

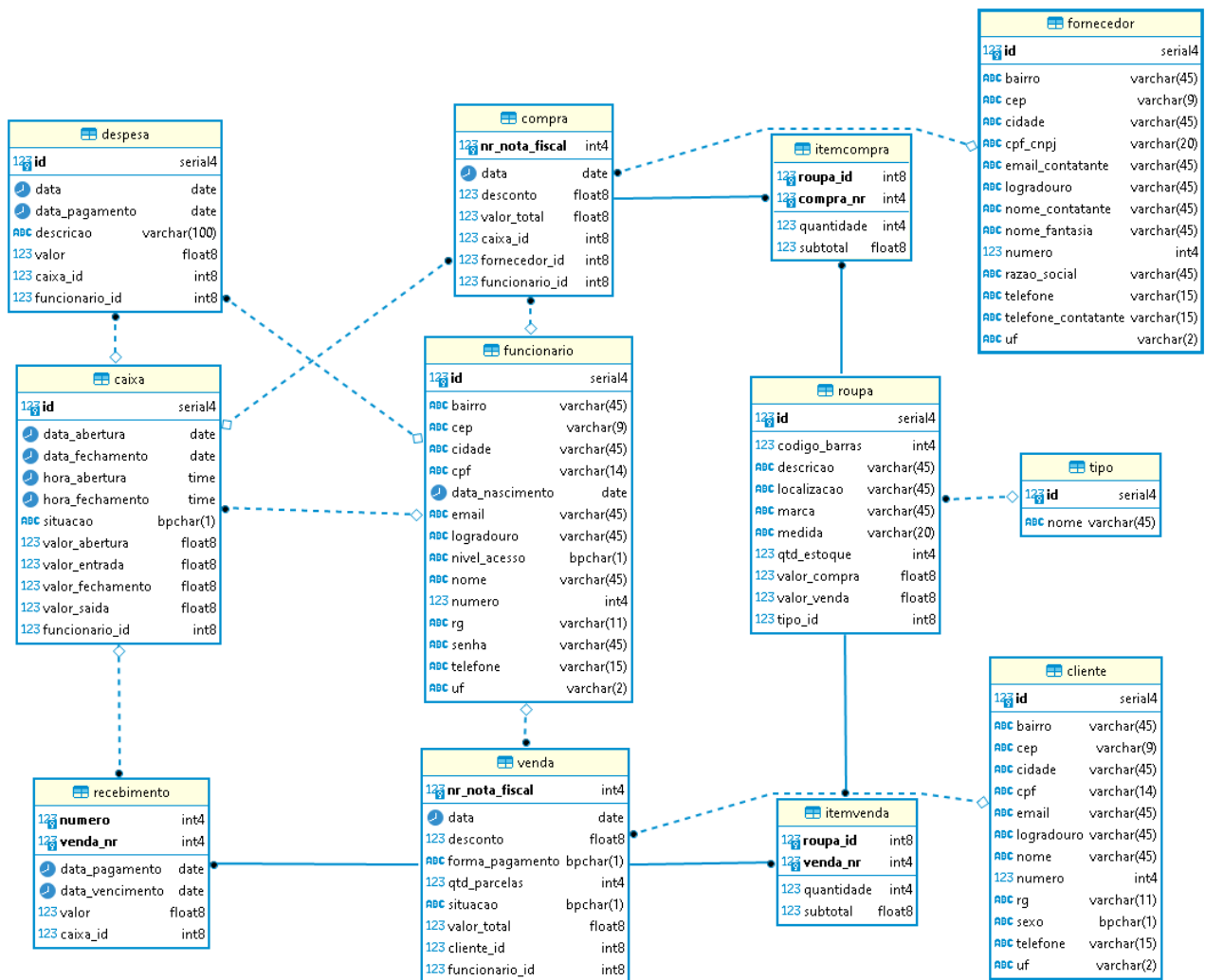
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Presidente Epitácio

Bacharelado em Ciência da Computação – 3º Semestre
Disciplina – Banco de Dados 2 – BD2C4

Envio do Projeto Integrado II

Eduardo Henrique Faustino de Souza - 3005801
Luiz Gustavo da Silva Correa – 3006883

Modelo Lógico)



