



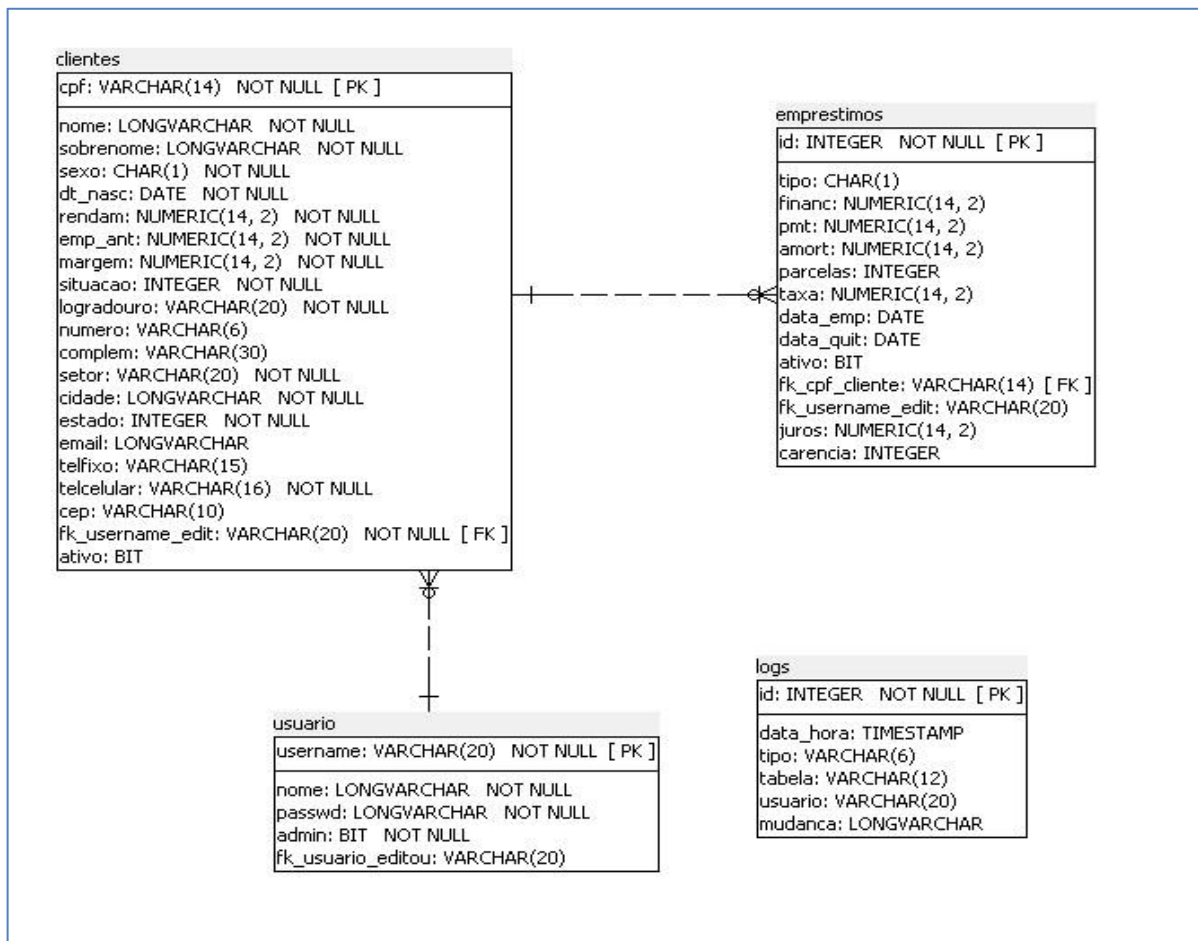
**FACULDADE DE TECNOLOGIA SENAC DE GOIÁS**  
**CURSO GESTÃO DA TECNOLOGIA DA INFORMAÇÃO**  
**BANCO DE DADOS**

**JOÃO PAULO NASCIMENTO OLIVEIRA**  
**PAULO ROBERTO VIEIRA**

**Modelo Relacional e Modelo Físico Banco Dados**

Com base nos conteúdos apresentados em sala de aula foi desenvolvido o modelo relacional e o modelo físico de um banco de dados para um software de simulação de empréstimos de uma casa bancária, esse modelo será utilizado por uma aplicação Java SE para executar as seguintes operações: inserir, apagar, alterar e recuperar dados do banco de dados desenvolvido, com base nas informações existentes no banco de dados a aplicação deve recuperar os dados para a execução dos cálculos necessários para a simulação sugerida.

