

UNIVERSIDADE DO OESTE DE SANTA CATARINA
CIÊNCIA DA COMPUTAÇÃO

JEAN CARLOS GAVA BARP, MATHEUS FELIPE CHRISTMANN E MICHELL
HENRIQUE DA SILVA SPAGNOL

GAIA

São Miguel Do Oeste, Santa Catarina
2022

JEAN CARLOS GAVA BARP, MATHEUS FELIPE CHRISTMANN E MICHEL
HENRIQUE DA SILVA SPAGNOL

GAIA

Projeto de desenvolvimento de um sistema para gerenciamento de farmácias apresentado ao curso de bacharelado em Ciência da Computação da Universidade do Oeste de Santa Catarina – UNOESC.

Orientadores: Franciele Petry, Otilia Donato Barbosa e Roberson Junior Fernandes Alves.

São Miguel do Oeste, Santa Catarina
2022

ÍNDICE DE FIGURAS

Figura 1: Funcionário e Cargo - Modelo Relacional.....	10
Figura 2: Modelo Relacional.....	11
Figura 3: Diagrama de Caso de Uso.....	12
Figura 4: Diagrama de Sequência - (Cadastro de Pessoa).....	16
Figura 5: Diagrama de Atividades - (Realizar Venda).....	17
Figura 6: Diagrama de Estado - (Status da Pessoa).....	18
Figura 7: Diagrama de Classes.....	18
Figura 8: Script de exemplo Banco de Dados e Tabela de Produto.....	19
Figura 9: Script de exemplo Criação dos índices.....	20
Figura 10: Script Function e Trigger de Auditoria.....	21
Figura 11: Script Function e Trigger de Movimentação de Estoque.....	21
Figura 12: Script Function Cálculo de Estoque.....	22
Figura 13: Script exemplo Usuários e Grupos.....	22
Figura 14: Script exemplo Permissões.....	23
Figura 15: Script View 01.....	23
Figura 16: Script View 02.....	24
Figura 17: Script View 03.....	24
Figura 18: Script View 04.....	25
Figura 19: Tela de Produtos.....	28
Figura 20: Tela de de Princípio Ativo.....	28
Figura 21: Tela de Cargos.....	29
Figura 22: Modal de Pesquisa de Produto.....	29
Figura 23: Modal de Pesquisa de Princípio Ativo.....	30
Figura 24: Modal de Pesquisa de Cargos.....	30
Figura 25: Opções da Navbar superior.....	31
Figura 26: Estrutura de Pastas.....	31

SUMÁRIO

1 INTRODUÇÃO	5
2 LEVANTAMENTO DE REQUISITOS	6
2.1 Introdução ao Levantamento de Requisitos	6
2.2 Entrevista	6
2.3 Requisitos do Sistema	7
3 MODELO RELACIONAL	9
3.1 Introdução ao Modelo Relacional	9
3.2 Construção do Modelo Relacional	10
4 MODELAGEM DE DIAGRAMAS	11
4.1 Diagrama de Caso de Uso	12
4.2 Fluxo de Caso de Uso	13
4.3 Diagrama de Sequência	15
4.4 Diagrama de Atividades	16
4.5 Diagrama de Estados	17
4.6 Diagrama de Classes	18
5 IMPLEMENTAÇÃO DO BANCO DE DADOS	19
5.1 O Banco de Dados e suas Tabelas	19
5.2 Índices	20
5.3 Procedures/Functions e Triggers	20
5.4 Usuários e Grupos (Permissões)	22
5.5 Views (Relatórios)	23
6 POLÍTICA DE BACKUP E RESTORE	25
7 TELAS DO SISTEMA	26
8 CONCLUSÃO	32
REFERÊNCIAS	33

1 INTRODUÇÃO

Nos dias atuais sabemos da importância de um sistema de gerenciamento eficiente no comércio e varejo, graças a ele conseguimos agilizar vários processos para que possamos nos organizar melhor e entregar uma boa experiência para quem frequenta o local. Desde pequenos locais de comércio até os mais altos estabelecimentos, o sistema específico da área é de suma importância para realizar as tarefas como administração do local.

Sabe-se que em farmácias e drogarias a grande variedade de produtos que pode ser encontrada, como remédios, produtos de higiene, estética e entre outros, necessitam de um controle de estoque, como também o armazenamento de suas respectivas características e informações.

De acordo com o texto informatização da farmácia (2020), “Quando uma farmácia faz uso da tecnologia, ou seja, de sistemas voltados para melhorar a administração, fica mais fácil analisar como está o seu cenário dentro do mercado em que atua. Dessa maneira, é possível tomar decisões mais assertivas, seja qual for o tamanho da sua drogaria.”

O sistema Gaia tem a ideia de poder administrar toda a parte de uma farmácia desde produtos que a mesma possui até aos seus funcionários, dando assim uma grande eficiência para o estabelecimento. Com toda a parte de banco de dados até o software programado na linguagem java, o proprietário do estabelecimento tem em mãos uma ampla praticidade referente a administração do seu negócio.

2 LEVANTAMENTO DE REQUISITOS

2.1 Introdução ao Levantamento de Requisitos

O primeiro processo a ser realizado antes da criação de um banco de dados e implementação de um sistema é realizar a coleta e tratamento de requisitos funcionais, não funcionais e regras de negócio. Esta etapa inicial do processo de desenvolvimento de um software visa entender e tratar as necessidades do cliente, ou seja, saber o que deverá ser desenvolvido, sanar dúvidas e compreender como o sistema funcionar

Visando compreender como o banco de dados Gaia deveria ser projetado para possuir uma boa estrutura e atender as necessidades de seus clientes, um plano de levantamento de requisitos foi pensado e executado através de uma entrevista com um profissional da área.

2.2 Entrevista

Algumas dúvidas foram aparecendo ao realizar pesquisas sobre o assunto, desse modo, foi escolhida uma pessoa para poder nos explicar como funciona toda a regra de negócio de uma farmácia desde ao estoque até um cadastro de cliente, e a pessoa que o time de desenvolvedores escolheu foi a Thabata Honse, acadêmica do curso de Farmácia da Universidade Do Oeste de Santa Catarina e atualmente funcionária da Dado Farma LTDA de Cunha Porã/SC.

