

Março - 2021

INF01145 - Fundamentos de Banco de Dados

Professora: Renata de Matos Galante

Trabalho Final - FASE I

Eduardo Eugênio Kussler (00315799)

Gabriel Couto Domingues (00302229)

1 Universo de Discurso

O universo de discurso escolhido foi um sistema de compra de jogos e conteúdo adicional baseado na plataforma Steam (<https://store.steampowered.com/>). O sistema tem como objetivo gerenciar os produtos da loja, o conjunto de usuários, e as compras de produtos. Não foi desenvolvida toda a plataforma, decidimos focar apenas na parte de compras na loja. O site, entretanto, tem uma Seção para comunidade e um mercado paralelo para itens de jogos. Algumas outras exclusões notáveis do sistema modelado em relação a Steam foram a lista de desejos, mais opções de pagamento e a opção de fazer compras para presente.

1.1 Produto

Cada produto na loja apresenta um nome, uma descrição opcional, preço original, e caso o produto esteja com desconto, a porcentagem do desconto, o preço com desconto e a data/hora de fim do desconto. É permitido que dois produtos tenham o mesmo nome.



Figura 1: Produto na loja com desconto

SOBRE ESTE JOGO

The Witcher: Wild Hunt é um RPG de mundo aberto de fantasia cheio de escolhas vitais. Em The Witcher, você joga como um caçador de monstros profissional, Geralt de Rívia, em busca da criança da profecia em um vasto mundo aberto, rico em cidades mercantis, ilhas piratas, passagens perigosas nas montanhas e cavernas esquecidas a serem exploradas.

Figura 2: Descrição de um produto

1.2 App

Na loja temos dois tipos de produtos, Apps e Pacotes, que são conjuntos de Apps. Um App representa um produto particular na loja, possuindo, além de todas informações de um produto, uma data de lançamento.



Figura 3: App na Loja

Um App também possui diversas formas de caracterização que estão pré cadastradas no sistema, como categorias, gêneros, empresa desenvolvedora e empresa distribuidora. É permitido que duas empresas tenham o mesmo nome no sistema. Um app só pode ter uma distribuidora e uma desenvolvedora mas pode ter vários gêneros e varias categorias.

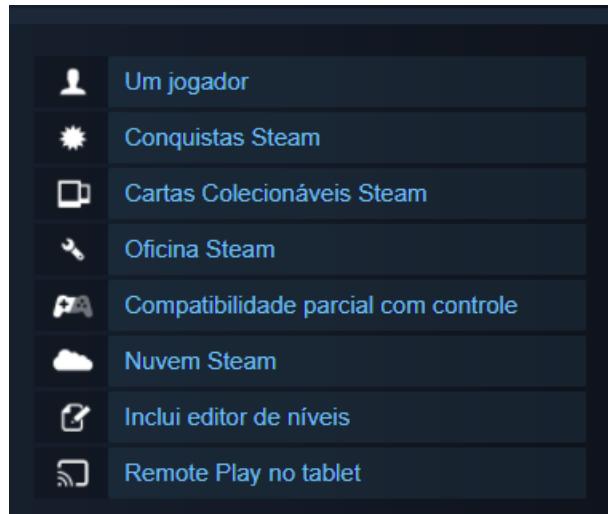


Figura 4: Categorias de um App

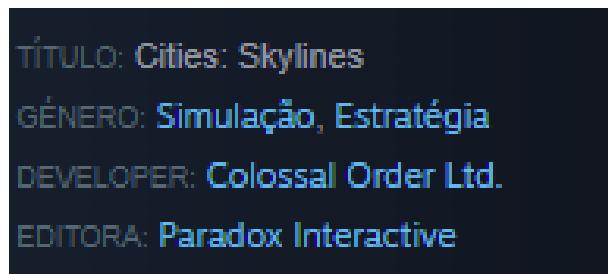


Figura 5: Generos e empresas de um App

Um App também pode receber tags (marcadores) de usuários, podendo receber mais de um marcador por usuário, sendo que essas não precisam estar pré cadastradas. As tags mais populares de um App aparecem na página principal da loja.

