

# Restaurante do Zé

Disciplina: Prática Integrada 2

Professores: Geomar Schreiner e Jean Carlos  
Henrichs

Acadêmico(s): Bruno Figueira , Matheus  
Giordani, Jonathan da Cruz

# Introdução



Neste trabalho realizamos o ciclo de desenvolvimento completo de um software para controle de múltiplos restaurantes.

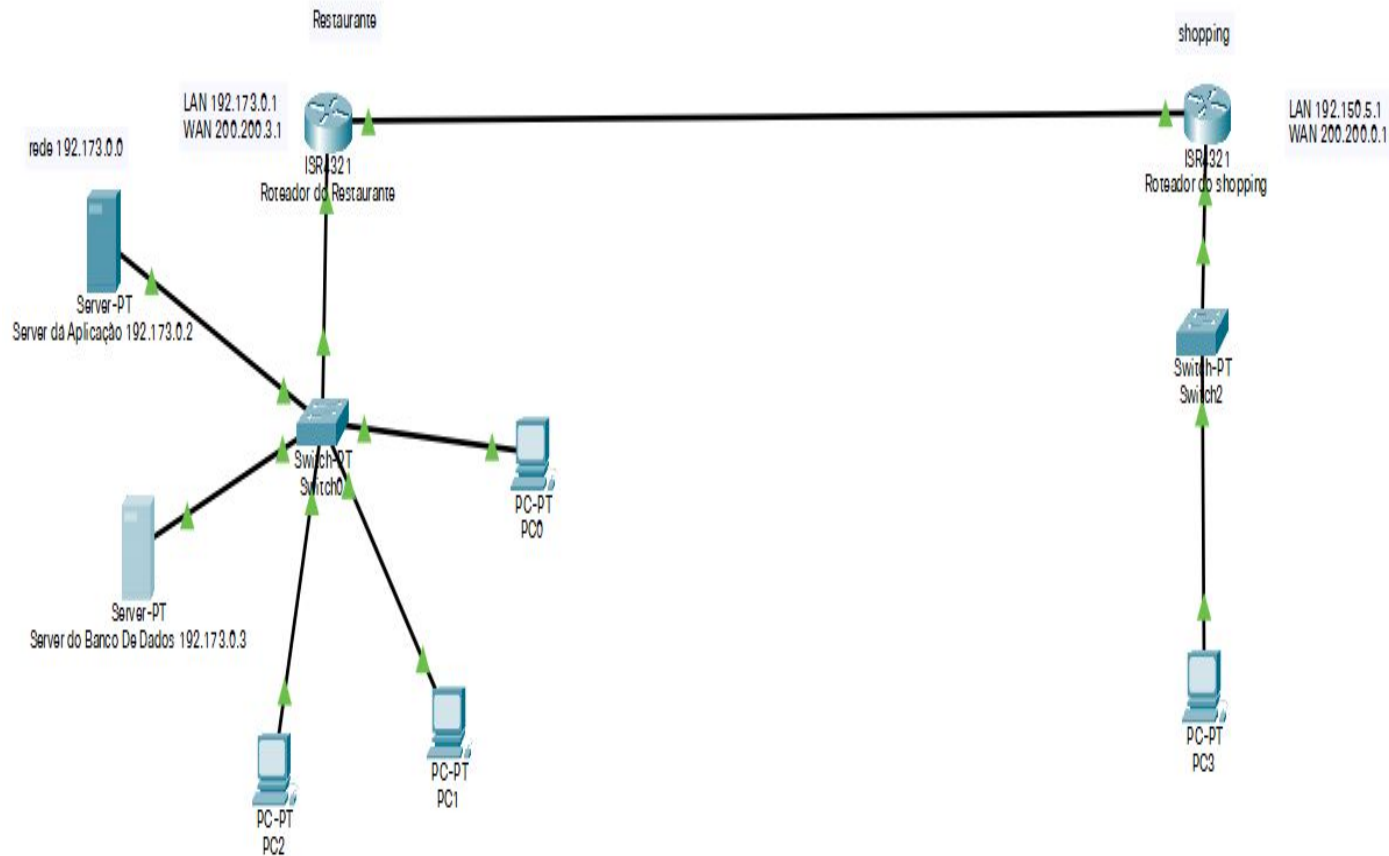


# REDES

Utilizado plataforma heroku para hospedagem das aplicações

- BD - ec2-18-215-96-54.compute-1.amazonaws.com
  - IP - 18.215.96.54
  - PORTA - 5432
- Backend <https://restaurantes-pi2.herokuapp.com/>
  - IP - 18.211.231.38
  - PORTA - 443
- Frontend <https://jonathandacruz.github.io/praticabkp/>

