

T1. Acordos do time SCRUM – Grupo 3

DISPONÍVEL EM: <https://github.com/RosembergAraujo/ibtcg3.api>

1. Equipe

Nome completo	Papel
Rosemberg Vieira Araújo Filho	Desenvolvedor
Lucas Pontes Santana	Documentação
Tobias Gustavo da Silva Soares	Banco de Dados

2. Política de equipe

Foi definido uma rotina de reuniões diárias no aplicativo Discord, no qual definimos como seria feito o projeto, organização, apresentação entre outros. Ademais, também tivemos reuniões com o professor nós auxiliando na estruturação do projeto.

3. Acordos de trabalho

A data limite para finalização de cada fase foi definida para o dia 16/08/2022.

4. Definições para o kanban

Fases:

3 fases, sendo elas:

- 1)To-Start
- 2)In progress
- 3)Completed

Regra de atribuição de responsável:

Documentação: Lucas Pontes

Criação da aplicação: Rosemberg Vieira

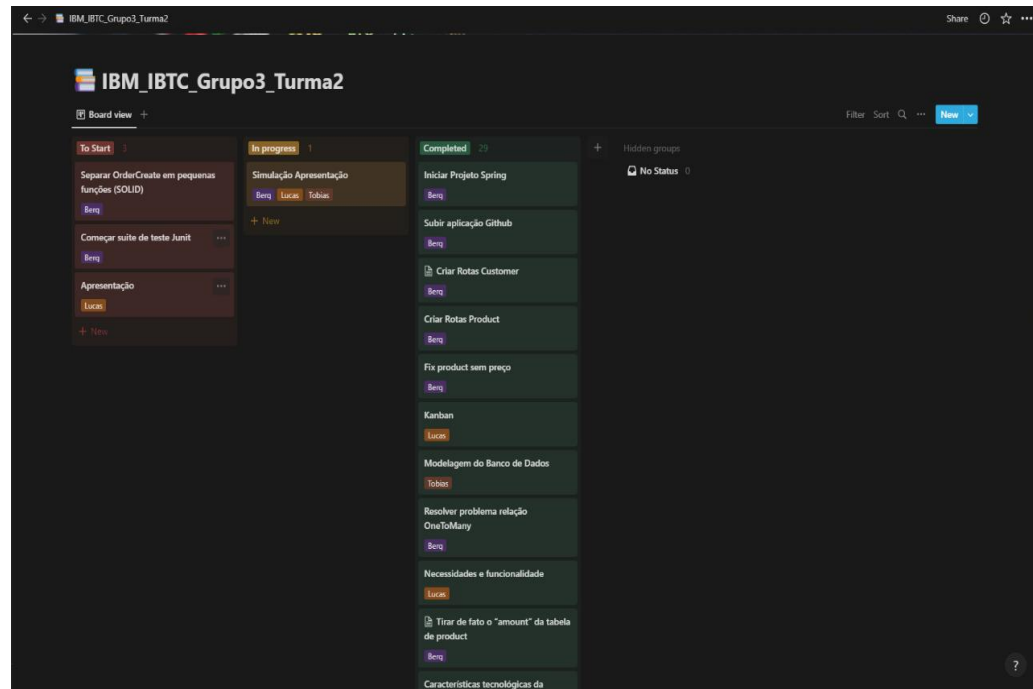
Banco de dados: Tobias Gustavo

Slides:

Ferramenta do kanban: Notion

5. Configurações

Ambiente de desenvolvimento



Acesso:

<https://dent-lilac-5c5.notion.site/a74f4870bf0d418a99c0ee0c070cf36a?v=a71f2ea9baa84c3f9a07d9e52cfc4b32>

T2. Documento de visão

1. Introdução

Atualmente, é notável que com aumento da necessidade de sistema numa velocidade exponencial haja uma grande complexidade de softwares, dificuldade nos gerenciamentos de códigos, e muitas das vezes com falhas de segurança, causando assim problemas seríssimos a empresas e usuáries do sistema como vazamento de dados, ataques de hacker a serviços essenciais às empresas com a observação desses fenômenos se fez necessidade de api para facilitação, do controle de seguranças da aplicação separando o front end do back end, melhorando a assim a manutenção do código e tornado o desenvolvimento mais rápido como um todo e menos falhas de sistema.

