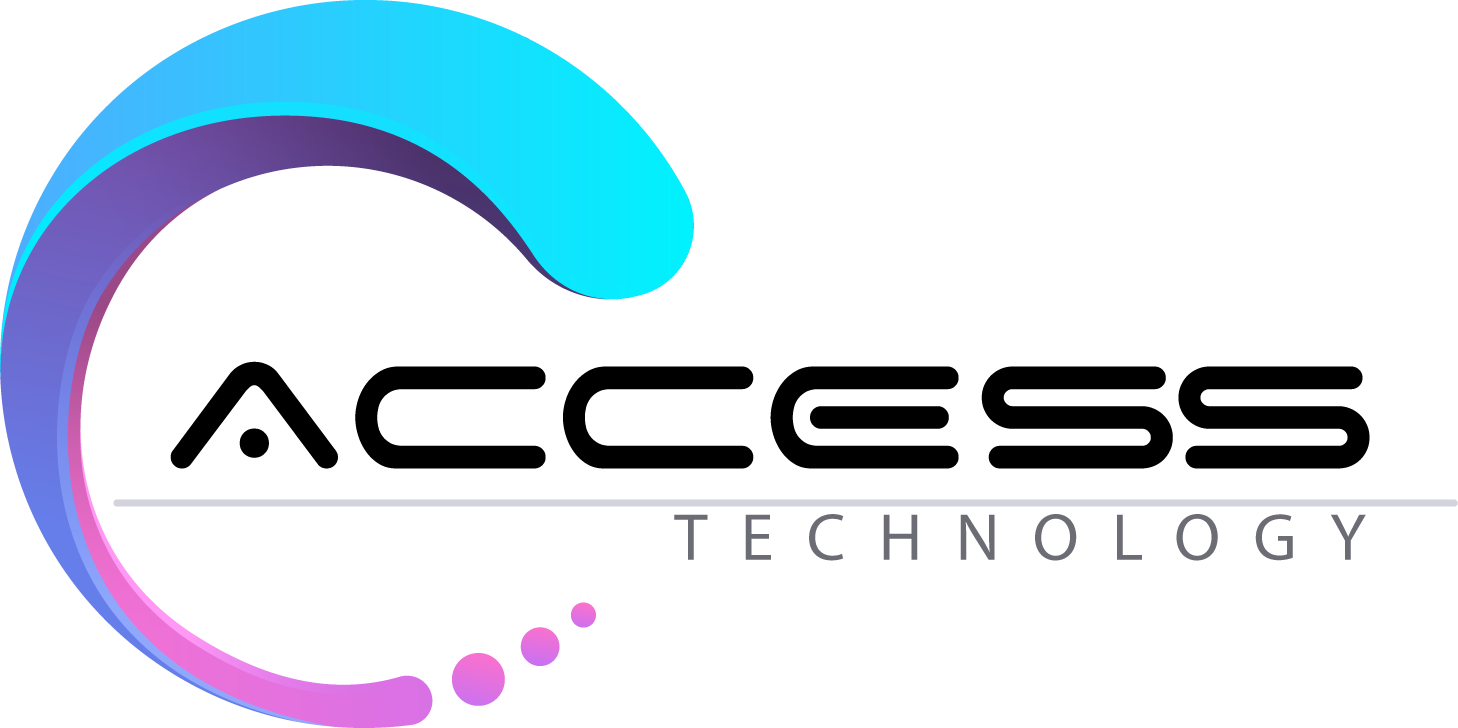
****

**B2. Battle one: First TCC Group Mission**

**São Paulo**

**2020**

# Integrantes

## INFO D

## Pedro Oliveira Sousa Nº 38

Sumário

[Integrantes 2](#_Toc46170914)

[Módulos 4](#_Toc46170921)

[Especificações 5](#_Toc46170922)

[USE CASE FUNCIONARIO 5](#_Toc46170923)

[UC-Acesso.001: 5](#_Toc46170924)

[UC-Acesso.002: 6](#_Toc46170925)

[UC-Acesso.003: 6](#_Toc46170926)

[UC-Acesso.004: 6](#_Toc46170927)

[USE CASE CORTE 6](#_Toc46170928)

[UC-Corte.001: 6](#_Toc46170929)

[UC-Corte.002: 7](#_Toc46170930)

[UC-Corte.003: 7](#_Toc46170931)

[USE CASE ESTOQUE 7](#_Toc46170932)

[UC-Estoque.001: 7](#_Toc46170933)

[UC-Estoque.002: 8](#_Toc46170934)

[UC-Consulta.003: 8](#_Toc46170935)

[UC-Remover(acabou).004: 8](#_Toc46170936)

[UC-Remover(danificado).005: 8](#_Toc46170937)

[USE CASE CLIENTE 8](#_Toc46170938)

[UC-Cliente.001: 8](#_Toc46170939)

[UC-Cliente.002: 9](#_Toc46170940)

[UC-Cliente.003: 9](#_Toc46170941)

[UC-Cliente.004: 9](#_Toc46170942)

[UC-Cliente.005: 9](#_Toc46170943)

[USE CASE PAGAMENTO 9](#_Toc46170944)

[UC-Pagamento.001: 9](#_Toc46170945)

[UC-Pagamento.002: 9](#_Toc46170946)

[UC-Pagamento.003: 10](#_Toc46170947)

[Protótipos 10](#_Toc46170948)

[PROTP-001.CadastrarFuncionario 10](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170949)

[PROTP-002.Login 11](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170950)

[PROTP-003.Esqueci a Senha 12](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170951)

[004.Alterar a senha 13](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170952)

[PROTP-005.Area de Funcionário 14](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170953)

[001.Cadastra Corte 16](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170954)

[002.Corte.Consulta de corte 17](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170955)