## Cisco Packet Tracer





## Configuração DHCP . DNS e HTTP

SERVICES

HTTP

DHCP

DHCPv6

TFTP

DNS

SYSLOG

AAA

NTP

EMAIL

FTP

IoT

VM Management

Radius EAP

HTTP

HTTP

☒ On ☐ Off

HTTPS

☒ On ☐ Off

File Manager

	File Name	Edit	Delete
1	copyrights.html	(edit)	(delete)
2	cscoptlogo177x111.jpg		(delete)
3	helloworld.html	(edit)	(delete)
4	image.html	(edit)	(delete)
5	index.html	(edit)	(delete)



## Configuração DHCP . DNS e HTTP

SERVICES

HTTP

**DHCP**

DHCPv6

TFTP

DNS

SYSLOG

AAA

NTP

EMAIL

FTP

IoT

VM Management

Radius EAP

DHCP

Interface FastEthernet0 Service ☒ On ☐ Off

Pool Name serverPool

Default Gateway 192.173.0.1

DNS Server 192.173.0.2

Start IP Address : 192 173 0 11

Subnet Mask: 255 255 255 0

Maximum Number of Users : 244

TFTP Server: 0.0.0.0

WLC Address: 0.0.0.0

Add

Save

Remove

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
serverPool	192.173...	192.173...	192.173...	255.255...	244	0.0.0.0	0.0.0.0