2. Declaração do problema

O problema de afeta	<i>Necessidade de Api para gerenciar as transações.</i> Todo o gerenciamento da farmácia .
cujo impacto é	Sem um sistema, a velocidade de atendimento ao cliente fica mais lenta, visto que, o funcionário terá que fazer o cadastro manualmente. Não obstante, também há a dificuldade de gerenciar a farmácia, já que fazer o controle de estoque, medicamentos com desconto, transações entre contas ficará mais complexo.
uma solução de sucesso deveria	Criação de uma api para para gerenciar as transações do sistema

3. Contexto de negócio

O trabalho foi feito por divisões de funções, ficou definido o que cada membro iria fazer, mas todos eram obrigados a saber fazer o desenvolvimento, quanto as reuniões, eram realizadas no discord, não foi especificada um prazo extremamente restrito para o término definitivo de todo o projeto, apenas que seria para dia 16/06/2022.

4. Declaração da visão do software

Para	Para o balconista da farmácia.
Que	Diante do constante crescimento da farmácia houve a necessidade de criação de um sistema.
A	API
É um	Produto que gerenciará as requisições ao Banco de Dados

Que	O software tem o objetivo de dar o poder nas mãos do funcionário que gerencia toda a farmácia, ao invés de ter que fazer todo o processo por meio de arquivos físicos. Promovemos a melhora no atendimento, gerenciamento e melhor aproveitamento do tempo do funcionário.
Nosso produto	Propõe uma melhoria imediata em toda a farmácia de forma simples e rápida.

5. Visão geral do produto

Produto que busca reduzir tempo e dificuldade de gerenciamento, facilitar o controle da farmácia .

5.1 Necessidades e funcionalidades

Necessidade	Funcionalidade	Prioridade	Responsável
Modelagem do banco de dados	Tabelas: 1. Sales 2. Product 3. Order 4. Customer	1-Alta 2-Alta 3-Alta 4-Alta	Tobias Gustavo

Necessidade	Funcionalidade	Prioridade	Responsável
Permitir cadastro de novos clientes, incluindo dados pessoais, dados para contato e controle de estoque.	1. Cadastro de clientes 2. Listagem de clientes 3. Atualização de clientes 4. Deletar clientes 5. Histórico de transações entre contas 6. Controle de estoque 7. Medicamentos genéricos contêm 20% de desconto.	1-Alta 3-Alta 4-Alta 5-Alta 6-Alta 7-Alta	Rosemberg Vieira

Necessidade	Funcionalidade	Prioridade	Responsável
Gerais	1. Liste os endpoints no README.md. 2. Testes automatizados. 3. Documentação da Aplicação.	1-Alta 2-Alta 3-Alta	Lucas Pontes

5.2 Características tecnológicas da solução

Utilizamos a linguagem Java JDK-17 e o framework Spring, em conjunto com a IDE IntelliJ, para a construção da aplicação.