**Modelo Relacional**





## Modelo Físico

### CREATE TABLE public.usuario

```
(
    nome text NOT NULL,
    username character varying(20) NOT NULL,
    passwd text NOT NULL,
    admin boolean NOT NULL,
    fk_usuario_editou character varying(20),
    CONSTRAINT usuario_pkey PRIMARY KEY (username),
    CONSTRAINT fk_usuario_editou FOREIGN KEY (fk_usuario_editou)
REFERENCES public.usuario (username) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
);
```

### CREATE TABLE public.clientes

```
(
    cpf character varying(14) NOT NULL,
    nome text NOT NULL,
    sobrenome text NOT NULL,
    sexo character(1) NOT NULL,
    dt_nasc date NOT NULL,
    rendam numeric(14,2) NOT NULL,
    emp_ant numeric(14,2) NOT NULL,
    margem numeric(14,2) NOT NULL,
    situacao integer NOT NULL,
    logradouro character varying(20) NOT NULL,
    numero character varying(6),
    complem character varying(30),
    setor character varying(20) NOT NULL,
    cidade text NOT NULL,
    estado integer NOT NULL,
    email text,
    telfixo character varying(15),
    telcelular character varying(16) NOT NULL,
    cep character varying(10),
    fk_username_edit character varying(20),
    ativo boolean NOT NULL,
    CONSTRAINT clientes_pkey PRIMARY KEY (cpf),
    CONSTRAINT fk_username_edit FOREIGN KEY (fk_username_edit)
REFERENCES public.usuario (username) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
);
```

### CREATE TABLE public.emprestimos

```
(
    id serial NOT NULL,
    tipo character(1) NOT NULL,
    financ numeric(14,2),
    pmt numeric(14,2),
    amort numeric(14,2),
    parcelas integer,
    taxa numeric(14,2),
```



```
data_emp date,  
data_quit date,  
ativo boolean,  
fk_cpf_cliente character varying(14),  
fk_username_edit character varying(20),  
juros numeric(14,2),  
carencia integer,  
CONSTRAINT emprestimo_pkey PRIMARY KEY (id),  
CONSTRAINT fk_cpf_cliente FOREIGN KEY (fk_cpf_cliente)  
REFERENCES public.clientes (cpf) MATCH SIMPLE  
ON UPDATE NO ACTION  
ON DELETE NO ACTION,  
CONSTRAINT fk_username_edit FOREIGN KEY (fk_username_edit)  
REFERENCES public.usuario (username) MATCH SIMPLE  
ON UPDATE NO ACTION  
ON DELETE NO ACTION  
);
```

#### **CREATE TABLE public.logs**

```
(  
    id serial NOT NULL,  
    data_hora timestamp without time zone,  
    tipo character varying(6),  
    tabela character varying(12),  
    usuario character varying(20),  
    mudanca text,  
    CONSTRAINT logs_pkey PRIMARY KEY (id)  
);
```

```
INSERT INTO usuario(nome, username, passwd, admin) VALUES ('Administrador', 'admin',  
md5('adminadmin'), true);
```

#### **CREATE OR REPLACE FUNCTION public.cliente\_log()**

**RETURNS** trigger AS \$BODY\$

**DECLARE** changed character varying = '';

**BEGIN**

**IF** (TG\_OP = 'INSERT') THEN

**INSERT INTO** logs(data\_hora, usuario, tipo, tabela, mudanca) VALUES (current\_timestamp,  
NEW.fk\_username\_edit, TG\_OP, TG\_TABLE\_NAME, 'O cliente ' || NEW.cpf || ' foi  
adicionado.');

**RETURN NEW;**

**ELSIF** (TG\_OP = 'UPDATE') THEN

if (new.nome<>old.nome) then changed = changed||' nome de '||old.nome||' para  
'||new.nome||', '; end if;

if (new.sobrenome<>old.sobrenome) then changed = changed||' sobrenome de  
'||old.sobrenome||' para '||new.sobrenome||', '; end if;

if (new.sexo<>old.sexo) then changed = changed||' sexo de '||old.sexo||' para '||new.sexo||', '  
end if;

if (new.dt\_nasc<>old.dt\_nasc) then changed = changed||' dt\_nasc de '||old.dt\_nasc||' para  
'||new.dt\_nasc||', '; end if;

if (new.rendam<>old.rendam) then changed = changed||' rendam de '||old.rendam||' para  
'||new.rendam||', '; end if;