## Configuração DHCP . DNS e HTTP

**SERVICES** ^

HTTP

DHCP

DHCPv6

TFTP

**DNS**

SYSLOG

AAA

NTP

EMAIL

FTP

IoT

VM Management

Radius EAP

DNS

DNS Service ☒ On ☐ Off

Resource Records

Name

Type **A Record** v

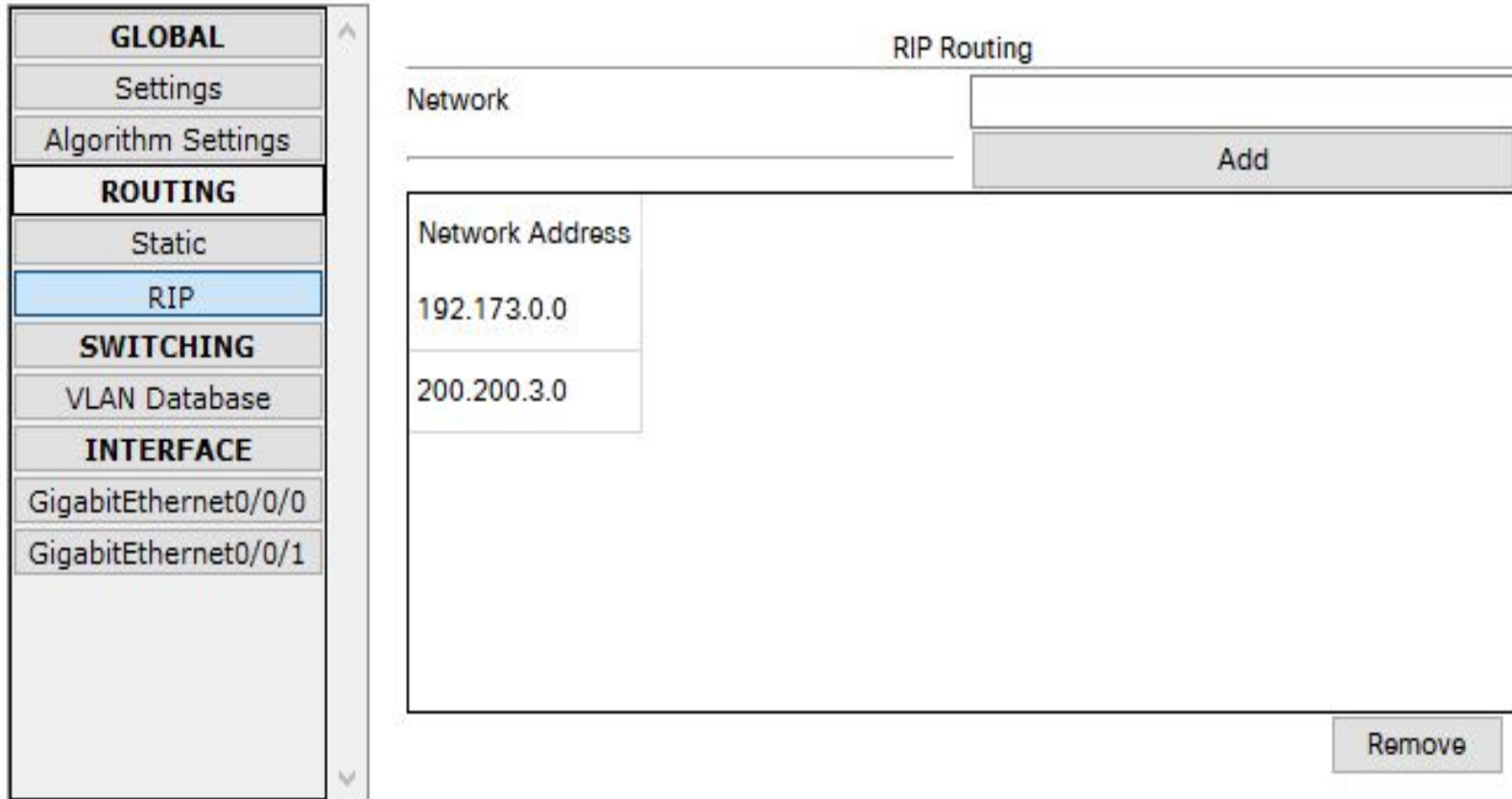
Address

Add Save Remove

No.	Name	Type	Detail
0	www.integrador.com.br	A Record	192.173.0.2



## Configuração Roteador



The screenshot displays a network configuration interface for a router. On the left is a vertical sidebar with a tree view of configuration categories: GLOBAL, Settings, Algorithm Settings, ROUTING, Static, SWITCHING, VLAN Database, INTERFACE, GigabitEthernet0/0/0, and GigabitEthernet0/0/1. The 'ROUTING' category is expanded, and 'RIP' is selected. The main area is titled 'RIP Routing'. It features a 'Network' label above a text input field and an 'Add' button. Below this is a table with the header 'Network Address' containing two entries: '192.173.0.0' and '200.200.3.0'. A 'Remove' button is located at the bottom right of the table.

Network Address
192.173.0.0
200.200.3.0

Copyright 1998 Cisco Systems, Inc.