[003.Corte.Remover Corte 18](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170956)

[001.Estoque.Inserir no estoque 19](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170957)

[PROT-002.Estoque.Consultar Estoque 20](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170958)

[003.Estoque.Consultar para analise de Estoque 21](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170959)

[004.Estoque.Remover(acabou) 22](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170960)

[005.Estoque.Remover(danificada) 23](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170961)

[001.Cliente.Cadastrar Cliente 24](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170962)

[PROTP-002.Cliente.ConsultarDadosDoCliente 25](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170963)

[003.Cliente.ConsultarDadosDoCliente 26](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170964)

[003.Cliente.ConsultaHorariodosClientes 27](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170965)

[001.Pagamento.Pagar 28](file:///C:\Users\pedro\Documents\Projeto%20Cabelereiro\Product%20Owner.docx#_Toc46170966)

[MER 30](#_Toc46170967)

[**Modelagem** 31](#_Toc46170968)

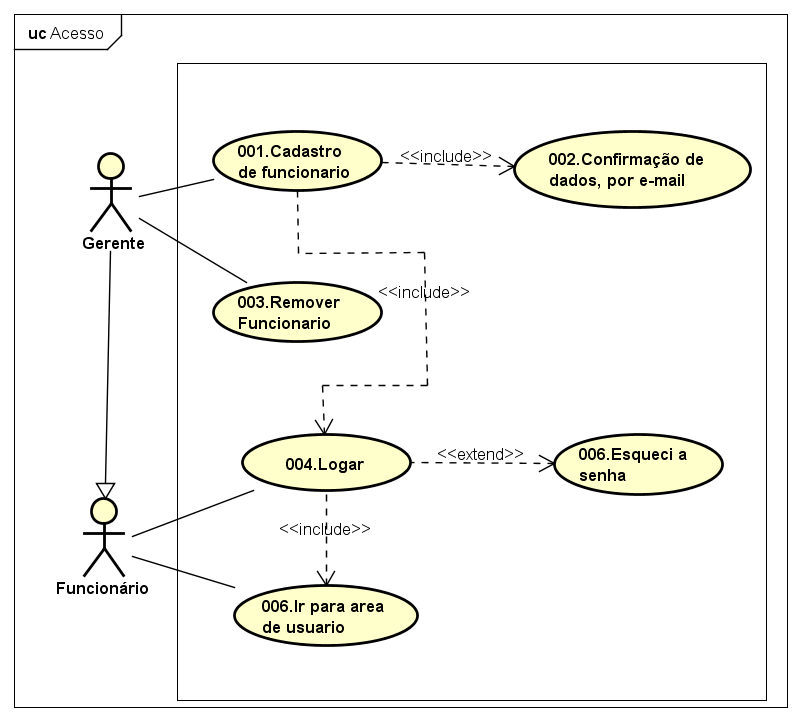
[**Script** 32](#_Toc46170969)

# Módulos

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

# Especificações

## USE CASE FUNCIONARIO



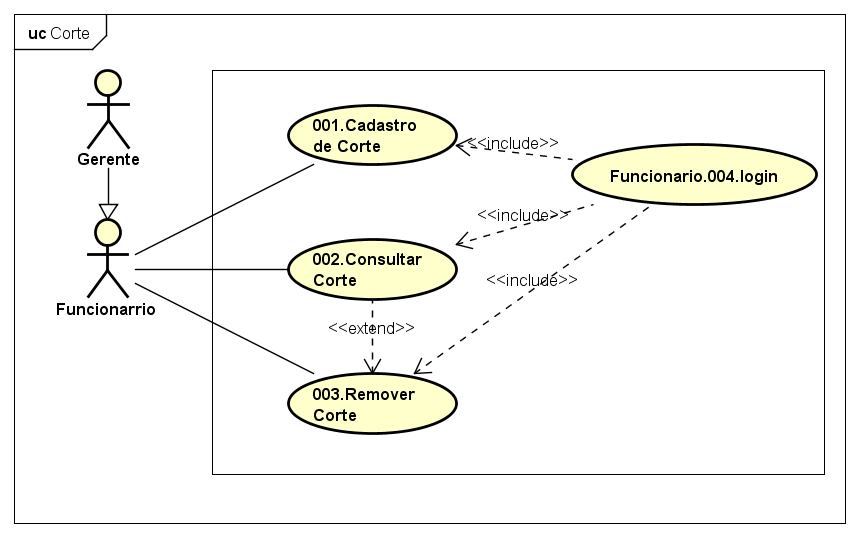
UC-Acesso.001: Os dados necessários para o cadastro são Nome completo, Número dos documentos RG ou CPF, foto 3x4, Currículo, número de telefone, endereço de e-mail, endereço residencial, cep, data de nascimento, pdf com o contrato assinado pelo gerente e contratado. Se não haver registro de usuários 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 valido 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 sida de cada funcionário, estoque;

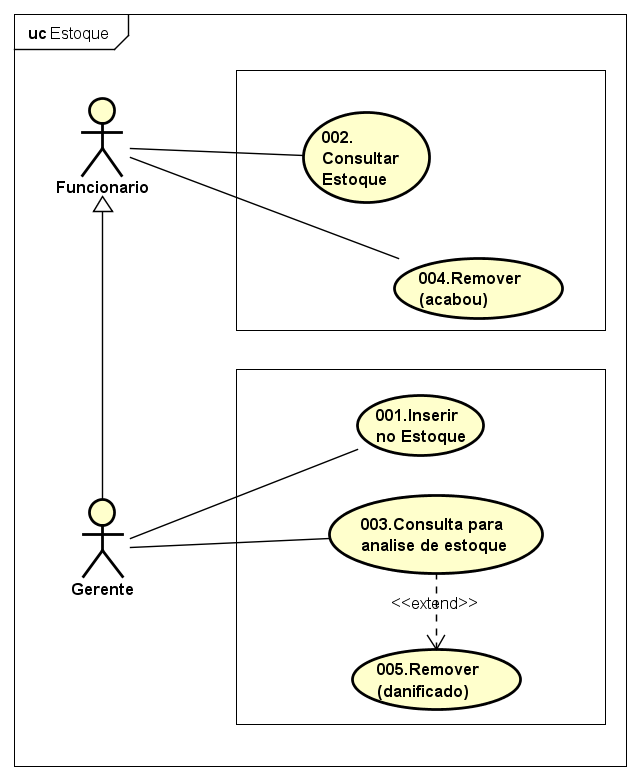
## USE CASE CORTE

****

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, avalição 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.