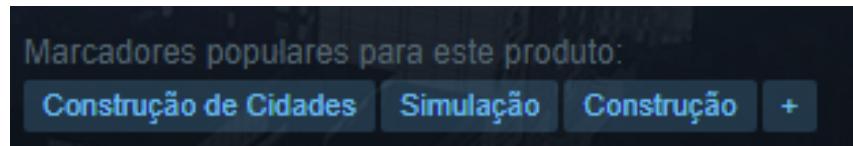


Figura 6: Tags de um App

Um app também recebe avaliações de usuários em que um usuário recomenda ou não o App, e opcionalmente deixa um comentário. Na página do App é apresentada a porcentagem de avaliações positivas e a quantidade total de avaliações.



Figura 7: Análises de Usuários de um App

1.3 Pacote

Um pacote é apenas um conjunto de Apps, portanto contém todas informações de um produto na loja e une informações sobre todos Apps que fazem parte do pacote como gênero, categorias, desenvolvedores e distribuidoras. Um App pode pertencer a mais de um Pacote. Vale observar que o preço do pacote pode ser diferente da soma dos preços individuais de cada item, mostrando a diferença de preço caso isso ocorra.



Figura 8: Pacote na Loja

1.4 Jogo e DLC

Apps estão separados em dois tipos, Jogos e Dlcs (conteúdo adicional) que pertencem aos Jogos. Um jogo na loja deve apresentar as DLCs que estão disponíveis para este Jogo.



Figura 9: Jogo na Loja

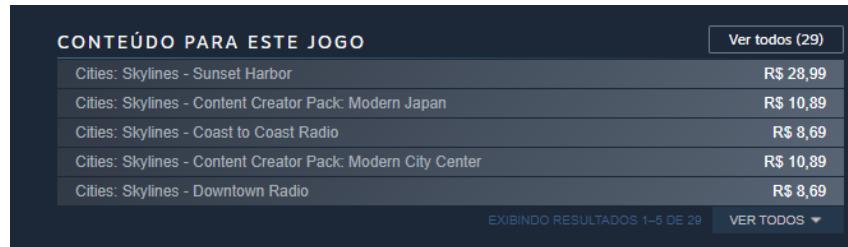


Figura 10: DLCs de um Jogo



Figura 11: DLC na Loja

1.5 Usuário

Um usuário que é cadastrado no sistema possui um id (único), nome e email (único). Cada usuário possui um conjunto de Produtos que comprou na loja, e para essas compras o usuário utiliza um dos cartões de crédito adicionados em sua conta, sendo possível que haja dois usuários com o mesmo cartão de crédito adicionado em suas contas.

1.6 Carrinho

O carrinho de um usuário armazena todos produtos que ele adicionou para comprar posteriormente. No carrinho pode-se ver informações do total a ser pago, o preço individual de cada item, o nome dos itens, e os itens inclusos em pacotes no carrinho.



Figura 12: Carrinho de um usuário

1.7 Compra

Para um usuário fazer uma compra no sistema, deve primeiro adicionar os itens que deseja comprar em seu carrinho e após isso utilizar um dos cartões de crédito adicionados em sua conta para efetuar o pagamento, sendo possível verificar se o pagamento já foi aprovado ou não. Após a compra, os produtos são adicionados à conta do usuário e é salvo o preço dos produtos comprados, a porcentagem de desconto dos produtos, a data da compra, o total e a forma de pagamento.