A empresa em que Thabata trabalha já possui um sistema para o seu gerenciamento, sendo assim, através da troca de mensagens de texto e áudio, algumas perguntas foram realizadas para entender o funcionamento desse software.

2.3 Requisitos do Sistema

Requisitos Funcionais:

RF 01 - O sistema deverá permitir o cadastro de clientes com suas informações básicas;

RF 02 - O sistema deverá possibilitar a distinção no cadastro entre clientes e fornecedores;

RF 03 - O sistema deverá armazenar a data de cadastro do cliente no sistema;

RF 04 - O sistema deverá possibilitar a visualização de relatórios pré-definidos pela gerência;

RF 05 - O sistema deverá permitir o cadastro de funcionários e manipulação de seus dados;

RF 06 - O sistema deve possibilitar o cadastro de cargos para poder vinculá-los aos funcionários já cadastrados;

RF 07 - O sistema deve possibilitar o cadastro de endereços contendo informações de localização(Município, bairro, rua e nº residência) e dados de contato do endereço(Nº telefone, nº celular e email);

RF 08 - O sistema deverá ter "tipos de endereços"(Comercial, Residencial Fixo, Residencial Temporário...) pré-cadastrados para tiparem os endereços dos clientes;

RF 09 - O sistema deverá ter pré-cadastrado uma série de Estados e Municípios;

RF 10 - O sistema deverá permitir o cadastro de produtos com suas informações básicas como nome, descrição e preço;

RF 11 - O sistema deverá permitir o cadastro de princípios ativos para poder vinculá-los aos produtos já cadastrados;

RF 12 - O sistema deve ter tipos de medicamentos(Referência, similar e genérico) e tipos de tarja(Sem tarja, vermelha, amarela, e preta) pré-cadastrados para poder vinculá-los aos produtos já cadastrados;

RF 13 - O sistema deve possuir já cadastrados tipos de pagamentos(Dinheiro, Cheque, Cartão de Crédito...) para poder vinculá-los a processos de compra e venda de produtos;

RF 14 - O sistema deverá possibilitar a realização de compras de produtos onde será informado data, fornecedor, tipo de pagamento, funcionário que está realizando o processo e os dados dos itens a serem comprados;

RF 15 - O sistema deverá possibilitar a realização de vendas de produtos onde será informado data, fornecedor, tipo de pagamento, funcionário que está realizando o processo e os dados dos itens a serem vendidos;

RF 16 - O sistema deve possuir um controle de estoque dos produtos que sofrem movimentação através de compras e vendas;

RF 17 - O sistema deve possibilitar a anexação de receitas médicas durante a venda de produtos caso o cliente apresente o documento;

RF 18 - O sistema deve possibilitar a inserção e manipulação de notificações de compra que são emitidas por um atendente para o comprador caso um produto esteja com o estoque baixo ou não tenha mais estoque;

RF 19 - O sistema deve ter um controle de status de ativo e inativo no cadastro de clientes e produtos;

RF 20 - O sistema deve ter um controle de status de aberto ou concluído no cadastro de notificações de compra;

Requisitos não Funcionais:

RFN 01 - O sistema deverá ter um documento contendo o diagrama de classes, diagrama de caso de uso e demais diagramas usados para criação do sistema;

RFN 02 - O banco de dados utilizado pelo sistema será o PostgreSQL;

RFN 03 - O sistema deverá ser disponibilizado e acessado via web;

RNF 04 - O sistema deverá ter uma interface amigável e intuitiva;

RFN 05 - O sistema deverá realizar consultas com tempo de resposta desejável menor que 10 segundos;

RFN 06 - O sistema deverá ser escrito na linguagem Java;

RFN 07 - O sistema deverá ser implementado com cores que contrastam entre claro e escuro;

RFN 08 - O sistema deve estar disponível para acesso 24 horas por dia;

RNF 09 - O código fonte do sistema deverá ter uma documentação;

RNF 10 - O sistema deve ser desenvolvido seguindo o padrão de orientação a objetos;

RNF 11 - O sistema deverá realizar backups periódicos conforme acordado nas políticas de backup;

3 MODELO RELACIONAL

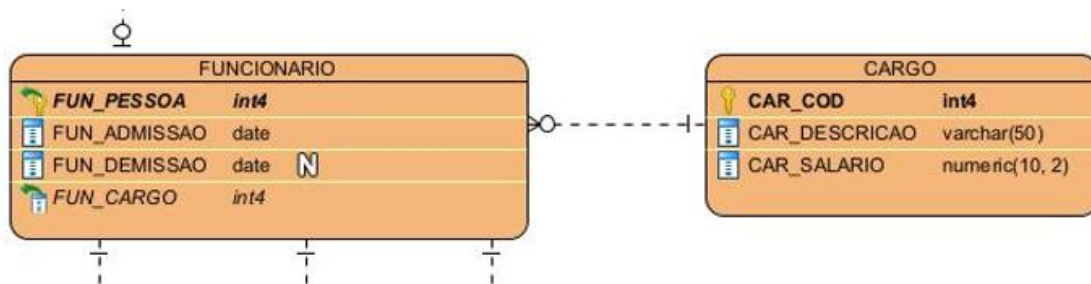
3.1 Introdução ao Modelo Relacional

Com as informações obtidas até o momento, é possível desenvolver o modelo relacional do projeto, ou seja, de trabalharmos os requisitos e os transformá-los em um mapa da estrutura para o nosso banco de dados.

Comumente o modelo relacional é criado após a construção do modelo conceitual (um esboço do banco de dados com alto nível de abstração), no entanto, em nosso projeto unimos os dois processos em um só para acelerarmos o desenvolvimento do mesmo.

O processo de criação do modelo relacional se trata de criarmos visualmente a estrutura que nosso banco de dados terá, ou seja, definir as tabelas, suas devidas ligações, suas colunas e de atribuirmos o tipo de dado que será armazenado em cada coluna.

Conforme a imagem abaixo você encontra um exemplo da criação da nossa tabela FUNCIONÁRIO ligada a tabela Cargo.



Fonte: Autoria própria, 2022.