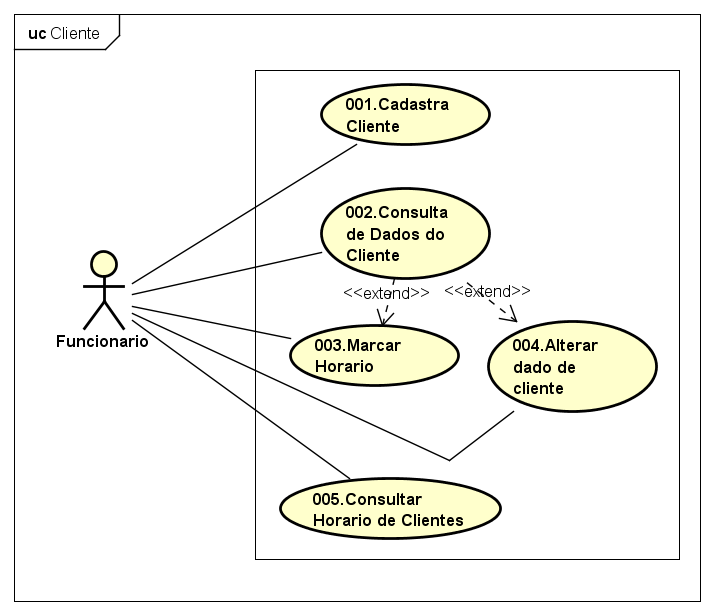
UC-Estoque.002: Nesta área qualquer funcionário tem acesso onde será mostrado a os dados cadastrados e a quantidade sempre atualizada de acordo com as remoções de itens do sistema. Poderá realizar filtros para facilitar a identificação de um produto. Em caso de não identificar os campos inseridos no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

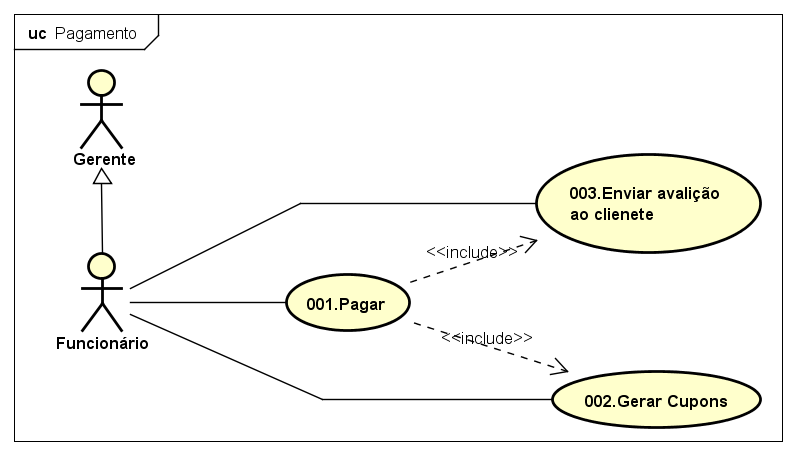
UC-Cliente.002: Consulta e feita pelo nome e cpf e então é mostrado as seguintes informações: horários marcados, nome, número de telefone, e-mail. A partir da consulta deve-se poder marcar horário para cliente e alterara dados do cliente.

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 avalição)

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

**Evento:** EV-001.CadastrodeFuncionario.Click

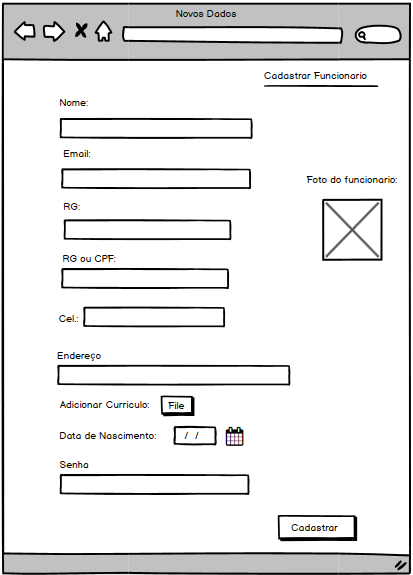
**Entrada:** Nome, E-mail, RG ou CPF, Cel, Endereço, Adicionar Currículo, Data de nascimento, Senha;

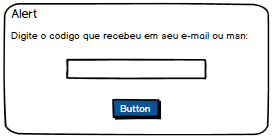
**Saída:** p*op-up (Alert de verificação)*