2 Modelo Conceitual

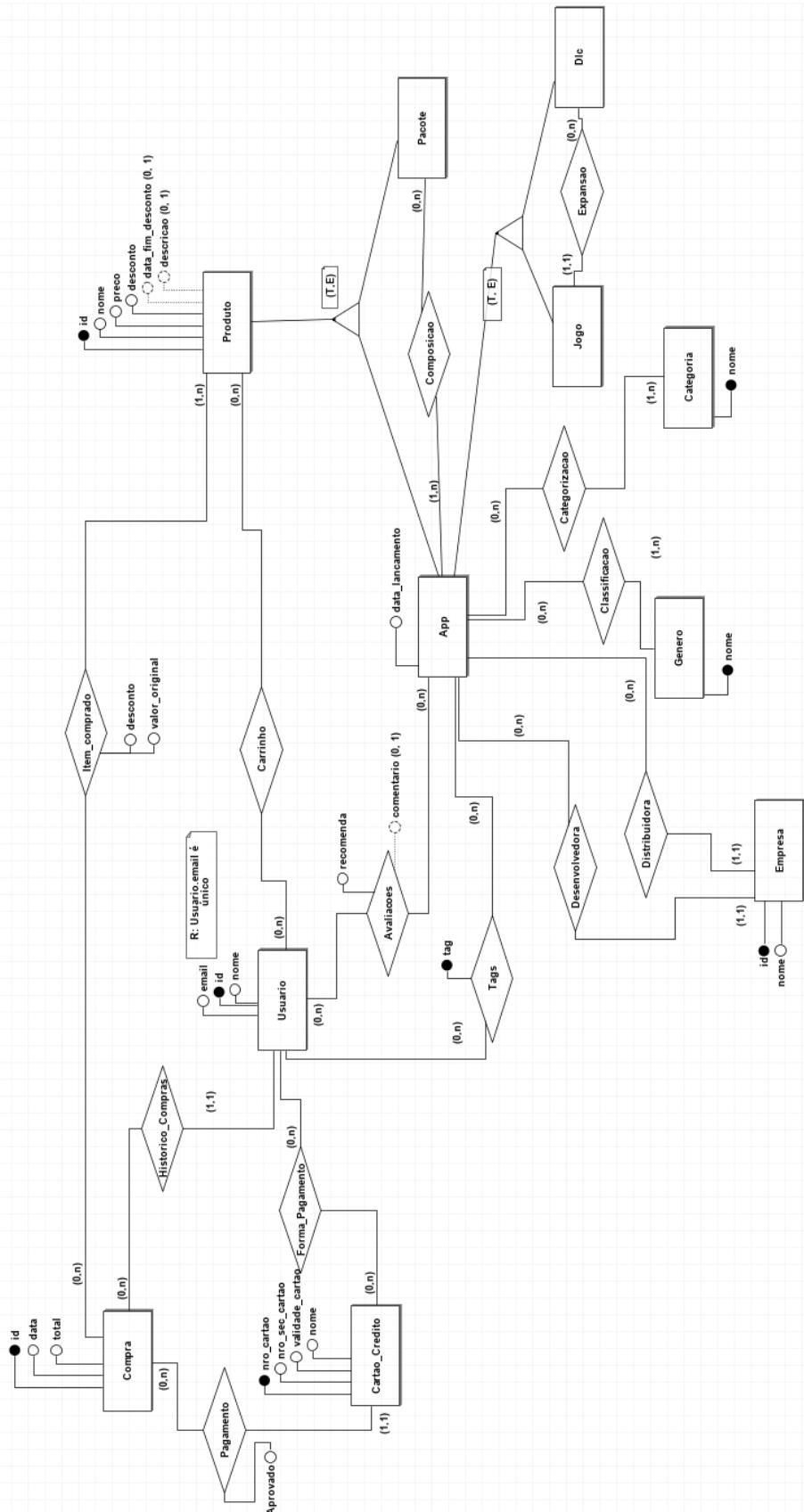


Figura 13: Modelo Conceitual

3 Dicionário de Dados

Abaixo está definido o Dicionário de Dados. Chaves candidatas foram definidas em **negrito** e atributos opcionais foram definidos com um *. As entidades são representadas em **cinza** e os relacionamentos são representados em **azul**.

Produto			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
id	SERIAL	Identificador do produto	004
nome	VARCHAR(50)	Nome do produto	Terraria
preco	DECIMAL(6,2)	Preço do produto	19.99
desconto	SMALLINT	Desconto do produto(limitado de 0 a 100)	0
data_fim_desconto*	TIMESTAMP	Data/hora do fim do desconto	2021-01-08 04:05:06
descricao*	VARCHAR(1000)	Descrição do produto	Dig, fight, explore, build! Nothing is impossible in this action-packed adventure game. Four Pack also available!

A entidade Produto representa um produto na loja. Este produto pode ser adicionado ao carrinho e comprado por um usuário.

App			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
data_lancamento	TIMESTAMP	Data/hora de lançamento do App	2011-05-16 04:05:06

A entidade App é uma especialização de produto, indica que o produto não é um pacote e sim um produto individual na loja.

Categoria			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
nome	VARCHAR(50)	nome da categoria	Um Jogador

A entidade Categoria representa as diferentes categorias disponíveis para um App.

Categorizacao

O relacionamento Categorizacao representa a relação entre uma Categoria e um App, ou seja representa que um determinado App é daquela categoria.

Genero			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
nome	VARCHAR(50)	nome do genero	Action

A entidade Genero representa os diferentes generos disponíveis para um App.

Classificacao

O relacionamento Classificacao representa a relação entre um Genero e um App, ou seja representa que um determinado App é daquele genero.