## Configuração Roteador

Physical

Config

CLI

Attributes

GLOBAL

Settings

Algorithm Settings

ROUTING

Static

RIP

SWITCHING

VLAN Database

INTERFACE

GigabitEthernet0/0/0

GigabitEthernet0/0/1

Static Routes

Network

Mask

Next Hop

Add

Network Address

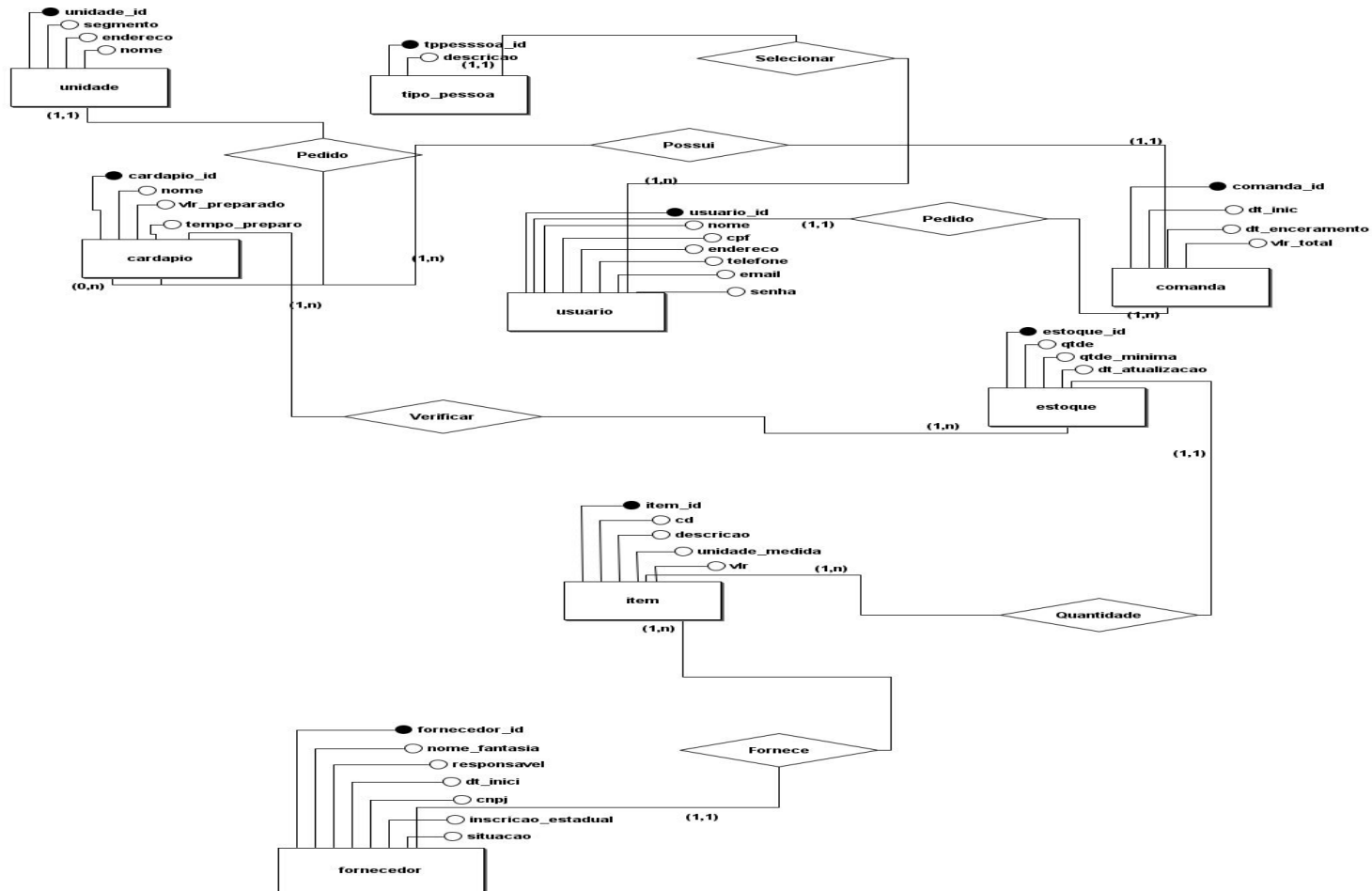
192.150.5.0/24 via 200.200.3.2

Remove

# BANCO DE DADOS

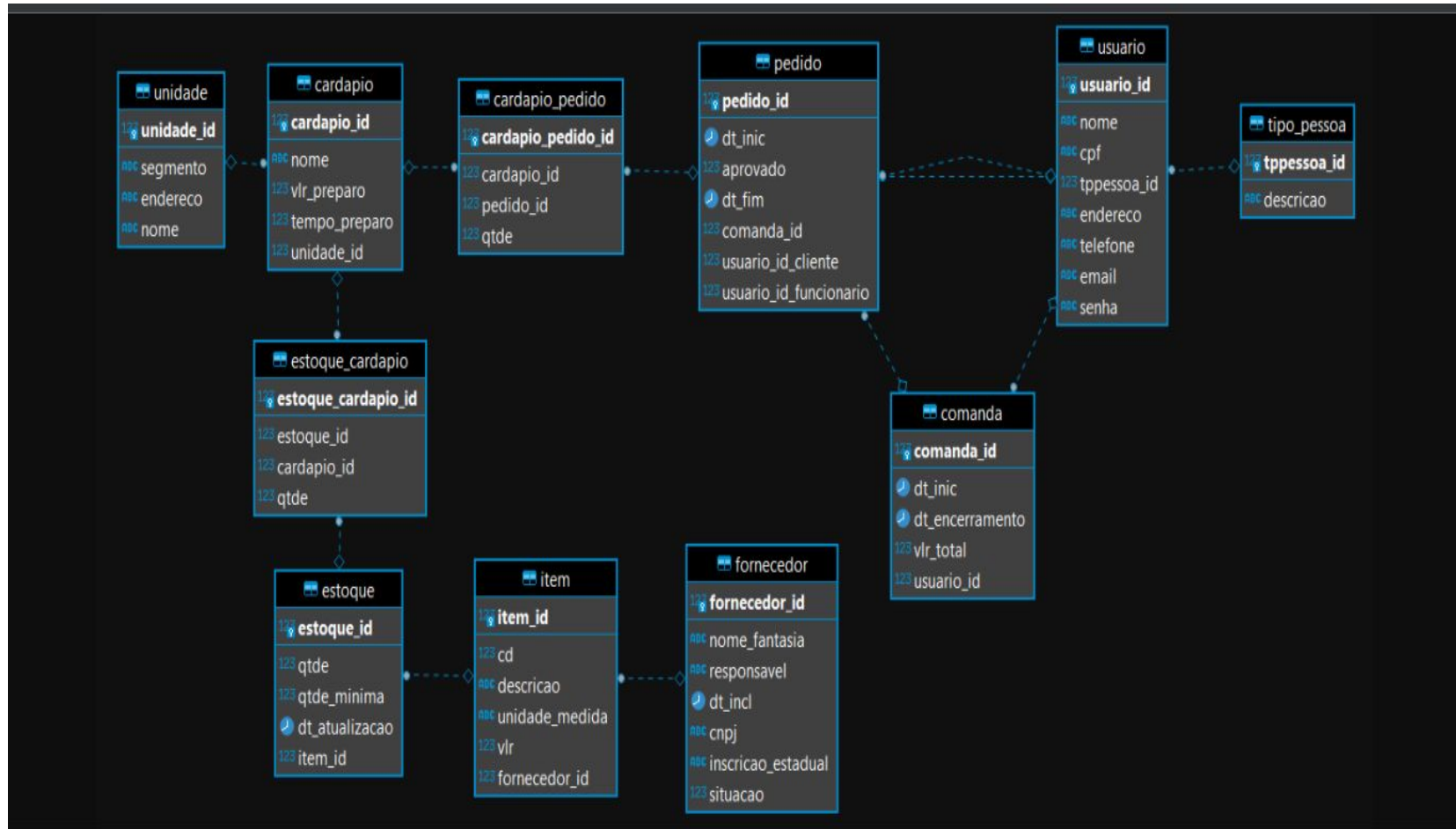
# Banco de Dados

## Modelo E.R



# Banco de Dados

## Modelo l3gico



# Banco de Dados

## Função fechar comanda

```
CREATE OR REPLACE FUNCTION fkg_fecha_comanda (en_usuario_id_clie int) RETURNS int AS $$
DECLARE
    vn_tot_pagar      float;
    vn_comanda_id     int;
    vn_return         int;
BEGIN
    SELECT sum( ( (i.vlr * ec.qtde) + c2.vlr_preparo) * cp.qtde )
        , c.comanda_id
    INTO vn_tot_pagar
        , vn_comanda_id
    FROM comanda c
    JOIN pedido p      ON (c.usuario_id = p.usuario_id_cliente AND c.comanda_id = p.comanda_id )
    JOIN cardapio_pedido cp ON p.pedido_id = cp.pedido_id
    JOIN cardapio c2     ON cp.cardapio_id = c2.cardapio_id
    JOIN estoque_cardapio ec ON c2.cardapio_id = ec.cardapio_id
    JOIN estoque e       ON ec.estoque_id = e.estoque_id
    JOIN item i          ON e.item_id = i.item_id
    WHERE c.usuario_id = en_usuario_id_clie
        AND p.aprovado = 2 --Pedido Aprovado
        AND c.dt_encerramento IS NULL
    GROUP BY c.comanda_id;

    IF vn_comanda_id IS NOT NULL THEN
        --
        UPDATE comanda SET dt_encerramento = now()
            , vlr_total = vn_tot_pagar
        WHERE comanda_id = vn_comanda_id;
        --
        vn_return := vn_comanda_id;
    ELSE
        vn_return := -1;-- 'Não foi selecionada nenhuma comanda aberta para o usuário em questão!';
    END IF;

    RETURN vn_return;

EXCEPTION

    when others then
        raise notice 'The transaction is in an uncommittable state. '