**Evento:** EV-001.CadastrodeFuncionario.Click

**Entrada:** Código de validação;

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





## PROTP-002.Login

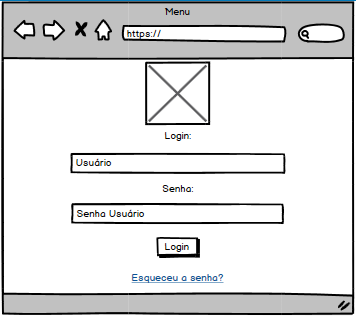
**Evento:** EV-001.Login.Click

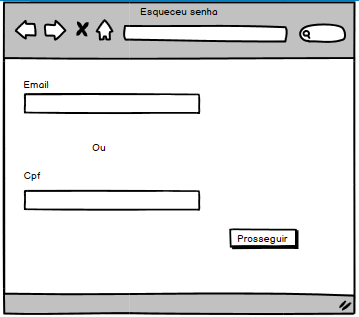
**Entrada:** e-mail e senha

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

**Evento:** EV-001.EsqueceuaSenha.Click

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





## PROTP-003.Esqueci a Senha

**Evento:** EV-001.Prosseguir.Click

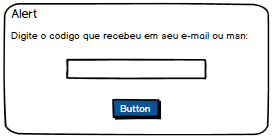
**Entrada:** e-mail ou cpf

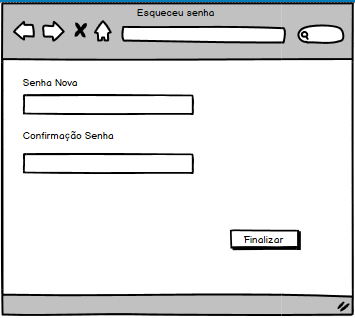
**Saída:** *pop-up (Alert de verificação)*

**Evento:** EV-002.CadastrodeFuncionario.Click

**Entrada:** Código de validação;

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





## 004.Alterar a senha

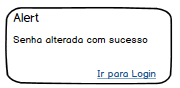
**Evento:** EV-001.Finalizar.Click

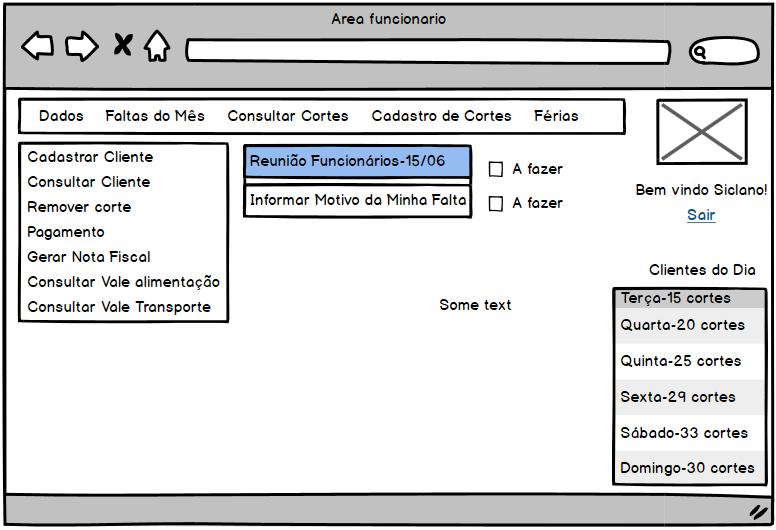
**Entrada:** nova senha, comfirmação da senha(nova)

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

**Evento:** EV-002.Ir para login.Click

**Saída:** *Redireciona para [[Login]];*





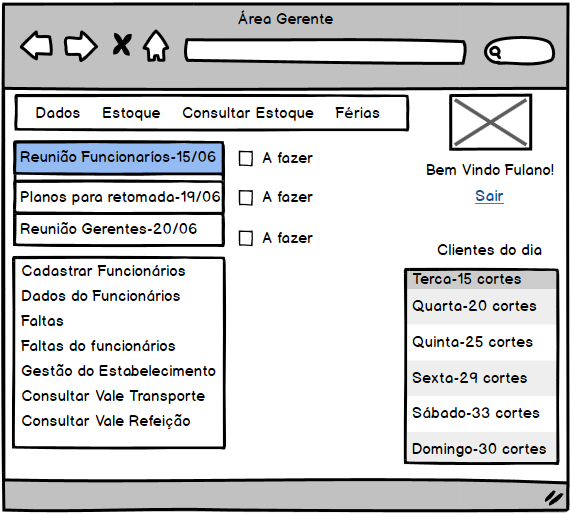
## PROTP-005.Area de Funcionário

**Evento:** EV-001.Sair.Click

**Entrada:** nova senha, comfirmação da senha(nova)

**Saída:** *Redirecionado para [[002.Login]]*

**4**



## 001.Cadastra Corte

**Evento:** EV-001.CadastroCorte. Click

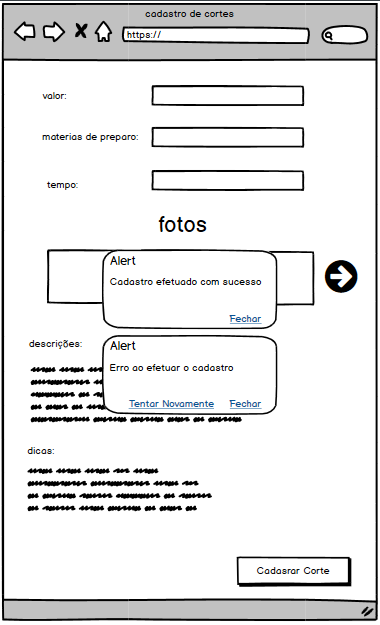
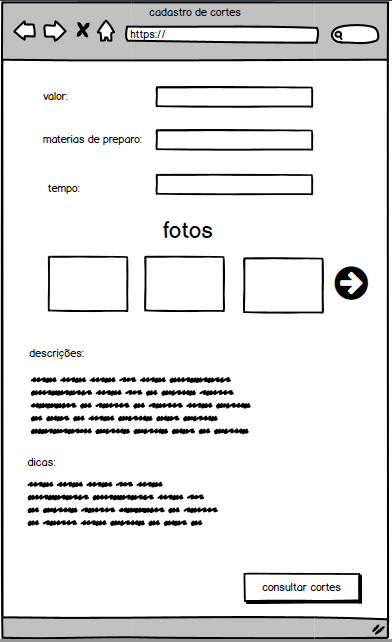
**Entrada:** Valor, Material de preparo, tempo, imagens, descrição, dicas;