3.2 Construção do Modelo Relacional

Através do uso da ferramenta Visual Paradigm, fomos capazes de colocar nossas ideias em prática e de arquitetar o nosso modelo onde basicamente seguimos e priorizamos as 5 etapas abaixo (não necessariamente nessa ordem):

1. Criação da tabela e sua colunas;
2. Tipagem correta das colunas;
3. Definição de Chaves Primárias;
4. Ligações da tabela;
5. Criação de documentação para as tabela e suas colunas.

Ao final deste processo obtivemos o seguinte resultado em nosso modelo relacional:

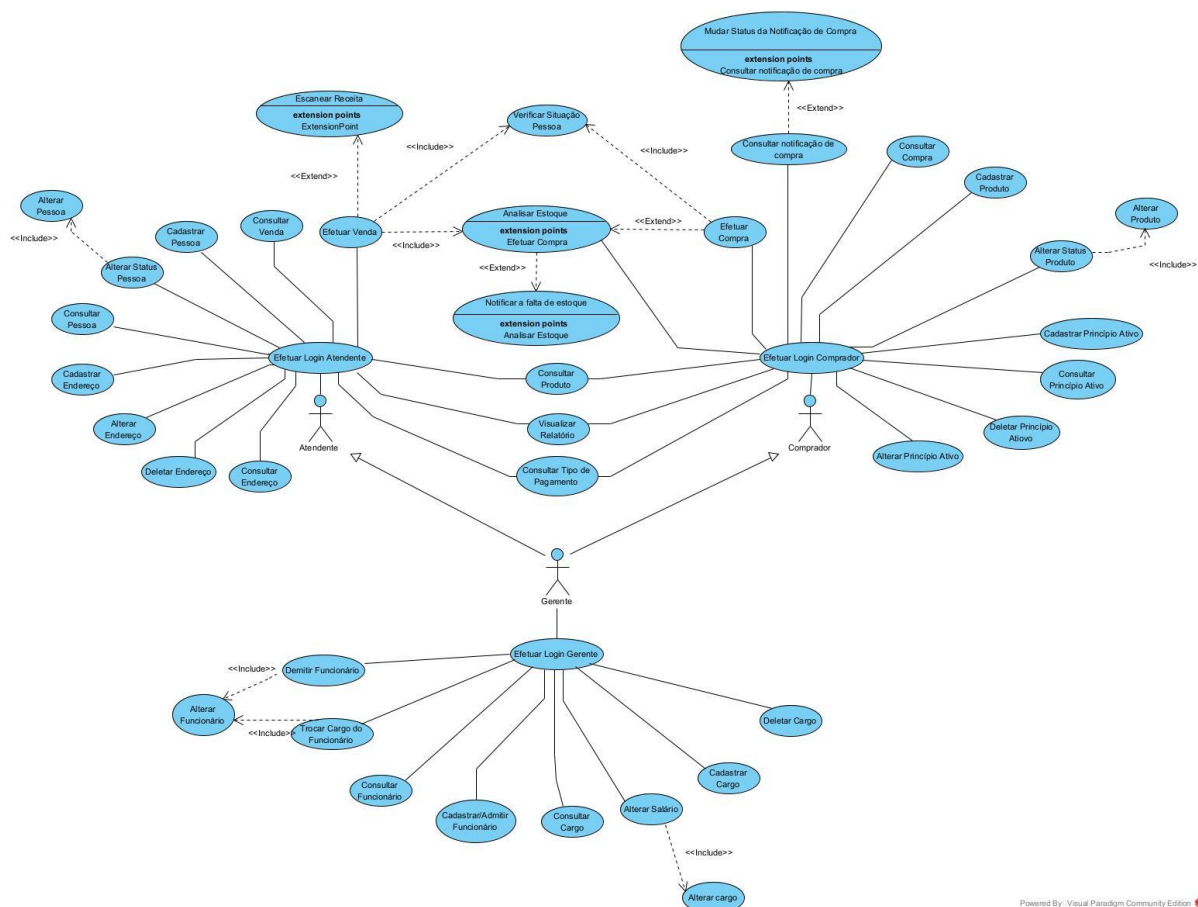
No sistema Gaia foi realizado os seguintes diagramas:

- Diagrama de Caso de Uso;
- Diagrama de Sequência.
- Diagrama de Atividades;
- Diagrama de Estados;
- Diagrama de Classes;

4.1 Diagrama de Caso de Uso

No diagrama de caso de uso é possível ter um detalhamento de como o usuário(Ator) vai poder interagir com o sistema e quais caminhos o mesmo pode seguir com suas interações.

Na figura abaixo é possível observar a arquitetura do diagrama de caso de uso completo do Sistema Gaia.



Fonte: Autoria própria, 2022.

Para uma melhor visualização do diagrama de caso de uso do sistema Gaia, o mesmo é possível ser encontrado pelo caminho em nosso repositório: Diagramas/Diagrama de Caso de Uso.

4.2 Fluxo de Caso de Uso

O fluxo de caso de uso especifica o comportamento de um caso de uso, ou seja, é uma sequência de comandos declarativos que descreve as etapas de execução de um Caso de Uso. Abaixo você confere dois exemplos do sistema relatando o fluxo de caso de uso da realização de uma venda para um cliente e da realização da compra com um fornecedor:

Fluxo de Caso de Uso 01 - Efetuando uma Venda para um Cliente

Ator Primário: Atendente.

Precondições: Cliente a qual vai ser realizado a venda estar devidamente cadastrado no sistema.

Fluxo Normal:

- 1 - Atendente realiza Login no sistema.
- 2 - Atendente verifica a situação do cliente (Está Ativo).
- 3 - Atendente efetua o preenchimento de dados para realizar a venda.
- 4 - Atendente analisa o estoque dos produtos requeridos pelo cliente para venda (Possui estoque).
- 5 - Atendente conclui a venda com sucesso.

Fluxo Alternativo: Cliente está com status inativo.

- 2a - Atendente verifica que o status do cliente está inativo.

2b - Atendente altera o status (Inativo para Ativo) no cadastro do cliente.

Retoma o passo 3.

Fluxo Alternativo: Atendente escaneia receita para realizar a venda.