            'Transaction was rolled back';
        raise notice '% %', SQLERRM, SQLSTATE;
END;
$$ LANGUAGE plpgsql;
```

# Banco de Dados

## Função gerar pedido

```
CREATE OR REPLACE FUNCTION fkg_gera_pedido ( en_usuario_id_clie      int
                                              , en_usuario_id_func      int
                                              , en_unidade_id            int
                                              , en_cardapio_id           int
                                              , en_qtde                   int
                                              ) RETURNS int AS $$

DECLARE
    vn_pedido_id      int;
    vn_comanda_id      int;
    vn_existe_estoque  int;
    vn_cardapio_id     int;
    vn_qtde            int := 0;

BEGIN
    --CONSULTA EXISTENCIA DA COMANDA, SE NÃO CRIA NOVA
    SELECT FKG_VAL_COMANDA(en_usuario_id_clie) INTO vn_comanda_id;
    --VALIDA SE EXISTE ESTOQUE PARA PRODUZIR O PEDIDO, SE SIM JÁ REALIZA UPDATE NO VALOR DO ESTOQUE E DIMINUI A QUANTIDADE
    SELECT FKG_VAL_ESTOQUE(en_unidade_id , en_cardapio_id, en_qtde) INTO vn_existe_estoque;
    --
    IF vn_existe_estoque = 1 THEN
        --CRIA REGISTRO DE PEDIDO

        SELECT p.pedido_id
            INTO vn_pedido_id
        FROM pedido p
        JOIN comanda c ON p.comanda_id = c.comanda_id
        WHERE p.dt_fim IS NULL
        AND p.usuario_id_cliente = en_usuario_id_clie
        AND c.dt_encerramento IS NULL
        ORDER BY p.pedido_id DESC
        LIMIT 1;

        IF vn_pedido_id is NULL THEN

            INSERT INTO pedido (dt_inic, aprovado, dt_fim, comanda_id, usuario_id_cliente, usuario_id_funcionario)
            VALUES
            (now(), 0, NULL, vn_comanda_id, en_usuario_id_clie, en_usuario_id_func) RETURNING pedido_id INTO vn_pedido_id;
            --GRAVA RELACAO DO PEDIDO COM O CARDAPIO E QUANTIDADE
        END IF;
    END IF;
```