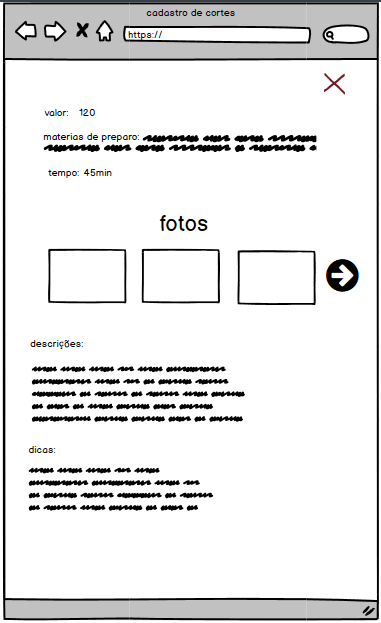
**Saída:** Fechar - Redirecionamento para *[[área de Consulta de Cortes]]*;

**Evento:** EV-002.CadastroCorte. Click

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

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





## 002.Corte.Consulta de corte

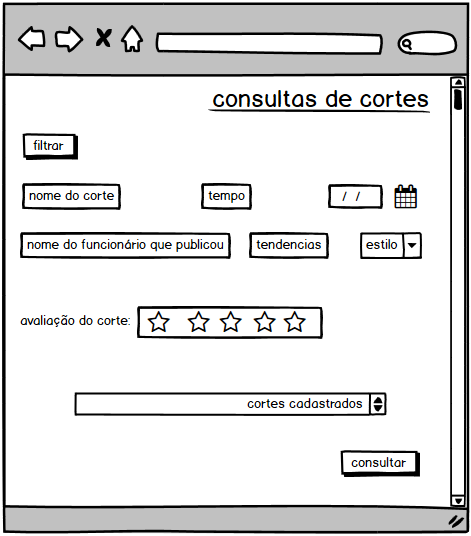
**Evento:** EV-001.Consultar Cortes. Click

**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

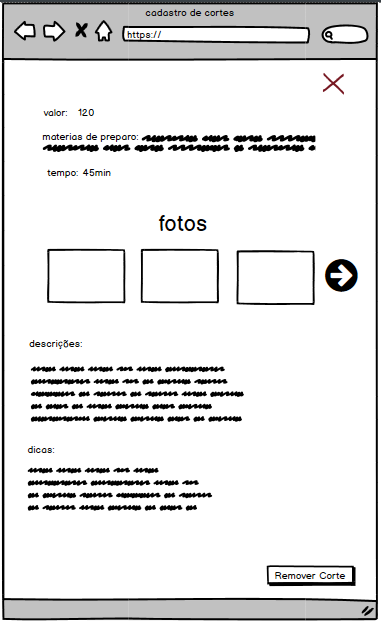
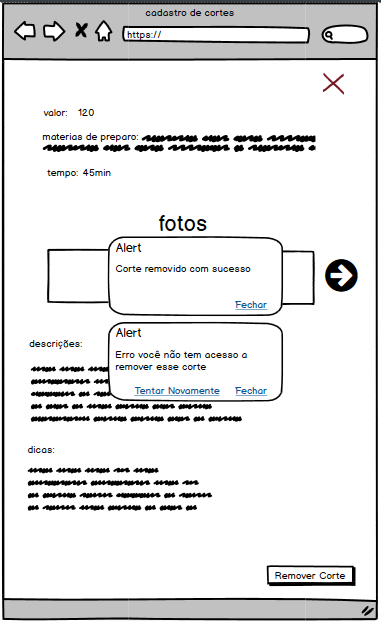
**Saída:** Redirecionamento para *[[retorna para tela de consulta de corte]]*