3a - Cliente apresenta receita médica para a compra de produtos.

3b - Atendente escaneia a receita médica e anexa no sistema.

Retoma o passo 2.

Fluxo Alternativo: Produto a ser vendido sem estoque.

4a - Atendente verifica que não há estoque do(s) produto(s) a ser(em) vendido(s).

4b - Atendente notifica a falta de estoque no sistema.

4c - Atendente informa o cliente da falta de estoque.

Se encerra o processo de venda devido a falta de estoque do(s) produto(s).

Fluxo de Caso de Uso 02 - Efetuando uma compra de um Fornecedor

Ator Primário: Comprador

Precondições: Fornecedor do qual vai ser realizado a compra estar devidamente cadastrado no sistema e com o status ativo.

Fluxo Normal:

1 - Comprador efetua login no sistema.

2 - Comprador verifica a situação do fornecedor (Está Ativo).

3 - Comprador efetua o preenchimento de dados para realizar a compra.

4 - Comprador conclui a compra com sucesso.

Fluxo Alternativo: Comprador analisa o estoque antes de efetuar a compra

3a - Comprador analisa o estoque dos produtos para saber quais produtos estão com estoque mínimo atingido.

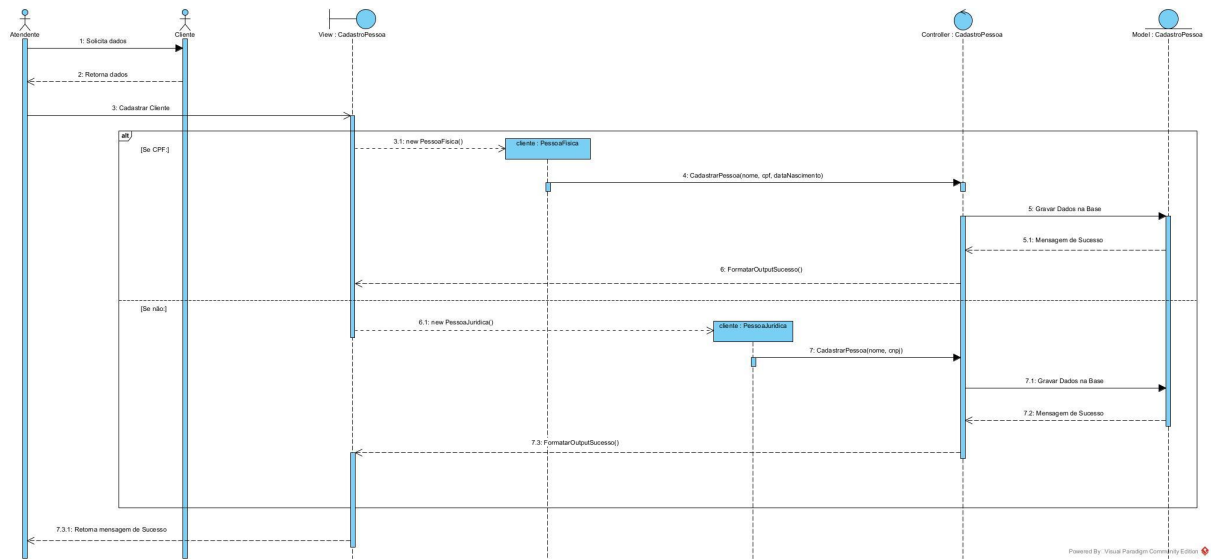
3b - Retoma ao passo 3.

4.3 Diagrama de sequência

Em determinadas situações um objeto pode mandar uma mensagem para um outro objeto, e esse objeto que recebeu a mensagem pode enviar outra mensagem, assim, formando uma sequência de mensagens. O diagrama de sequência detalha essa troca de mensagens e projeta as interações entre objetos do sistema. Existem seis elementos básicos em um diagrama de sequência, sendo eles:

- Atores;
- Objetos e classes;
- Mensagens;
- Linhas de vida e focos de controle;
- Criação e destruição de objetos;
- Iterações.

Na figura abaixo é possível observar um exemplo da arquitetura do diagrama de sequência da realização de cadastro de uma pessoa:

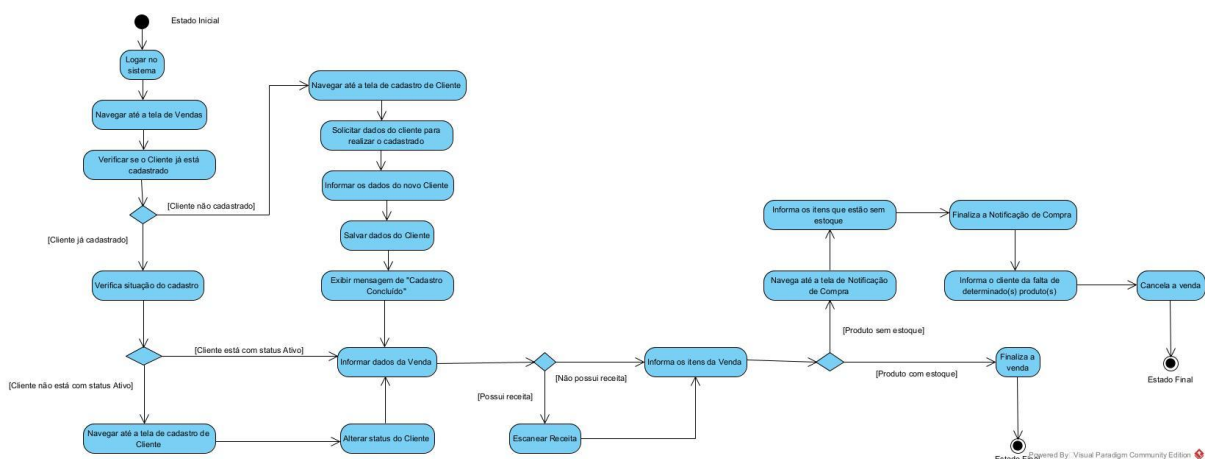


Fonte: Autoria própria, 2022.

Para uma melhor visualização do diagrama de sequência acima e de outro modelo no mesmo projeto, é possível serem encontrados seguinte pelo caminho em nosso repositório: Diagramas/Diagrama Sequência.