TRIGGERS)

```
CREATE OR REPLACE FUNCTION cria_recebimentos()
```

```
RETURNS TRIGGER AS $$
```

```
    DECLARE i INT DEFAULT 1;
```

```
BEGIN
```

```
    IF (TG_OP = 'INSERT') THEN
```

```
        IF (NEW.forma_pagamento = 'V') THEN
```

```
            INSERT INTO Recebimento (numero, data_pagamento,
data_vencimento, valor, venda_nr)
```

```

VALUES (1, NEW.data, NEW.data, NEW.valor_total,
NEW.nr_nota_fiscal);
ELSE
FOR i in 1..NEW.qtd_parcelas LOOP
INSERT INTO Recebimento (numero,
data_vencimento, valor, venda_nr)
VALUES (i, (NEW.data+(31*i)),
(NEW.valor_total/NEW.qtd_parcelas), NEW.nr_nota_fiscal);
END LOOP;
END IF;
RETURN NEW;
END IF;
END;
$$LANGUAGE PLPGSQL;

```

```

CREATE TRIGGER insere_recebimentos AFTER INSERT ON Venda
FOR EACH ROW EXECUTE PROCEDURE cria_recebimentos();

```

```

-----

CREATE OR REPLACE FUNCTION atualiza_estoque_venda()
RETURNS TRIGGER AS $$
BEGIN
IF (TG_OP = 'INSERT') THEN
UPDATE Roupas SET qtd_estoque = qtd_estoque -
NEW.quantidade WHERE id = NEW.roupa_id;
RETURN NEW;
END IF;
END;
$$LANGUAGE PLPGSQL;

```

```

CREATE TRIGGER reduz_estoque BEFORE INSERT OR UPDATE ON itemvenda
FOR EACH ROW EXECUTE PROCEDURE atualiza_estoque_venda();

```

```
-----  
  
CREATE OR REPLACE FUNCTION atualiza_estoque_compra()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF(TG_OP = 'INSERT') THEN  
        UPDATE Roupa SET qtd_estoque = qtd_estoque +  
NEW.quantidade WHERE id = NEW.roupa_id;  
        RETURN NEW;  
    END IF;  
END;  
$$LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER aumenta_estoque BEFORE INSERT OR UPDATE ON  
itemcompra  
FOR EACH ROW EXECUTE PROCEDURE atualiza_estoque_compra();  
  
-----
```

```
CREATE OR REPLACE FUNCTION soma_saida_despesa()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF(TG_OP = 'UPDATE') THEN  
        IF(OLD.caixa_id IS NULL AND NEW.caixa_id IS NOT NULL)  
THEN  
            UPDATE Caixa SET valor_saida = valor_saida +  
NEW.valor WHERE id = NEW.caixa_id;  
        END IF;  
        RETURN NEW;  
    END IF;  
END;  
$$LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER adiciona_saida_caixa AFTER UPDATE ON Despesa
FOR EACH ROW EXECUTE PROCEDURE soma_saida_despesa();
```

```
-----

CREATE OR REPLACE FUNCTION soma_saida_compra()
RETURNS TRIGGER AS $$
BEGIN
    IF (TG_OP = 'INSERT') THEN
        IF (OLD.caixa_id IS NULL AND NEW.caixa_id IS NOT NULL)
THEN
            UPDATE Caixa SET valor_saida = valor_saida +
NEW.valor_total WHERE id = NEW.caixa_id;
        END IF;
        RETURN NEW;
    END IF;
END;
$$LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER adiciona_saida_caixa_compra AFTER INSERT ON Compra
FOR EACH ROW EXECUTE PROCEDURE soma_saida_compra();
```

```
-----

CREATE OR REPLACE FUNCTION soma_entrada_caixa()
RETURNS TRIGGER AS $$
BEGIN
    IF (TG_OP = 'INSERT') THEN
        IF (NEW.caixa_id IS NOT NULL) THEN
            UPDATE Caixa SET valor_entrada = valor_entrada +
NEW.valor WHERE id = NEW.caixa_id;
        END IF;
    
```

```
        RETURN NEW;

    END IF;

    IF (TG_OP = 'UPDATE') THEN

        IF (OLD.caixa_id IS NULL AND NEW.caixa_id IS NOT NULL)
THEN
            UPDATE Caixa SET valor_entrada = valor_entrada +
NEW.valor WHERE id = NEW.caixa_id;

            END IF;

            RETURN NEW;

        END IF;

    END;

$$LANGUAGE PLPGSQL;
```

```
CREATE TRIGGER adiciona_valor_caixa AFTER INSERT OR UPDATE ON
Recebimento
FOR EACH ROW EXECUTE PROCEDURE soma_entrada_caixa();
```