## 003.Corte.Remover Corte

**Evento:** EV-001.RemoverCorte. Click

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

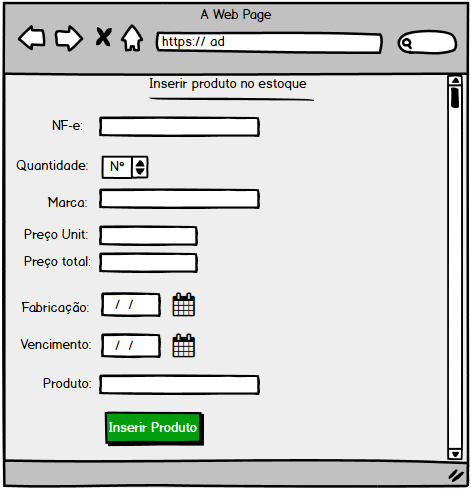


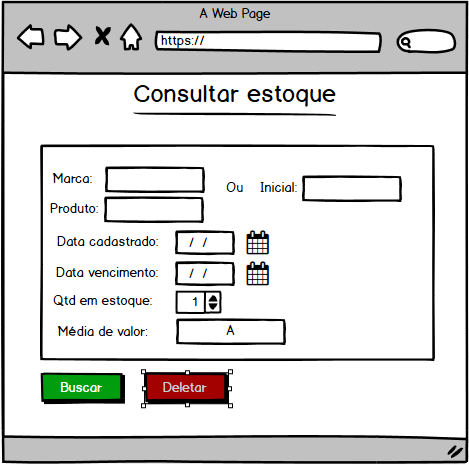
## 001.Estoque.Inserir no estoque

**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

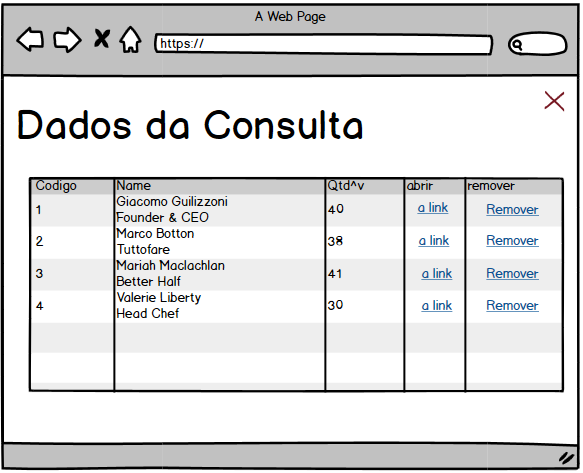
**Evento:** EV-001.Buscar. Click

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

**Saída:** *Redireciona para [Dados de Consulta]]*

**Evento:** EV-002.Deletar. Click

**Saída:** *Redireciona para [Remover Produto]]*



## 003.Estoque.Consultar para analise de Estoque

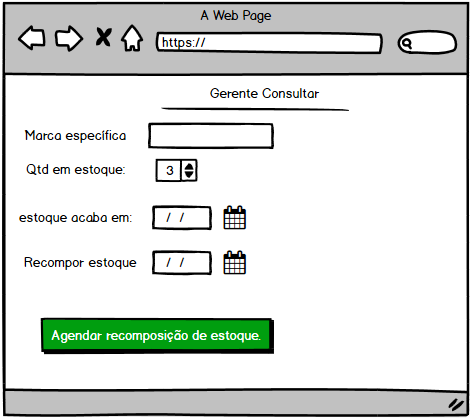
**Evento:** EV-001.AgendarReposiçãodeEstoque. Click

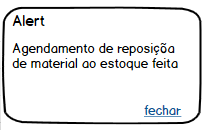
**Entrada:** Marca, Qtd;

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

**Evento:** EV-001.Fechar. Click

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





## 004.Estoque.Remover(acabou)

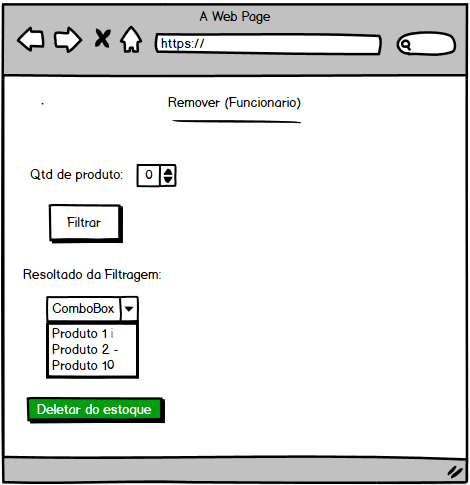
**Evento:** EV-001.Filtrar. Click

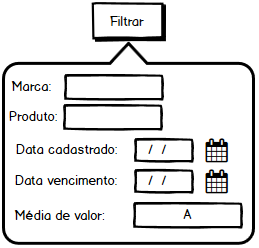
**Entrada:** Marca, Produto, Data cadastro, Data vencimento, média de valor;

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

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

**Entrada:** Qtd, Escolher – Resultado de Filtragem;





## 005.Estoque.Remover(danificada)

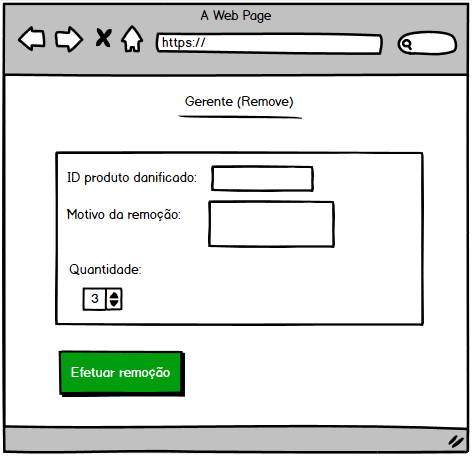
**Evento:** EV-001.Efetuar remoão. Click

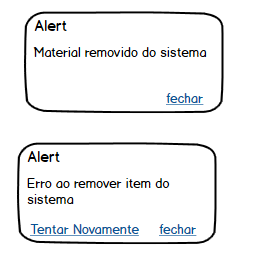
**Entrada:** Código do produto, Motivo de remoção, quantidade;

**Saída:** *Aleta de confirmação*

**Evento:** EV-002.fecha.Click;

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





## 001.Cliente.Cadastrar Cliente

**Evento:** EV-001CadastraCliente.Click

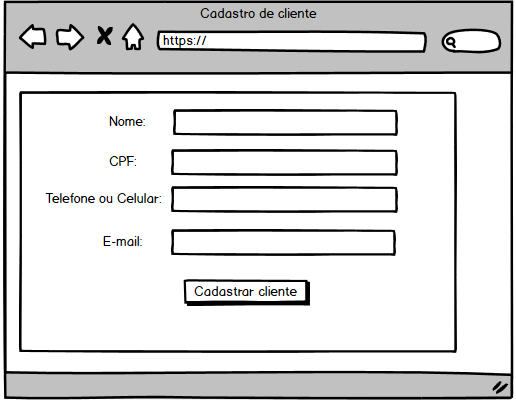
**Entrada:** nome, cpf, telefone, e-mail