Empresa			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
id	SERIAL	identificador da empresa. Criado pois duas empresas podem ter o mesmo nome na plataforma	004
nome	VARCHAR(50)	Nome da empresa	EA

A entidade Empresa representa as diferentes Empresas cadastradas no sistema. As empresas podem estar relacionadas com um App sendo desenvolvedores ou distribuidoras desse App.

Desenvolvedora

O relacionamento Desenvolvedora representa uma das relações possíveis entre uma Empresa e um App, relacionando uma empresa com um jogo que desenvolveu.

Distribuidora

O relacionamento Distribuidora representa uma das relações possíveis entre uma Empresa e um App, relacionando uma empresa com um jogo que distribuiu.

Tags			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
tag	varchar(50)	tag dada ao produto	Fantasia

O relacionamento Tags representa uma tag que um Usuario pode dar a um App.

Avaliacoes			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
recomenda	BOOLEAN	Indica se o usuário recomenda (true) o jogo ou não (false)	true
comentario*	VARCHAR(1000)	Comentário da Avaliação	Makes me feel real smart and then real dumb and then real smart again. 10/10 would recommend

O relacionamento Avaliações representa uma avaliação que um Usuario pode dar a um App.

Pacote

A entidade Pacote é uma especialização de Produto que representa um conjunto de Apps na loja.

Composicao

O relacionamento Composição representa a relação entre um Pacote e um App, indicando que aquele App pertence a esse Pacote.

Jogo

A entidade Jogo é uma especialização de App e representa um jogo na Loja.

Dlc

A entidade Dlc é uma especialização de App e representa uma Dlc na Loja. Uma Dlc está associada a um jogo.

Expansao

O relacionamento Expansão representa a relação entre uma Dlc e um Jogo, indicando que aquela Dlc é daquele jogo.

Usuario			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
id	SERIAL	id do usuario	004
nome	VARCHAR(50)	nome do usuario	O verdadeiro
email	VARCHAR(100)	email do usuario	umemailgenerico@gmail.com

A entidade Usuario representa um usuário do sistema. Um usuário pode comprar os produtos disponíveis na loja e verificar seu histórico de compras e seus Produtos.

Cartao_Credito			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
nro_cartao	CHAR(19)	Número do cartão	1234 1234 1234 1234
nro_sec_cartao	CHAR(3)	Número de segurança do cartão	248
validade_cartao	DATE	Data de validade do cartão	2021-10-10
nome	VARCHAR(100)	Nome escrito no cartão	José da Silva

A entidade Cartao_credito representa um cartão de crédito que é armazenado na conta de um Usuário e usado para comprar Produtos na loja.

Forma_Pagamento

O relacionamento Forma_Pagamento representa a relação entre um cartão e um usuário, indicando que esse usuário tem esse cartão como forma de pagamento em sua conta.

Carrinho

O relacionamento Carrinho é uma relação entre um Produto na loja e um Usuario, indicando que o usuário tem aquele produto adicionado em seu carrinho, sendo assim, representa os produtos que estão no carrinho de um Usuario.

Compra			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
id	SERIAL	identificador da compra no sistema	004
data	TIMESTAMP	data e hora da compra	1999-01-08 04:05:06
total	DECIMAL(10,2)	valor total da compra. Este valor poderia ser calculado dos itens comprados, mas como ele não muda, ele será armazenado	10.95

A entidade Compra representa uma compra feita na loja por um Usuario.

Item_comprado			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
desconto	SMALLINT	Desconto no preço em porcentagem. Valor deve ser limitado de 0 a 100	20
valor_original	DECIMAL(6,2)	Valor original do produto na compra sem o desconto	248

O relacionamento Item_comprado é uma relação entre um Produto na loja e um Usuario, representando um produto comprado na loja por um Usuário. As informações do preço (valor e desconto) são armazenadas pois um produto na loja pode ter seu valor e desconto alterado após sua compra por um Usuário.

Historico_Compras

O relacionamento Historico_Compras representa a relação entre um Usuario e uma Compra, indicando que uma compra foi feita por um determinado usuário.

Pagamento			
ATRIBUTO	TIPO	DESCRIÇÃO	EXEMPLO
Aprovado	BOOLEAN	Indica se o pagamento foi aprovado	true

O relacionamento Pagamento representa a relação entre uma Compra de um Usuario e o Cartão utilizado pelo usuário para efetuar a Compra, apresentando a informação de confirmação do pagamento.