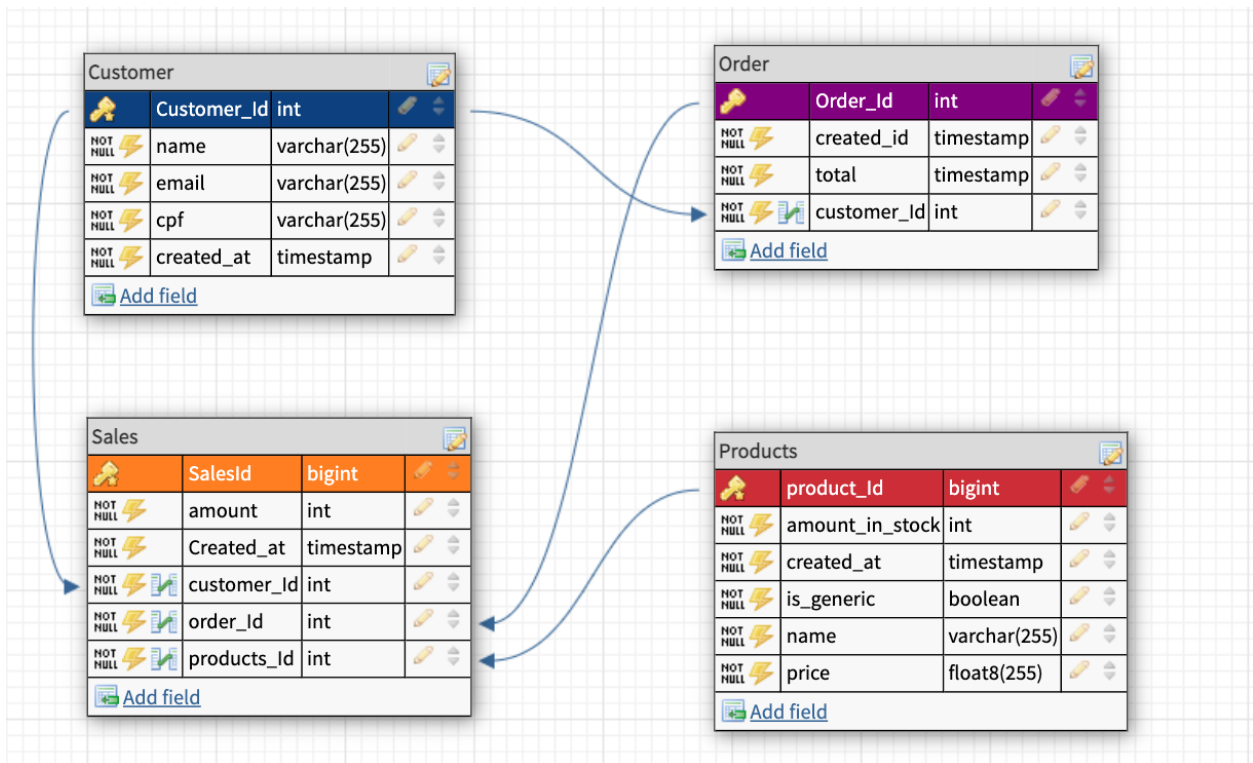
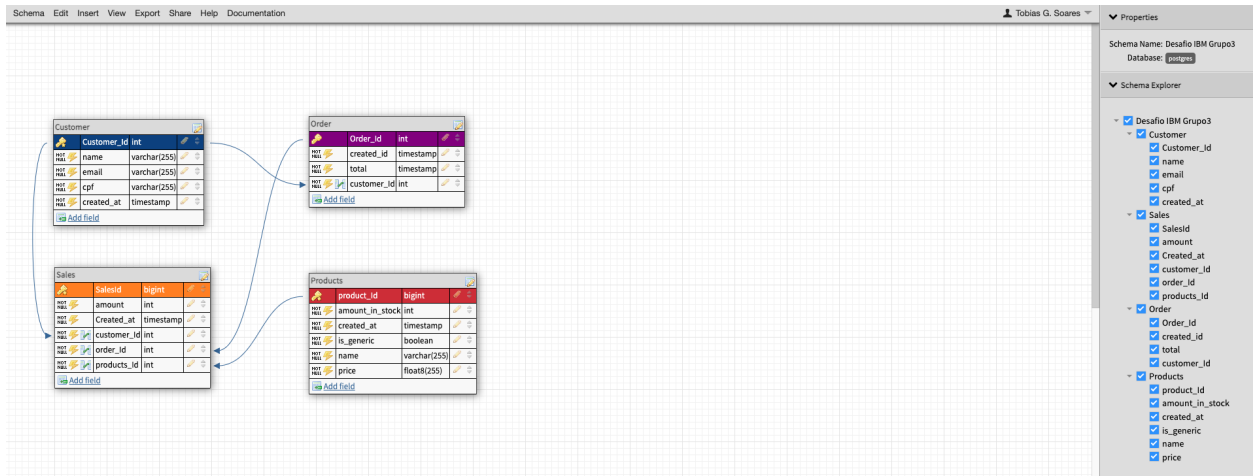
Postman para a execução dos métodos HTTP e suas requisições.