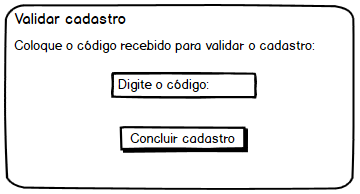
**Saída:** *Aleta de validação;*

**Evento:** EV-002.ConcluirCadastro.Click;

**Entrada:** Código;

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





## PROTP-002.Cliente.ConsultarDadosDoCliente

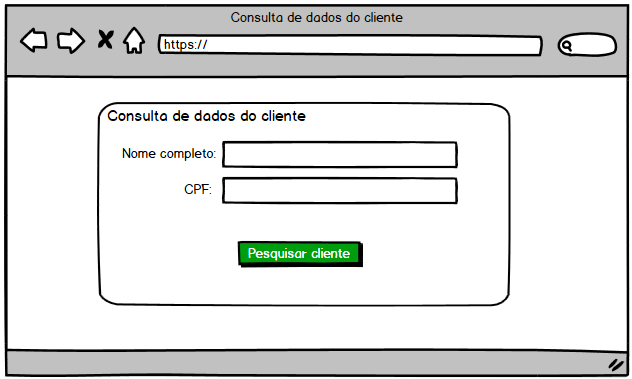
**Evento:** EV-001.PesquisarCliente.Click

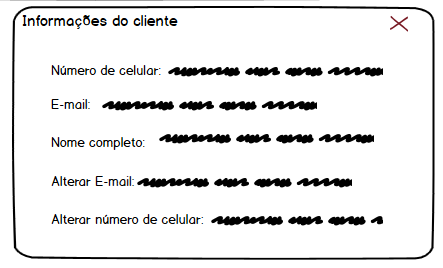
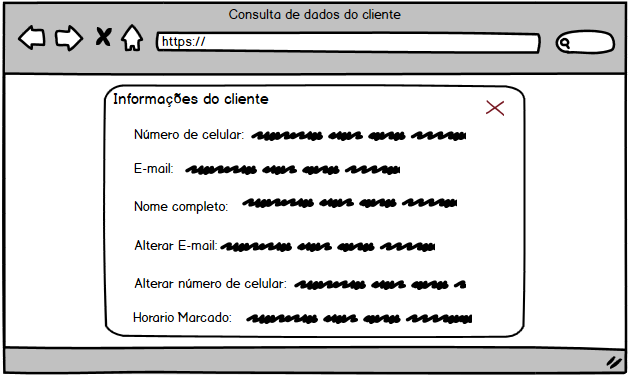
**Entrada:** Nome e CPF do clientes;

**Saída:** *Pop-upc com dados do cliente*

**Evento:** EV-002.fechar.Click;

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





## 003.Cliente.ConsultarDadosDoCliente

**Evento:** EV-001.MarcarHorario.Click

**Entrada:** Nome, CPF e Horário Marcado;

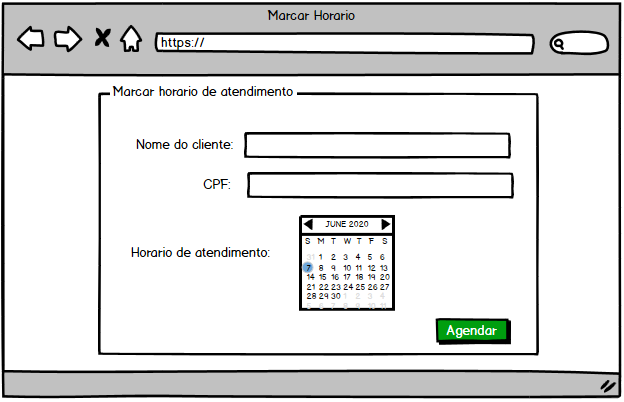
**Saída:** *Pop-upc com alerta de confirmação;*

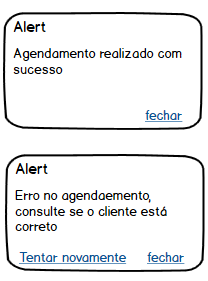
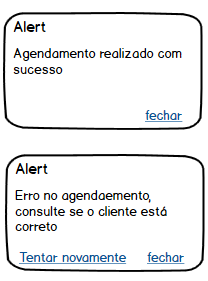
**Evento:** EV-002.fechar.Click;

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

**Evento:** EV-003.TentarNovamente.Click;

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





## 003.Cliente.ConsultaHorariodosClientes

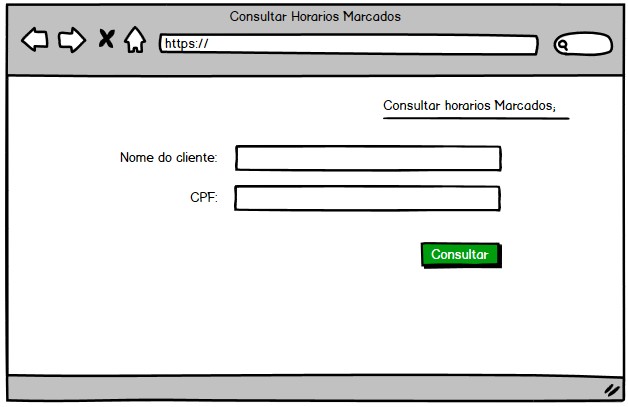
**Evento:** EV-001.MarcarHorario.Click

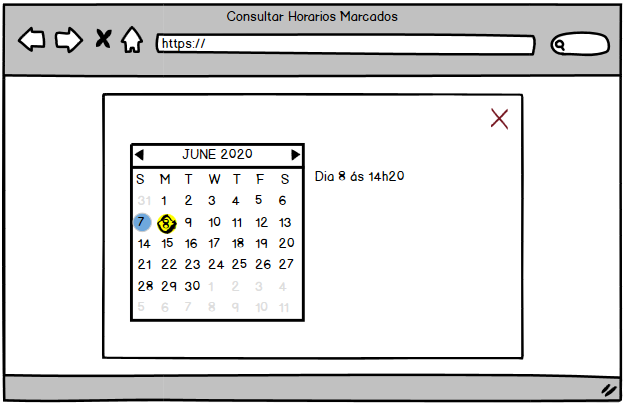
**Entrada:** Nome e CPF;