4 Descrição do Mapeamento

Para o mapeamento consideramos entidades mapeadas diretamente como entidades que viraram tabelas com o mesmo nome e consideramos atributos mapeados diretamente como atributos que viraram atributos de sua respectiva tabela.

- Entidade **Produto** mapeada diretamente
 - Atributo id mapeado diretamente, obrigatório e utilizado como chave primaria pois é o único atributo identificador de um produto no sistema,
 - Atributo nome mapeado diretamente e obrigatório,
 - Atributo preco mapeado diretamente e obrigatório,
 - Atributo desconto mapeado diretamente e obrigatório com uma verificação se o valor está entre [0,100],
 - Atributo data_fim_desconto mapeado diretamente e opcional,
 - Atributo descricao mapeado diretamente e opcional
 - Escolheu-se ter uma tabela para Produto e para suas especializações (App e Pacote) pois cada uma das especializações tem relacionamentos específicos e a entidade Produto tem dois relacionamentos que serão muito utilizados no sistema.
- Entidade **App** mapeada diretamente
 - Atributo id criado como chave estrangeira obrigatória para Produto(id), e como chave primaria pois todo App é um produto e Produto(id) é unico,
 - Atributo data_lancamento mapeado diretamente e obrigatório,
 - Atributo fk_Empresa_id_desenvolvedora do Relacionamento **Desenvolvedora** mapeado para chave estrangeira obrigatória referenciando Empresa(id) pois é um relacionamento 1 para N,
 - Atributo fk_Empresa_id_distribuidora do Relacionamento **Distribuidora** mapeado para chave estrangeira obrigatória referenciando Empresa(id) pois é um relacionamento 1 para N
 - Escolheu-se ter uma tabela para App e para suas especializações (Jogo e Dlc) pois as especializações tem cardinalidades diferentes no relacionamento Expansao e a entidade App tem varios relacionamentos no sistema.
- Entidade **Categoria** mapeada diretamente
 - Atributo id adicionado como chave primaria do tipo SERIAL. Este atributo foi criado pois o único outro atributo que poderia ser identificador, nome, é do tipo VARCHAR.

- Atributo nome mapeado diretamente, obrigatório e único,
- Relacionamento **Categorizacao** mapeado para tabela própria, por ser relacionamento N para N
 - Atributos fk_Categoria_id e fk_App_id como chave primaria composta pois a tabela representa um relacionamento entre uma Categoria e um App,
 - Atributo fk_Categoria_id como chave estrangeira obrigatória para Categoria(id),
 - Atributo fk_App_id como chave estrangeira obrigatória para App(id)
- Entidade **Genero** mapeada diretamente
 - Atributo id adicionado como chave primaria do tipo SERIAL. Este atributo foi criado pois o único outro atributo que poderia ser identificador, nome, é do tipo VARCHAR.
 - Atributo nome mapeado diretamente, obrigatório e único
- Relacionamento **Classificacao** mapeado para tabela própria, por ser relacionamento N para N
 - Atributos fk_Genero_id e fk_App_id como chave primaria composta pois a tabela representa um relacionamento entre um Genero e um App,
 - Atributo fk_Genero_id como chave estrangeira obrigatória para Genero(id),
 - Atributo fk_App_id como chave estrangeira obrigatória para App(id)
- Entidade **Empresa** mapeada diretamente
 - Atributo id mapeado diretamente, obrigatório e utilizado como chave primaria pois é o único atributo identificador de uma Empresa no sistema
 - Atributo nome mapeado diretamente e obrigatório
- Relacionamento **Tags** mapeado para tabela própria, por ser relacionamento N para N
 - Atributos fk_Usuario_id, fk_App_id e tag como chave primaria composta pois a tabela representa um relacionamento entre um Usuario e um App e um Usuario pode dar varias tags diferentes para um mesmo App,
 - Atributo fk_App_id como chave estrangeira obrigatória para App(id),
 - Atributo fk_Usuario_id como chave estrangeira obrigatória para Usuario(id),
 - Atributo tag mapeado diretamente e obrigatório
- Relacionamento **Avaliacoes** mapeado para tabela própria, por ser relacionamento N para N
 - Atributos fk_App_id e fk_Usuario_id como chave primaria composta pois a tabela representa um relacionamento entre um Usuario e um App,
 - Atributo fk_App_id como chave estrangeira obrigatória para App(id),
 - Atributo fk_Usuario_id como chave estrangeira obrigatória para Usuario(id),
 - Atributo recomenda mapeado diretamente e obrigatório,