# ENGENHARIA DE SOFTWARE



# Engenharia de Software

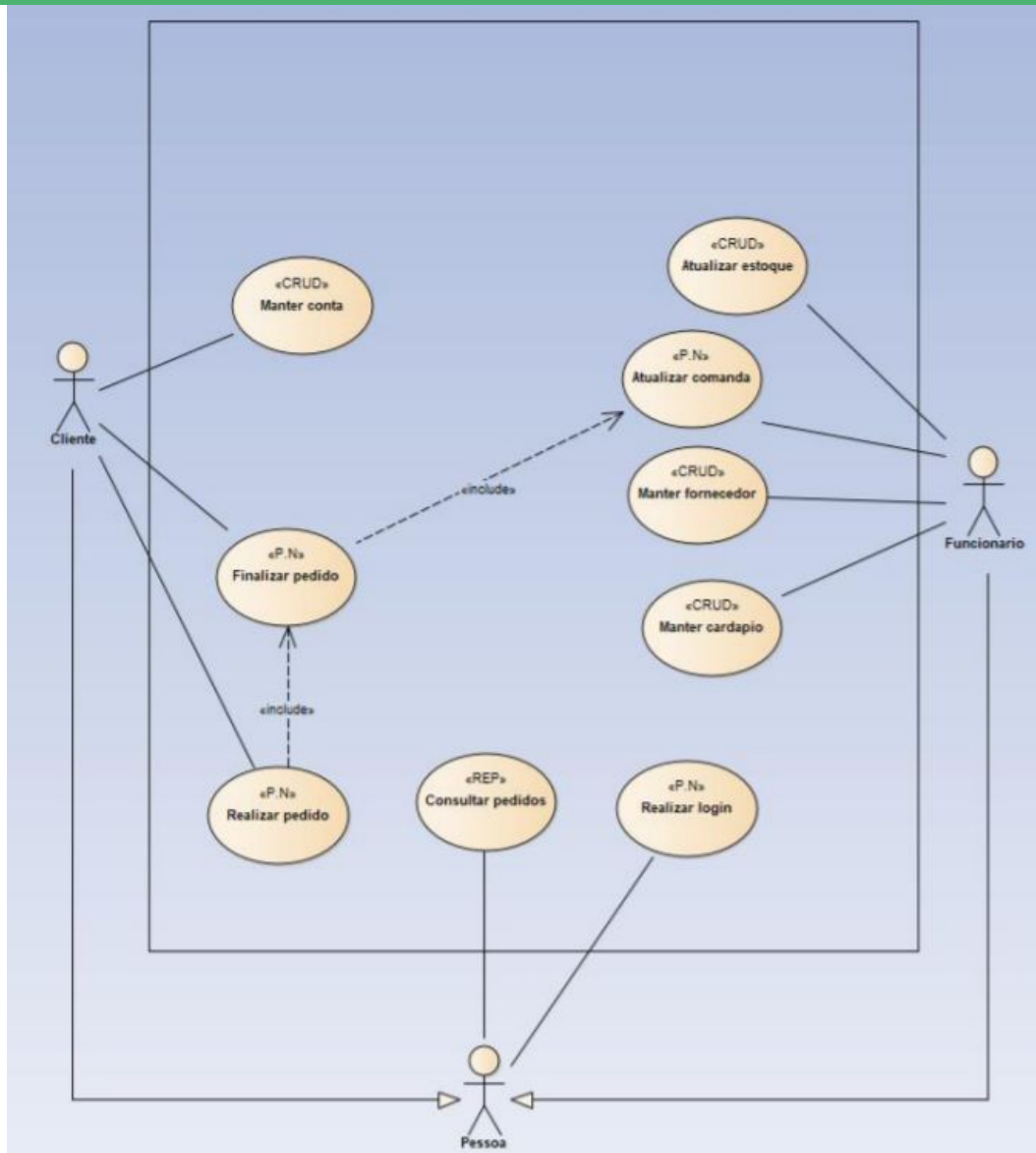
## Utilização do modelo ágil

- Realizado planning para definir tarefas.
- Utilizando board trello.
- Após finalização das tarefas, feita revisão pelos colegas para realização de melhorias.