4.4 Diagrama de atividades

Este diagrama projeta o caminho a ser percorrido para uma determinada coisa acontecer no sistema, ou seja, vai descrever os passos a passos a serem percorridos para a conclusão de uma venda, por exemplo. Abaixo você confere o diagrama de atividades da realização de uma venda no sistema Gaia:



Fonte: Autoria própria, 2022.

Para uma melhor visualização do diagrama de atividades acima e de outro modelo no mesmo projeto, é possível serem encontrados seguinte pelo caminho em nosso repositório: Diagramas/Diagrama de Atividades.

4.5 Diagrama de Estados

Esse diagrama é uma representação do estado ou situação em que um objeto pode se encontrar no decorrer da execução de processos de um sistema.

Na figura abaixo é possível observar o diagrama de estados do projeto Gaia exemplificando o status de um cliente:

Fonte: Autoria própria, 2022.

Para uma melhor visualização do diagrama de classes acima é possível encontrá-lo pelo seguinte caminho em nosso repositório: Diagramas/Diagrama de Classes.

5 BANCO DE DADOS

Todos os comandos para criação do banco de dados, suas tabelas e documentação(comentários), índices, procedures e triggers, usuários e grupos e views podem ser encontrados pelo seguinte caminho em nosso repositório: Banco de Dados/Scripts/...

5.1 O Banco de Dados e suas Tabelas

```
create database gaiadb;
comment on database "gaiadb" is 'Banco de Dados do sistema Gaia';

create table produto(
    pro_cod serial constraint produto_pk primary key,
    pro_nome varchar(40) not null,
    pro_descricao varchar(70),
    pro_preco numeric(10,2) not null,
    pro_status char(1) constraint pro_status_ck check (pro_status in ('A', 'I')) not null,
    pro_principioativo integer constraint pro_principio_ativo_fk references principio_ativo(pro_cod) not null,
    pro_tipmedicamento char(1) not null,
    pro_tiptarja char(1) not null
);

comment on table produto is 'Cadastro de produtos/medicamentos.';
comment on column produto.pro_cod is 'Código do produto.';
comment on column produto.pro_nome is 'Nome do produto.';
comment on column produto.pro_descricao is 'Descrição do produto.';
comment on column produto.pro_preco is 'Preço de venda do produto.';
comment on column produto.pro_status is 'Campo que indica se o cadastro do produto está inativo ou ativo (I = inativo, A = ativo).';
comment on column produto.pro_principioativo is 'Princípio ativo do produto.';
comment on column produto.pro_tipmedicamento is 'Tipo de medicamento do produto. (R = Referência, S = Similar, G = Genérico).';
comment on column produto.pro_tiptarja is 'Tipo de tarja do produto (S = Sem Tarja, A = Amarela, V = Vermelha, P = Preta).';
```

Fonte: Autoria própria, 2022.

5.2 Índices

Índice, no contexto da estrutura de dados, é uma referência associada a uma chave, que é utilizada para fins de otimização, permitindo uma localização mais rápida de um registro quando efetuada uma consulta.

```
create index com_pessoa_sk
on compra(com_pessoa);

create index com_pagamento_sk
on compra(com_pagamento);

create index com_funcionario_sk
on compra(com_funcionario);

create index coi_numcompra_sk
on compra_item(coi_numcompra);

create index coi_produto_sk
on compra_item(coi_produto);

create index pro_nome_sk
on produto(pro_nome);

create index pra_descricao_sk
on principio_ativo(pra_descricao);

create index pes_nome_sk
on pessoa(pes_nome);

create index mun_nome_sk
on municipio(mun_cod);
```

Fonte: Autoria própria, 2022.

5.3 Procedures/Functions e Triggers

Procedures e triggers podem executar códigos em SQL. A trigger funciona com um gatilho, quando é executado comandos como insert, update ou delete em alguma tabela é disparada a trigger. Procedure seria um bloco de comandos ou instruções SQL que são ajustados para executar uma ou mais tarefas. Ela pode ser acionada através de uma chamada simples por exemplo, e que executa uma serie de outros comandos.

```

create or replace function func_auditoria() returns trigger as
$body$
begin
    if (tg_op = 'DELETE') then
        insert into auditoria(aud_tabela, aud_usuario, aud_data, aud_operacao, aud_regold)
            select tg_relname, user, current_timestamp, 'D', old::text;
        return old;
    elsif (tg_op = 'UPDATE') then
        insert into auditoria(aud_tabela, aud_usuario, aud_data, aud_operacao, aud_regnew, aud_regold)
            select tg_relname, user, current_timestamp, 'A', new::text, old::text;
        return new;
    elsif (tg_op = 'INSERT') then
        insert into auditoria(aud_tabela, aud_usuario, aud_data, aud_operacao, aud_regnew)
            select tg_relname, user, current_timestamp, 'I', new::text;
        return new;
    end if;
    return null;
end;
$body$
language plpgsql;

create trigger estado_audit_trig
after insert or update or delete on estado
for each row execute procedure func_auditoria();

create trigger municipio_audit_trig
after insert or update or delete on municipio
for each row execute procedure func_auditoria();

```

Fonte: Autoria própria, 2022.

```

/* FUNCTION PARA INSERIR REGISTROS NA TABELA DE PRODUTO_MOVIMENTO - (VENDA) */
create or replace function func_movestoque_venda() returns trigger as
$body$
declare
    pessoa integer;
begin
    if (tg_op = 'INSERT') then
        select venda.ven_pessoa into pessoa from venda inner join venda_item on (venda.ven_numero = new.vei_numvenda); -- Busca a Pessoa envolvida
                                                -- na Movimentação

        insert into produto_movimento(mov_data, mov_pessoa, mov_produto, mov_qtd_produto) -- Insere o registro na tabela
            select current_timestamp, pessoa, new.vei_produto, -new.vei_qtd_item; --Lança um valor negativo na tabela pois é uma saída de estoque

        return new;

    elsif (tg_op = 'DELETE') then
        select venda.ven_pessoa into pessoa from venda inner join venda_item on (venda.ven_numero = old.vei_numvenda); -- Busca a Pessoa envolvida
                                                -- na Movimentação

        insert into produto_movimento(mov_data, mov_pessoa, mov_produto, mov_qtd_produto) -- Insere o registro na tabela
            select current_timestamp, pessoa, old.vei_produto, old.vei_qtd_item; --Lança um valor positivo na tabela pois o usuário exclui o item,
                                                -- ou seja, volta o estoque anterior

        return old;
    end if;
    return null;
end
$body$
language plpgsql;

create trigger mov_estoque_venda_trig
after insert or delete on venda_item
for each row execute procedure func_movestoque_venda();

