.ConsultaHorariodosClientes]]*

MER

Modelagem



Script

```
create database bd_cabelereiro;
```

```
use bd_cabelereiro;
```

```
create table tb_cliente(
```

```
    id_cliente                int primary key auto_increment,
    nm_cliente                varchar(75) not null,
    ds_cpf                    varchar(15) not null,
    ds_cel_ou_tel             varchar(20) not null,
    ds_email                  varchar(100) not null
);
```

```
create table tb_funcionario(
```

```
    id_funcionario            int primary key auto_increment,
    nm_funcionario             varchar(100) not null,
    ds_cargo                   varchar(35) not null,
    dt_nascimento              date not null,
    ds_cpf                     varchar(15) not null,
    ds_endereco                varchar(70) not null,
    ds_cep                     varchar(11) not null,
    ds_complemento             varchar(20),
    ds_tel_ou_cel              varchar(20),
    ds_email                   varchar(65) not null,
    dt_posse                    date not null,
    dt_ult_feria               date,
    ds_foto                    varchar(150),
    ds_curriculo                varchar(150) not null,
    nm_usuario                  varchar(35) not null,
    ds_senha                   varchar(8) not null
);
```



```
create table tb_agenda(  
    id_agenda int primary key auto_increment,  
    id_funcionario int,  
    id_cliente int,  
    dt_agenda datetime,  
    foreign key (id_funcionario) references tb_funcionario (id_funcionario),  
    foreign key (id_cliente) references tb_cliente (id_cliente)  
);
```

```
create table tb_produto(  
    id_produto int primary key auto_increment,  
    nm_produto varchar(70),  
    nm_marca varchar(50),  
    nr_codigo int,  
    vl_total decimal(5, 2),  
    dt_cadastro datetime,  
    dt_fabricacao date,  
    dt_vencimento date,  
    ds_nota_fiscal varchar(100)  
);
```

```
create table tb_estoque(  
    id_estoque int primary key auto_increment,  
    id_produto int,  
    ds_motivo varchar(50),  
    qt_disponivel int,  
    dt_atualizacao datetime,  
    foreign key (id_produto) references tb_produto (id_produto)  
);
```

```

CREATE TABLE tb_corte (
    `id_corte` INT primary key AUTO_INCREMENT,
    `id_funcionario` INT NOT NULL,
    `nm_corte` VARCHAR(70) NOT NULL,
    `tp_corte` VARCHAR(50) NOT NULL,
    `vl_corte` DECIMAL(3,2) NOT NULL,
    `id_produto` INT NOT NULL,
    `hr_preparo` TIME NULL,
    `ds_foto` VARCHAR(150) NULL,
    `ds_comentario` VARCHAR(70) NULL,
    `ds_dicas` VARCHAR(150) NULL,
    `dt_publicacao` DATETIME NULL,
    `vl_avaliacao` DECIMAL(2,2) NULL,
    foreign key (`id_funcionario`) references `tb_funcionario` ( `id_funcionario`)
);

```

```

create table tb_venda(
    id_venda                                int primary key auto_increment,
    id_funcionario                          int,
    id_cliente                             int,
    id_corte                                int,
    dt_venda                               datetime,
    nm_cupom                                int,
    dt_validade_cupom                       date,
    vl_desconto_cupom                       decimal(2, 2),
    foreign key (id_funcionario) references tb_funcionario (id_funcionario),
    foreign key (id_cliente) references tb_cliente (id_cliente),
    foreign key (id_corte) references tb_corte (id_corte)
);

```

```

create table tb_venda_produto(
    id_venda_produto          int primary key auto_increment,
    id_venda                  int,
    id_produto                int,
    foreign key (id_venda) references tb_venda (id_venda),
    foreign key (id_produto) references tb_produto (id_produto)
);

```

```

create table tb_avaliacao_atendimento(
    id_avaliacao              int primary key auto_increment,
    id_cliente                int,
    id_funcionario            int,
    id_corte                  int,
    vl_funcionario            decimal(2, 2),
    vl_corte                  decimal(2, 2),
    vl_estabelecimento        decimal(2, 2),
    ds_experiencia            varchar(255),
    foreign key (id_cliente) references tb_cliente (id_cliente),
    foreign key (id_funcionario) references tb_funcionario (id_funcionario),
    foreign key (id_corte) references tb_corte (id_corte)
);

```

```

/*002.Corte*/
select *
    from tb_corte                c
inner join tb_funcionario        f
    on f.id_funcionario = c.id_funcionario
where nm_corte like "
    and hr_preparo = '20:10'
    and dt_publicacao = '2020-05-05'
    and nm_funcionario like "

```

and tp_corte like ";

/*PROT-002*/

```
select nr_codigo, nm_produto, qt_disponivel
from tb_estoque                e
inner join tb_produto          p
on p.id_produto = e.id_produto;
```

/*PROTP-002.Cliente*/

```
select *
from tb_cliente
where ds_cpf = "
and nm_cliente like ";
```

/*003.Cliente.ConsultaHorariodosClientes*/

```
select dt_agenda
from tb_agenda                a
inner join tb_cliente          c
on c.id_cliente = a.id_cliente
where nm_cliente like " and ds_cpf = ";
```