- Atributo comentário mapeado diretamente e opcional
- Entidade **Pacote** mapeada diretamente
 - Atributo id como chave estrangeira obrigatória para Produto(id) e chave primaria pois todo Pacote é um produto e Produto(id) é único
- Relacionamento **Composicao** mapeado para tabela própria, por ser relacionamento N para N
 - Atributos fk_App_id e fk_Pacote_id como chave primaria composta pois a tabela representa um relacionamento entre um Pacote e um App,
 - Atributo fk_App_id como chave estrangeira obrigatória para App(id),
 - Atributo fk_Pacote_id como chave estrangeira obrigatória para Pacote(id)
- Entidade **Jogo** mapeada diretamente
 - Atributo id como chave estrangeira obrigatória para App(id) e chave primaria pois todo Jogo é um App e App(id) é único
- Entidade **Dlc** mapeada diretamente
 - Atributo id como chave estrangeira obrigatória para App(id) e chave primaria pois toda Dlc é um App e App(id) é único
 - Atributo fk_Jogo_id do Relacionamento **Expansao** mapeado para chave estrangeira obrigatória referenciando Jogo(id) pois é um relacionamento 1 para N
- Entidade **Usuario** mapeada diretamente
 - Atributo id mapeado diretamente, obrigatório e utilizado como chave primaria pois o único outro atributo que era único para um Usuario, email, era do tipo VARCHAR,
 - Atributo nome mapeado diretamente e obrigatório,
 - Atributo email mapeado diretamente, obrigatório e único
- Entidade **Cartao_Credito** mapeada diretamente
 - Atributo id adicionado como chave primaria do tipo SERIAL. Este atributo foi criado pois o único outro atributo que era único para um cartão, nro_cartao, era do tipo CHAR,
 - Atributo nro_cartao mapeado diretamente, obrigatório e único,
 - Atributo nro_sec_cartao mapeado diretamente e obrigatório,
 - Atributo validade_cartao mapeado diretamente e obrigatório,
 - Atributo nome mapeado diretamente e obrigatório
- Relacionamento **Forma_Pagamento** mapeado para tabela própria, por ser relacionamento N para N
 - Atributos fk_Usuario_id e fk_Cartao_Credito_id como chave primaria composta pois a tabela representa um relacionamento entre um Usuario e um Cartao_Credito,

- Atributo fk_Cartao_Credito_id como chave estrangeira obrigatória para Cartao_Credito(id),
- Atributo fk_Usuario_id como chave estrangeira obrigatória para Usuario(id)
- Relacionamento **Carrinho** mapeado para tabela própria, por ser relacionamento N para N
 - Atributos fk_Usuario_id e fk_Produto_id como chave primaria composta pois a tabela representa um relacionamento entre um Usuario e um Produto
 - Atributo fk_Produto_id como chave estrangeira obrigatória para Produto(id),
 - Atributo fk_Usuario_id como chave estrangeira obrigatória para Usuario(id)
- Entidade **Compra** mapeada diretamente
 - Atributo id mapeado diretamente, obrigatório e utilizado como chave primaria pois é o único atributo identificador de uma Compra no sistema,
 - Atributo data mapeado diretamente e obrigatório,
 - Atributo total mapeado diretamente e obrigatório,
 - Atributo fk_Usuario_id do relacionamento **Historico_Compras** mapeado para chave estrangeira obrigatória referenciando Usuario(id) pois é um relacionamento 1 para N,
 - Atributo fk_Cartao_Credito_id do relacionamento **Pagamento** mapeado para chave estrangeira obrigatória referenciando Cartao_credito(id) pois é um relacionamento 1 para N,
 - Atributo Aprovado do relacionamento **Pagamento** mapeado diretamente e obrigatório
- Relacionamento **Item_comprado** mapeado para tabela própria, por ser relacionamento N para N
 - Atributos fk_Compra_id e fk_Produto_id utilizados como chave primaria composta pois a tabela representa um relacionamento entre um Produto e uma Compra,
 - Atributo fk_Produto_id como chave estrangeira obrigatória para Produto(id),
 - Atributo fk_Compra_id como chave estrangeira obrigatória para Compra(id),
 - Atributo desconto mapeado diretamente, obrigatório e limitado de 0 a 100,
 - Atributo valor_original mapeado diretamente e obrigatório