```

Fonte: Autoria própria, 2022.

```

create or replace function func_calc_estoqueproduto(cod_produto integer) returns integer as
$body$
declare
    totalEstoque integer;
begin
    select sum(mov_qtd_produto) into totalEstoque from produto_movimento where mov_produto = cod_produto group by mov_produto;
    return totalEstoque;
end
$body$
language plpgsql;

```

Fonte: Autoria própria, 2022.

5.4 Usuários e Grupos (Permissões)

Em um banco de dados existem privilégios para os usuários e grupos de usuários, como exemplo no sistema Gaia onde há o superusuário, que seria o dono da base, denominado gaiadba, os grupos gerente, comprador e atendente e por fim os usuários comuns.

```

/* USUÁRIOS */
create user gaiadba with superuser password '@gaiadba2022@';

create user otilia with password '@otilia123';
create user roberson with password '@roberson123';
create user franciele with password '@franciele123';
create user matheus with password '@matheus123';
create user michell with password '@michell123';
create user jean with password '@jean123';

/* GRUPOS */
create group atendente;
create group comprador;
create group gerente;

```

Fonte: Autoria própria, 2022.


```

/* PERMISSÕES PARA O GRUPO COMPRADOR */

grant select on relatorio01_vw, relatorio02_vw, relatorio03_vw
to comprador;

grant select, insert, update, delete on principio_ativo
to comprador;

grant select, insert, update on produto
to comprador;

grant select, insert on compra, compra_item
to comprador;

grant select on pessoa
to comprador;

grant select, insert, update on notificacao_compra
to comprador;

grant select on tipo_pagamento
to comprador;

```

Fonte: Autoria própria, 2022.

5.5 Views (Relatórios)

A linguagem SQL é muito poderosa quando o assunto é geração de relatórios, pois podemos nos utilizar de comandos de consulta que retornam os dados conforme nossas necessidades. Através dos comandos de consulta (Conhecidos como Select's) podemos retornar os dados de acordo com certas condições a serem atendidas, ordenar, agrupar, contar e até ligar as tabelas entre si para retornar dados de outras tabelas. Abaixo você confere 4 requisições de consulta para geração de relatórios do sistema Gaia:

1 - Relacionar o código, nome e tipo de todos os medicamentos. Ordene o relatório de forma ascendente pelo nome;

```

create view relatorio01_vw as
select
    produto.pro_cod,
    produto.pro_nome,
    case produto.pro_tipmedicamento
        when 'R' then 'Referência'
        when 'S' then 'Similar'
        else 'Genérico'
    end as Tipo_Medicamento
from produto
order by produto.pro_nome asc;

```

Fonte: Autoria própria, 2022.

2 - Relacionar o nome do medicamento e o nome do fornecedor para todos os medicamentos. Filtre somente produtos com estoque maior que 10. Ordene o relatório de forma descendente pelo nome do fornecedor;

```
create view relatorio02_vw as
select distinct
    produto.pro_nome,
    pessoa.pes_nome
from produto
inner join produto_movimento on (produto_movimento.mov_produto = produto.pro_cod
and produto_movimento.mov_qtd_produto > 0)
inner join pessoa on (pessoa.pes_cod = produto_movimento.mov_pessoa)
where produto.pro_cod in (select
                            mov_produto
                            from produto_movimento
                            group by mov_produto
                            having sum(mov_qtd_produto) > 10)
and pessoa.pes_tipo = 'J'
order by pessoa.pes_nome desc;
```

Fonte: Autoria própria, 2022.

3 - Relacionar o código do medicamento, nome do medicamento, mês da venda, quantidade total de vendas no mês para meses entre anos de 2020 e 2021. Ordene o relatório do produto com mais vendas (em termos de quantidade) para o produto com menos vendas;

```
create view relatorio03_vw as
select
    produto.pro_cod,
    produto.pro_nome,
    extract(month from venda.ven_data) as "Mes",
    extract(year from venda.ven_data) as "Ano", -- Serve para que os meses não se agrupem com anos diferentes
    sum(venda_item.vei_qtd_item) as "Qtd Total Vendas"
from produto
inner join venda_item on (venda_item.vei_produto = produto.pro_cod)
inner join venda on (venda.ven_numero = venda_item.vei_numvenda)
where extract(year from venda.ven_data) in (2020,2021)
group by 1, 2, 3, 4
order by "Qtd Total Vendas" desc;
```

Fonte: Autoria própria, 2022.

4 - Relacionar o CPF do cliente, nome do cliente e o total gasto em medicamentos. Filtrar somente clientes do sexo masculino, com idades entre 20 e 40 anos e que realizaram compras em meses pares de 2021. Ordene o relatório do cliente com mais gastos para o cliente com menos gastos.


```

create view relatorio04_vw as
select
    pesfisica.fis_cpf,
    pessoa.pes_nome,
    sum(venda_item.vei_total_item) as "Total em Compras"
from pessoa
inner join pesfisica on (pesfisica.fis_pessoa = pessoa.pes_cod)
inner join venda on (venda.ven_pessoa = pessoa.pes_cod)
inner join venda_item on (venda.ven_numero = venda_item.vei_numvenda)
where pesfisica.fis_genero = 'M'
and extract (year from age(current_date , pesfisica.fis_datanascimento)) between 20 and 40
and mod(extract(month from venda.ven_data),2) = 0 and extract(year from venda.ven_data) = 2021
group by 1, 2
order by "Total em Compras" desc;

```

Fonte: Autoria própria, 2022.