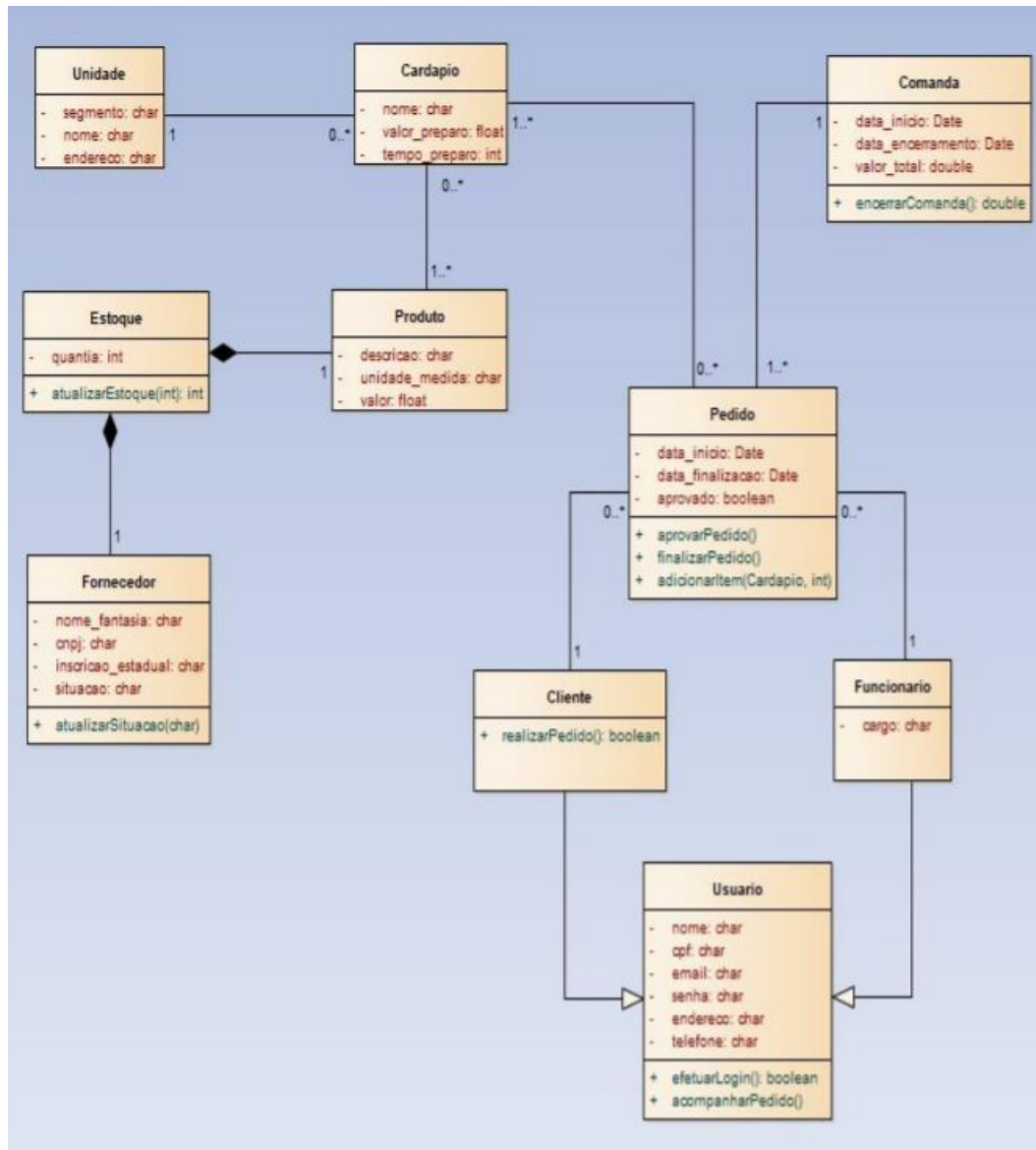
# Engenharia de Software

## Use cases



# Engenharia de Software

## Classes



# Programação

# Programação

Desenvolvida API em java utilizando framework Spring Boot.