DbDesigner em conjunto ao Beekeeper para a modelagem dos modelos conceitual e lógico do banco de dados.

Notion para a realização do Kanban.

Git e GitHub para o versionamento.

T3. Modelo de Banco De Dados - DER / MER



Acesso: <https://dbdesigner.page.link/Q77bBSpVfyvL678K7>

TABLES

COLUMNS

CONSTRAINTS

RELATIONSHIPS

SIGN-IN

Schema Tables

Generated on Wednesday, 17 August 2022

Database Type: POSTGRES

4 TABLES

21 COLUMNS

4 CONSTRAINTS

Search:

Table	Children	Parents	Columns	Constraints
Customer	2	0	5	0
Order	1	1	4	1
Products	1	0	6	0
Sales	0	3	6	3

Table: Customer

Generated on Wednesday, 17 August 2022

Search:

Column	Type	Size	Nulls	Auto	Default	Children	Parents
cpf	varchar	255		✓	null		
created_at	timestamp			✓	null		
Customer_Id	int			✓	null	Sales Order	
email	varchar	255		✓	null		
name	varchar	255		✓	null		

Acesso:

https://sqlspy.io/import_db_designer/c3B5LTM0ODM2MjQtMjA5ODgwNjQ2M2IzMWExMC01NDc5ODM=

Beekeeper - Open Source

Entities

ENTITIES 188		+
	name	varchar(255)
▼	orders	
	order_id	int4
	created_at	timestamp(6)
	total	float8
	customer_id	int4
▼	products	
	product_id	int4
	amount_in_stock	int4
	created_at	timestamp(6)
	is_generic	bool
	name	varchar(255)
	price	float8
▼	sales	
	sale_id	int4
	amount	int4
	created_at	timestamp(6)
	customer_id	int4
	order_id	int4
	products_id	int4
▼	information_schema	
>	sql_features	
>	sql_implementation_info	
>	sql_sizing	
>	administrable_role_authorizations	
>	applicable_roles	
ibm		postgresql

TABLES: customers, orders, products, sales.

<>	Query #1	customers [all]	orders [all]	columns [all]	products [all]	sales [all]	
<>	customer_id	equals	Enter Value				
	customer_id int4	cpf varchar	created_at timestamp	email varchar	name varchar		
1		999999999292	2022-08-15 01:33:07.193011	Lucas@gmail.com	lucas Ponte		
2		999999999292	2022-08-15 01:33:11.46942	tobias@gmail.com	tobias soares		
3		999999999292	2022-08-15 13:43:01.296926	email@email.com	Cliente01		
4		999999999292	2022-08-15 14:55:55.13099	email@email.com	Cliente03		

<>	Query #1	customers [all]	orders [all]	columns [all]	products [all]	sales [all]	+
<>	order_id	equals	Enter Value				
	order_id int4	created_at timestamp	total float8	customer_id int4			
1		2022-08-15 01:33:32.921577	15	1			
2		2022-08-15 01:33:39.383578	15	1			
3		2022-08-15 01:50:25.783456	15	2			

<>	Query #1	customers [all]	orders [all]	products [all]	sales [all]	columns [all]	+
<>	product_id	equals	Enter Value				
	product_id int4	amount_in_stock int4	created_at timestamp	is_generic bool	name varchar	price float8	
1		97	2022-08-15 01:33:18.223176	false	p1	5	
2		94	2022-08-15 01:33:21.100298	false	p2	5	
3		91	2022-08-15 01:33:24.013007	false	p3	5	

<>	Query #1	customers [all]	orders [all]	products [all]	sales [all]	columns [all]	+
<>	sale_id	equals	Enter Value				
	sale_id int4	amount int4	created_at timestamp	customer_id int4	order_id int4	products_id int4	
1		1	2022-08-15 01:33:32.921577	1	1	1	
2		2	2022-08-15 01:33:32.921577	1	1	2	
3		3	2022-08-15 01:33:32.921577	1	1	3	