6 POLÍTICAS DE BACKUP E RESTORE

Esta Política de Backup e Restauração de Dados objetiva instituir diretrizes e responsabilidades que visam à segurança, proteção e disponibilidade dos dados definidos como de necessária salvaguarda para os clientes do Sistema Gaia. É fundamental estabelecer mecanismos que permitam a guarda dos dados e sua eventual restauração em casos de indisponibilidades ou perdas por erro humano, ataques, catástrofes naturais ou outras ameaças. O presente documento apresenta a Política de Backup e Restauração de Dados, onde se estabelece o modo e a periodicidade de cópia dos dados armazenados pelos sistemas computacionais.

Responsável pela Política de Backup: Matheus Felipe Christmann.

Aprovado por: Gerente da Farmácia (Cliente).

Data da Aprovação da Política: Mesma data de Implantação do Sistema.

Localização do Armazenamento: Todos os backup's serão armazenados em nuvem.

Revisão da Política: A presente política de backup e restore será sujeita a revisões e possíveis atualizações em um período de a cada 2 meses visando aperfeiçoar as técnicas já utilizadas e encontrar possíveis falhas que possam vir a ocorrer.

Backup's:

- Backup Físico programado para ser realizado 1 vez por dia as 03:00;
- Backup Lógico programado para ser realizado 2 vezes por dia as 12:00 e 18:00;
- Backup de logs programado para ser realizado 3 vezes por dia as 09:00, 15:00 e as 21:00.

Programação:

03:00: Backup Físico - Cold Backup
09:00: Backup de Logs - Hot Backup
12:00: Backup Lógico - Hot Backup
15:00: Backup de Logs - Hot Backup
18:00: Backup Lógico - Hot Backup
21:00: Backup de Logs - Cold Backup

Levando em conta uma farmácia que abre suas portas as 06:00 da manhã e encerra o expediente as 21:00, o caso mais grave de perda de dados com a política acima será de no máximo 03:00 horas. Caso a farmácia tenha o expediente de 24 horas, o caso mais grave de perda de dados com a política acima será de no máximo 06:00 horas, no entanto vale frisar que entende-se que no período noturno há menos movimentação dentro do estabelecimento, conseqüentemente diminuindo o "fluxo de alterações" no banco de dados.

Restore e Testes:

Os testes de restore da base de dados estão programados para ocorrerem diariamente para garantir a disponibilidade dos dados em caso de perdas ou catástrofes.

Comando para Backup Físico:

```
@echo off
set x=%DATE:~0,2%-~3,2%-~6,4%
echo %x%
set date=%x%
echo %date%
    set PGUSER=gaiadba
    set BACKUP_FILE=C:\Backup\gaiadbBackup_%date%.backup
    echo backup file name is %BACKUP_FILE%
    SET PGPASSWORD=@gaiadba2022@
    echo on
    "C:\Program Files\PostgreSQL\14\bin\pg_dump.exe" -h localhost -p 5432 -U
gaiadba -C -F p -v -f %BACKUP_FILE% gaiadb
pause
```

Comando para Restore do Backup Físico:

```
@echo off
    set PGUSER=gaiadba
    SET PGPASSWORD=@gaiadba2022@
    echo on
    "C:\Program Files\PostgreSQL\14\bin\psql.exe" -h localhost -p 5432 -d gaiadb -U
gaiadba -f C:\Backup\AQUI VAI O NOME DO BACKUP.backup
pause
```

7 TELAS DO SISTEMA

Produtos

Código:

Nome:

Preço:

Status:

P. Ativo:

Tipo de Medicamento:

Tarja:

Descrição:


Fonte: Autoria própria, 2022.

Princípio Ativo

Código:

Descrição:

Fonte: Autoria própria, 2022.

 Cadastros ▾

Cargos

Código:

Descrição:

Descrição

Salário:


Salário

Novo

Gravar

Pesquisar

Fonte: Autoria própria, 2022.

Pesquisa de Produtos: 

Informe o Nome:

p

Buscar

Cód	Nome	Preço	Status		
2	Produto 01	10	A	<div>Ver</div>	<div>Deletar</div>

Fechar

Fonte: Autoria própria, 2022.

Pesquisa de Princípio Ativo:

Informe o Nome/Descrição:

Buscar

Cód	Descrição	
1	Prin 01	Ver
2	Prin 02	Ver
4	Princípio Ativo 01	Ver

Fechar

Fonte: Autoria própria, 2022.

Pesquisa de Princípio Ativo:

Informe o Nome/Descrição:

Buscar

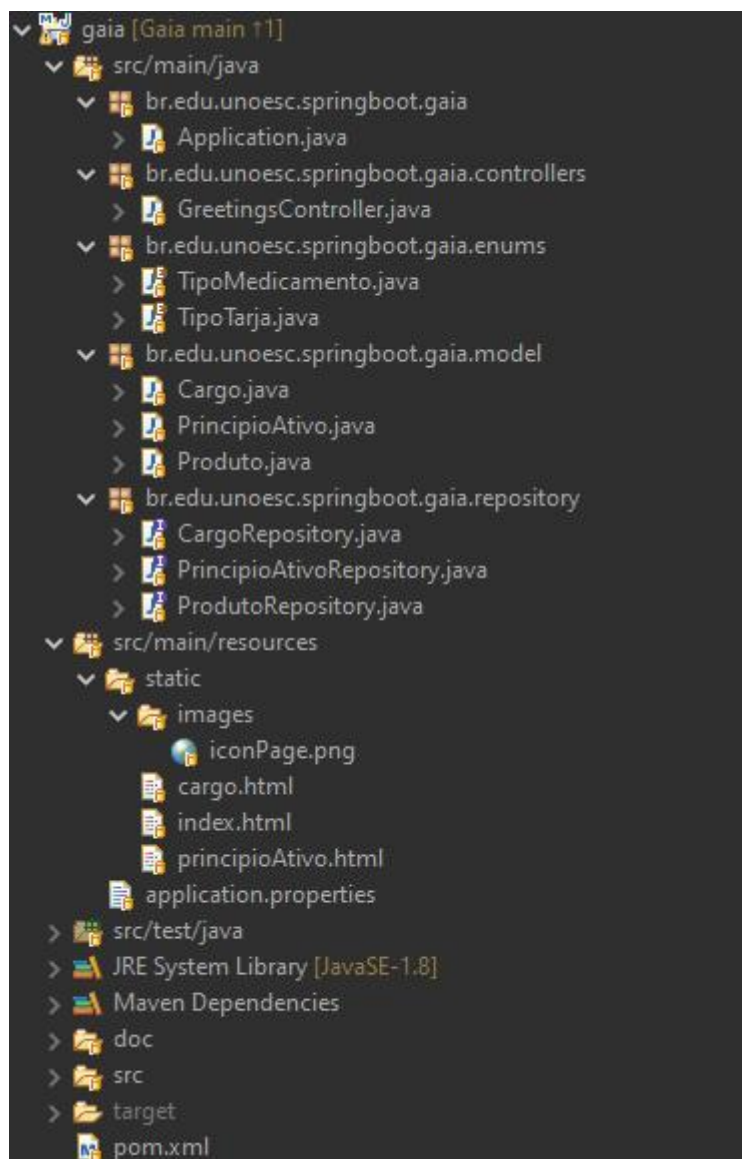
Cód	Descrição	
1	Prin 01	Ver
2	Prin 02	Ver
4	Princípio Ativo 01	Ver