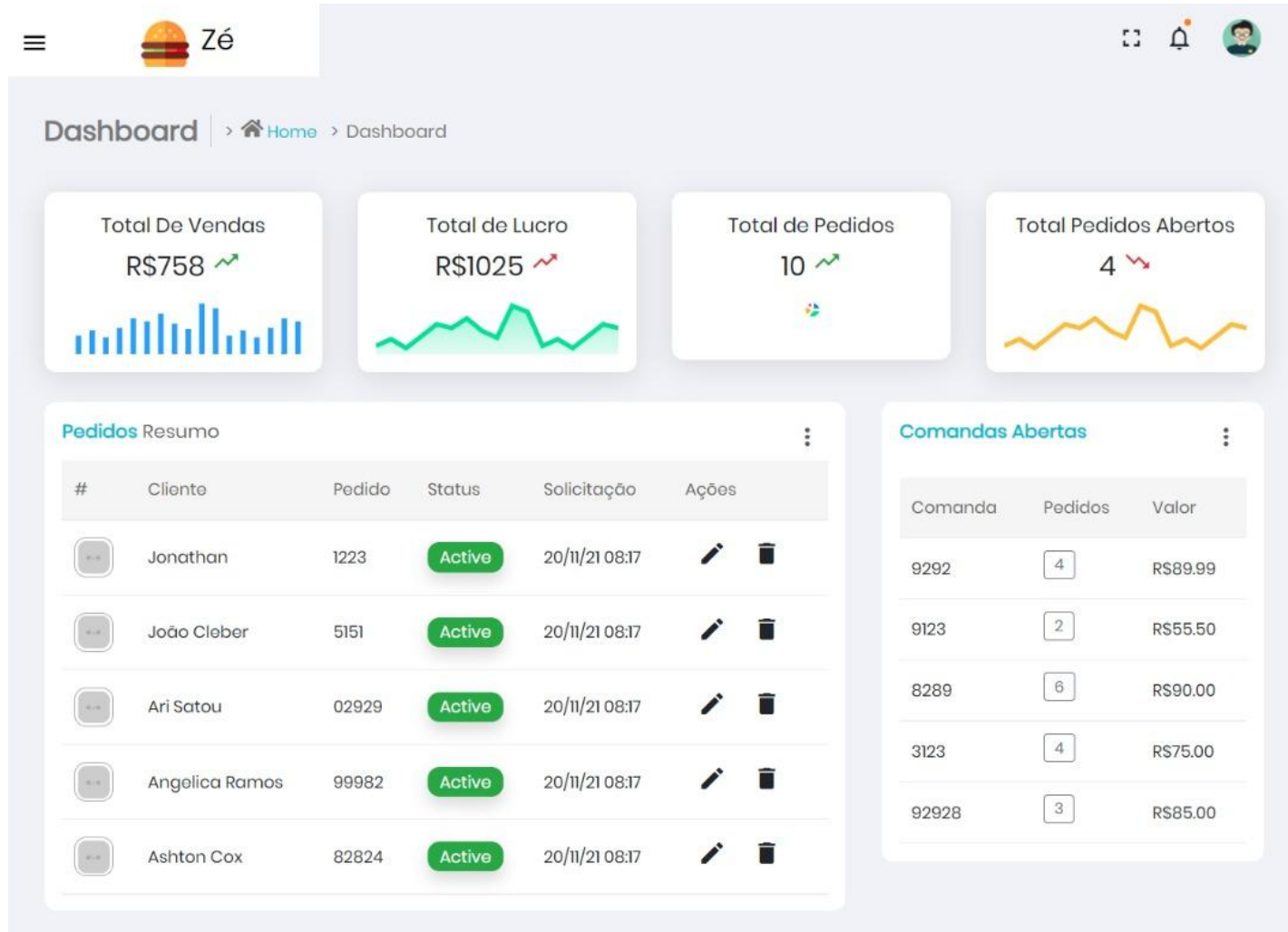
Funcionalidades:

- CRUD de itens
- CRUD de Usuário
- CRUD de Cardapios
- Controle de Fornecedores
- Controle do Estoque
- Fluxo do Pedido
- Autenticação do Usuario
- Controle de sessão com token JWT



# Programação

Frontend desenvolvido com o framework Angular



# Conclusão

Concluimos que a solução desenvolvida é o suficiente para solucionar o que foi proposto.

**Obrigado pela Atenção!**

**Bruno Figueira  
Matheus Giordani  
Jonathan da Cruz**