if (new.emp\_ant<>old.emp\_ant) then changed = changed||' emp\_ant de '||old.emp\_ant||' para  
'||new.emp\_ant||', '; end if;



```
if (new.margem<>old.margem) then changed = changed||' margem de '||old.margem||' para
'||new.margem||', '; end if;
if (new.situacao<>old.situacao) then changed = changed||' situacao de '||old.situacao||' para
'||new.situacao||', '; end if;
if (new.logradouro<>old.logradouro) then changed = changed||' logradouro de
'||old.logradouro||' para '||new.logradouro||', '; end if;
if (new.numero<>old.numero) then changed = changed||' numero de '||old.numero||' para
'||new.numero||', '; end if;
if (new.complem<>old.complem) then changed = changed||' complem de '||old.complem||'
para '||new.complem||', '; end if;
if (new.setor<>old.setor) then changed = changed||' nome de '||old.setor||' para '||new.setor||',
'; end if;
if (new.cidade<>old.cidade) then changed = changed||' cidade de '||old.cidade||' para
'||new.cidade||', '; end if;
if (new.estado<>old.estado) then changed = changed||' estado de '||old.estado||' para
'||new.estado||', '; end if;
if (new.email<>old.email) then changed = changed||' email de '||old.email||' para
'||new.email||', '; end if;
if (new.telfixo<>old.telfixo) then changed = changed||' telfixo de '||old.telfixo||' para
'||new.telfixo||', '; end if;
if (new.telcelular<>old.telcelular) then changed = changed||' telcelular de '||old.telcelular||'
para '||new.telcelular||', '; end if;
if (new.cep<>old.cep) then changed = changed||' cep de '||old.cep||' para '||new.cep||', '; end
if;
if (new.ativo<>old.ativo) then changed = changed||' ativo de '||old.ativo||' para '||new.ativo||', ';
end if;
if (new.fk_username_edit<>old.fk_username_edit) then changed = changed||'
fk_username_edit de '||old.fk_username_edit||' para '||new.fk_username_edit||', '; end if;
INSERT INTO logs(data_hora, usuario, tipo, tabela, mudanca) VALUES (current_timestamp,
NEW.fk_username_edit, TG_OP, TG_TABLE_NAME, 'O cliente ' || NEW.cpf || ' foi de
modificado de ' || changed);
RETURN NEW;
END IF;
RETURN NULL;
END;
$BODY$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION public.usuario_log()
RETURNS trigger AS $BODY$
DECLARE changed character varying = '';
BEGIN
IF (TG_OP = 'INSERT') THEN
INSERT INTO logs(data_hora, usuario, tipo, tabela, mudanca) VALUES (current_timestamp,
NEW.fk_usuario_editou, TG_OP, TG_TABLE_NAME, 'O usuario '||NEW.username||' foi
criado!');
RETURN NEW;
ELSIF (TG_OP = 'UPDATE') THEN
changed = 'O usuario '||OLD.username||' mudou: ';
if(OLD.passwd<>NEW.passwd)then changed = changed || ' senha, ';
elseif(OLD.admin<>NEW.admin)then changed = changed || ' admin de '||old.admin||' para
'||new.admin||', ';
elseif(OLD.fk_usuario_editou<>NEW.fk_usuario_editou)then changed = changed || '
fk_usuario_editou de '||old.fk_usuario_editou||' para '||new.fk_usuario_editou||', ';
else changed = changed || 'não houve mudança'; end if;
```



```
INSERT INTO logs(data_hora, usuario, tipo, tabela, mudanca) VALUES (current_timestamp,  
NEW.fk_usuario_editou, TG_OP, TG_TABLE_NAME, changed);
```

```
RETURN NEW;
```

```
END IF;
```

```
RETURN NULL;
```

```
END;
```

```
$BODY$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE FUNCTION public.emprestimos_log()
```

```
RETURNS trigger AS $BODY$
```

```
BEGIN
```

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

```
INSERT INTO logs(data_hora, usuario, tipo, tabela, mudanca) VALUES (current_timestamp,  
NEW.fk_username_edit, TG_OP, TG_TABLE_NAME, 'Empréstimo nº ' || NEW.id || ' para o  
cliente ' || NEW.fk_cpf_cliente);
```

```
RETURN NEW;
```

```
ELSIF (TG_OP = 'UPDATE') THEN
```

```
INSERT INTO logs(data_hora, usuario, tipo, tabela, mudanca) VALUES (current_timestamp,  
NEW.fk_username_edit, TG_OP, TG_TABLE_NAME, 'Empréstimo nº ' || NEW.id || ' para o  
cliente ' || NEW.fk_cpf_cliente || ' foi quitado!');
```

```
RETURN NEW;
```

```
END IF;
```

```
RETURN NULL;
```

```
END;
```

```
$BODY$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_clientes
```

```
AFTER INSERT OR UPDATE
```

```
ON public.clientes
```

```
FOR EACH ROW
```

```
EXECUTE PROCEDURE public.cliente_log();
```

```
CREATE TRIGGER trigger_emprestimos
```

```
AFTER INSERT OR UPDATE
```

```
ON public.emprestimos
```

```
FOR EACH ROW
```

```
EXECUTE PROCEDURE public.emprestimos_log();
```

```
CREATE TRIGGER trigger_usuarios
```

```
AFTER INSERT OR UPDATE
```

```
ON public.usuario
```

```
FOR EACH ROW
```

```
EXECUTE PROCEDURE public.usuario_log();
```