Fechar

Fonte: Autoria própria, 2022.



Fonte: Autoria própria, 2022.



Fonte: Autoria própria, 2022.

CONCLUSÃO

Quando decidimos realizar o curso de Ciência da Computação estávamos cientes que muitos desafios surgiriam ao longo da nossa graduação. Ao realizar o presente trabalho, a necessidade de coleta de informações por meio de terceiros e pesquisa extracurricular foi fundamental para que o trabalho fosse realizado com êxito.

Por se tratar de um ramo em que os integrantes do time não possuíam muito conhecimento, a busca por soluções em artigos da internet e conversas por aplicativos de comunicação com pessoas que estão integrados no assunto foram de grande utilidade para que pudéssemos entregar um banco de dados funcional e prático.

Conclui-se que apesar das dificuldades que apareceram ao decorrer do tempo em que o time estava realizando o trabalho, os conhecimentos adquiridos foram de extrema importância para formação de um raciocínio lógico e eficiente, consciência da administração de tempo e utilização de ferramentas novas.

REFERÊNCIAS

- Informatização da farmácia: um ponto fundamental para a diferenciação.** 2020. Disponível em: <<https://www.espacofarmaceutico.com.br/blog/2020/01/10/informatizacao-da-farmacia-um-ponto-fundamental-para-a-diferenciacao#:~:text=Quando%20uma%20farm%C3%A1cia%20faz%20uso,o%20ta%20manho%20da%20sua%20drogaria>>. Acesso em: 16 jul. 2022.
- HONSE, Thabata. **“Coleta de Requisitos.”** Entrevistadores: CHRISTMANN, Matheus F. Cunha Porã. Entrevista concedida em: 17 abr, 2022.
- MACHADO, Diego. **Normalização em Banco de Dados.** 2015. Disponível em: <<https://medium.com/@diegobmachado/normaliza%C3%A7%C3%A3o-em-banco-de-dados-5647cdf84a12>>. Acesso em: 17 jul. 2022.
- ALMEIDA. F. G; FARINA. R. M. **Rede de Farmácias: Um Estudo Utilizando Banco de Dados Distribuídos.** 2018. 17f. Pesquisa (Bacharelado em Engenharia da Computação) - Centro Universitário de Araraquara Uniara, Araraquara, São Paulo, 2018.
- OLIVEIRA, M. A. D. **Sistema Gerenciamento de Drogaria.** 2008. 77 f. TCC (Bacharelado em Tecnologia e Processamento de Dados) - IMESA - Instituto Municipal de Ensino Superior de Assis, Assis, São Paulo, 2008.
- ZANDONADI, F. H. M. **Sistema Para Gerenciamento de Farmácia.** 2014. 48 f. TCC (Bacharelado em Análise e Desenvolvimento de Sistemas) - IMESA - Instituto Municipal de Ensino Superior de Assis, Assis, São Paulo, 2014.
- OLIVEIRA, A. C. R. **Análise e desenvolvimento de um sistema Mobile e Web utilizando API para comparação de Preços de Medicamentos das Farmácias de Iporá - GO.** 2019. 52f. Trabalho de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal Goiano, Iporá, Goiás, 2019.

LOPES, B. M. S. **Sistema Web para farmácia Solidária utilizando Microserviços**. 2018. 101f. Monografia (Bacharelado em Sistemas de Informação) - Universidade Federal de Juiz de Fora, Juiz de Fora, Minas Gerais, 2018

ALVES, R. J. F. **Apostila de Banco de Dados**. 2021. 159f. Apostila (Bacharelado em Ciência da Computação) - Universidade do Oeste de Santa Catarina, São Miguel do Oeste, Santa Catarina, 2021.

GREZELE, D. F; COSTA, E. M; KLUNK, C. A. S; SCHWINGEL, M. F. A. I. **Trabalho Final: PandemicStats**. 2021. 26f. Trabalho de Curso (Bacharelado em Ciência da Computação) - Universidade do Oeste de Santa Catarina, São Miguel do Oeste, Santa Catarina, 2021.

LIMA, C. **Introdução ao Maven, aprenda como criar e gerenciar projetos Java**. 2021. Disponível em: <<https://www.treinaweb.com.br/blog/introducao-ao-maven-aprenda-como-criar-e-gerenciar-projetos-java>>. Acesso em: 1 out. 2022.

Modelagem de software com UML. Disponível em:<https://www.devmedia.com.br/modelagem-de-software-com-uml/20140> . Acessado em: 08/11/2022.

Diagrama de caso de uso UML: O que é, como fazer e exemplos. Disponível em:<https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml> . Acessado em: 25/11/2022.

PROCEDURES. Disponível em: https://materialdornel.readthedocs.io/pt_BR/latest/linguagem-sql/procedures.html. Acessado em 02/12/2022.

O que é UML e Diagramas de Caso de Uso: Introdução Prática á UML. Disponível em: <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>. Acessado em 08/11/2022.