**Saída:** *Pop-upc com alerta horários que o cliente tem marcados;*

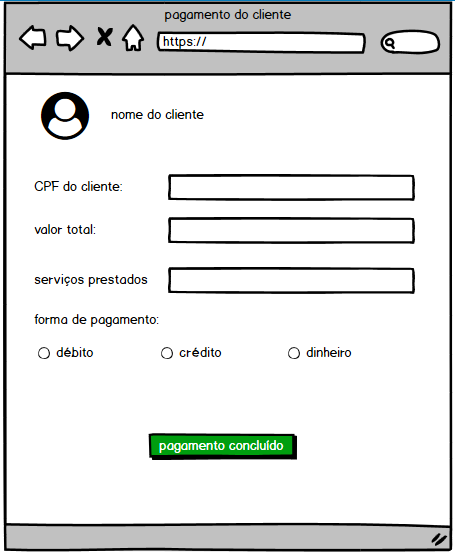
**Evento:** EV-002.fechar.Click;

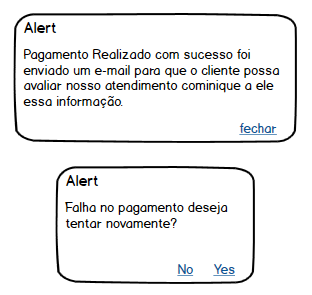
**Saída:** Recarregar pagina*[[003.Cliente.ConsultaHorariodosClientes]]*





## 001.Pagamento.Pagar





**Evento:** EV-001.Pagar.Click

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

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

**Evento:** EV-002.Yes.Click;

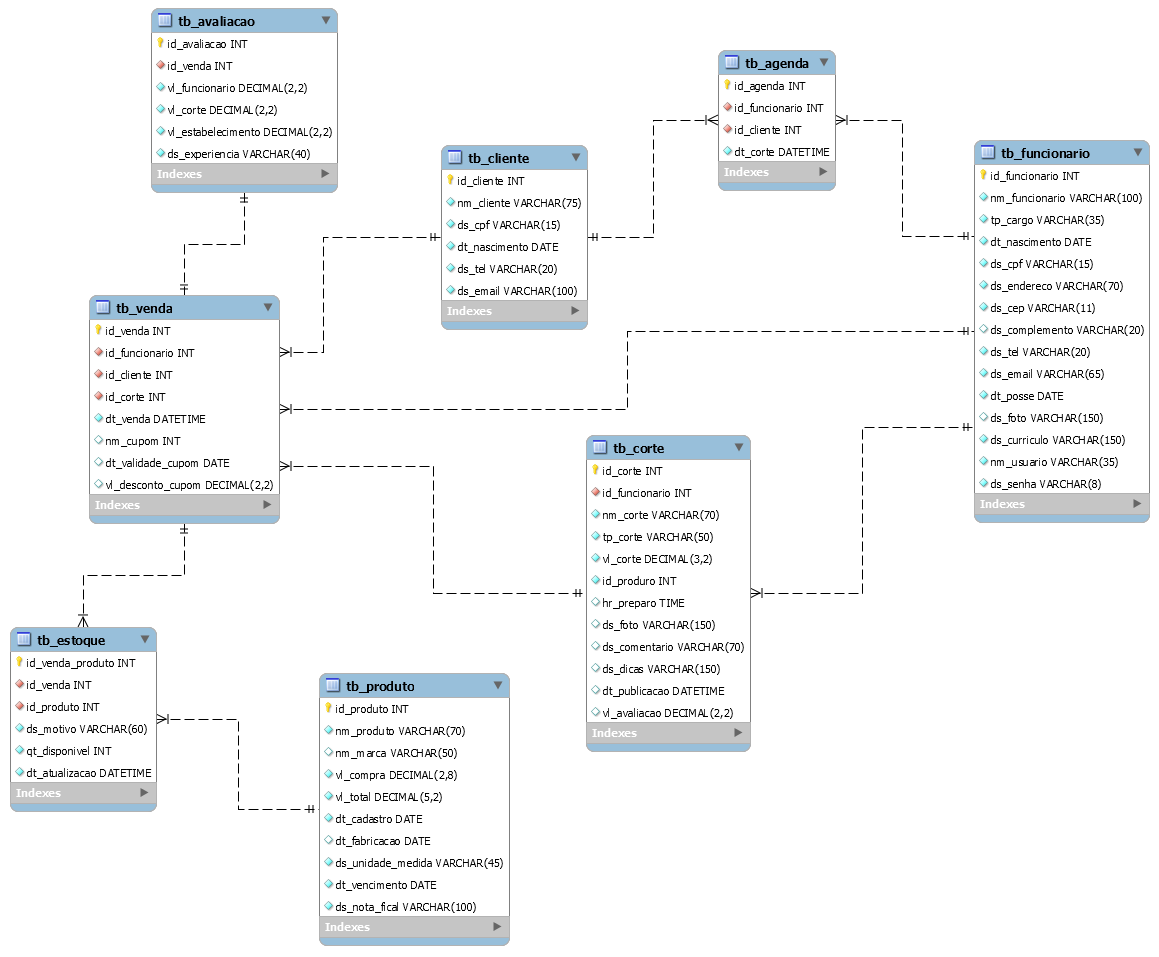
**Saída:** Recarrega a página *[[001.Pagamento.Pagar]]*

**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\_produro` 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\_avalicao 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 = '';