

Enunciado do trabalho

O trabalho prático da disciplina deve versar sobre o projeto e uso de uma base de dados para um sistema de informação (SI), a ser modelado e implantado em computador. O trabalho envolve a modelagem e o projeto da base de dados com o uso de ferramentas de modelagem, bem como criação, instanciação e manipulação em um modelo relacional.

Esse relatório se refere à primeira parte do trabalho, dizendo respeito à implementação de um projeto conceitual da Base de Dados e sua subsequente implementação em SGBD relacional.

Universo de Discurso

O universo de discurso é baseado no site/aplicativo Google Play[1], que é um serviço de distribuição digital de aplicativos, jogos, filmes, programas de televisão, músicas e livros, desenvolvido e operado pela Google[2]. Ela é a loja oficial de aplicativos para o sistema operacional Android, além de fornecer conteúdo digital.

O universo é composto por:

- **Usuário:** Definido por nome, senha, cpf, data de nascimento e email (único). O usuário é responsável pela maior parte das ações no aplicativo, sendo elas:
 - **Comprar Itens** (livros, álbuns de música, aplicativos e filmes): Pode comprar quantos e quaisquer itens que quiser, porém apenas uma vez cada um deles. As compras só podem ser feitas por cartão e aceitas se o usuário tiver um cartão cadastrado e pagar o preço referente ao item. Os itens gratuitos são classificados como compras também, porém não necessitam que o usuário tenha cartões cadastrados. Depois de comprado o item, o download do mesmo é feito, e a informação de que o download foi feito é salva. Fazendo com que o item seja eternamente do usuário, ou seja, se ele quiser excluir do celular, terá a possibilidade de baixar novamente quando quiser, sem precisar comprar de novo.
 - **Adicionar Itens na Lista de Desejos:** Quando o usuário encontra um item que gosta e tem intenção de comprar no futuro, adiciona na sua lista de desejos, que o possibilita voltar a qualquer momento e achar o item de forma rápida. Podem ser adicionados quantos e quaisquer itens que desejar, porém apenas uma vez cada um deles.
 - **Revisão de Itens:** Usuários podem dar sua opinião sobre itens, tanto os que já foram usados na plataforma, quanto vistos em outros lugares. Cabe ao bom senso do usuário comentar apenas em itens que já usou. Cada usuário só pode fazer uma revisão por item, mas pode revisar quantos itens quiser. Uma revisão é composta por uma nota (de 0 a 5), uma data e opcionalmente um breve comentário.
 - **Adicionar cartões:** Cartões são usados para realizar compras. O usuário pode adicionar quantos desejar, desde que estejam no seu próprio nome. Porém não pode

adicionar novamente o mesmo cartão.

- Baixar Itens: O download é ligado com a compra, cada vez que é feita uma compra, um download é realizado. O mesmo guarda a data e a informação de que o download foi feito, uma forma de demonstrar que o usuário possui o item baixado, para que futuramente por mais que o exclua, continue tendo permissão para baixar novamente.

- **Cartão:** Definido por número (único) data de vencimento e usuário que é dono (único). Sendo responsável, juntamente do usuário, pela ação de compra de itens

- **Item:** Pode ser livro, aplicativo, álbum de música ou filme, é o objeto de consumo do usuário e base da plataforma.

Definido por preço (sendo o item considerado gratuito quando for 0 ou nulo, e por isso é opcional), nome, data de lançamento, resumo e código(único). Além disso o item pode entrar na promoção em determinadas épocas, sendo necessário guardar o valor dele com desconto.

O item é ligado em quase todas as ações disponíveis do usuário (menos adicionar cartões), sendo dividido em quatro partes:

- Aplicativo: Além de todos os atributos do item possui versão, tamanho e um desenvolvedor.

- * Desenvolvedor: Possui endereço, e-mail, senha e uma quantidade n de aplicativos criados. É um tipo de conta, feita para criadores de aplicativos, onde é considerado um desenvolvedor qualquer pessoa que criou uma conta deste tipo, ou seja, não é necessário ter um aplicativo para ser um desenvolvedor.

- Livro: Além de todos os atributos do item possui isbn (único), número de páginas, um autor e uma linguagem.

- * Autor: Como o autor pode ser objeto de filtro para busca de livros, é importante que seja padronizado. Dessa forma, sempre que um livro é cadastrado, existe a opção de escolher um autor já existente e caso não exista criar um novo. Para que isso seja possível, autores tem um nome único. E tem pelo menos um livro, até n .

- * Linguagem: Linguagens são pré-definidas, para que possam, além de controlar as linguagens de livros existentes, ajudar em buscas específicas. Como linguagens são definidas antes de livros, existem linguagens sem livros cadastrados com elas, mas as mesmas podem ter até n livros. Livros sempre tem apenas uma linguagem, sendo ela obrigatoriamente percentente a tabela 1 definida após o texto.

- Filme: Além de todos os atributos do item possui duração, linguagens e integrantes.

- * Linguagem: Possui o mesmo objetivo da linguagem do livro, com a única diferença de que quando se relaciona com o filme pode ser de dois tipos, áudio ou legenda. Pode existir várias linguagens de áudio e legenda para um mesmo filme, porém não podem se repetir no mesmo filme (ex: duas vezes áudio em português).

- * Integrantes: Compostas por nome e função, são pessoas que participaram da produção do filme, com certas funções. Eles são padronizados assim como os autores de livros.
- .
- Álbum: Além de todos os atributos do item possui faixas de música e artista.
 - * Faixa de Música: São as músicas que compõem o álbum, elas tem um nome único e uma duração. Cada música só está presente em um álbum, porém um álbum tem várias músicas.
 - * Artista: É o cantor ou grupo que criou o álbum. Tem um nome único e pode possuir quantos álbuns quiser, porém um álbum só tem um artista que o compôs. Eles são padronizados assim como os autores de livros.
- .
- Além desta divisão de tipos, os itens também tem categorias e anexos, que são:
 - Categoria: Nada mais que os gêneros de um item (ex: ação para filmes, pop para músicas...). Gêneros são pré-definidos e um item só pode ser cadastrado com gêneros que já existem. Além disso, itens só podem ter gêneros que fazem parte do seu tipo, por exemplo, um álbum não pode ter gênero ação.
Dado isso, um item pode ter quantos gêneros quiser, desde que não repita o mesmo, seja do seu tipo e esteja contido na tabela de gêneros (definida após o texto na tabela 2).
 - Anexos: Itens podem ter fotos e vídeos anexados para demonstrar seu conteúdo. Sendo necessário que para cada anexo adicionado, tenha caminho do arquivo (único), tipo de arquivo, nome e extensão.

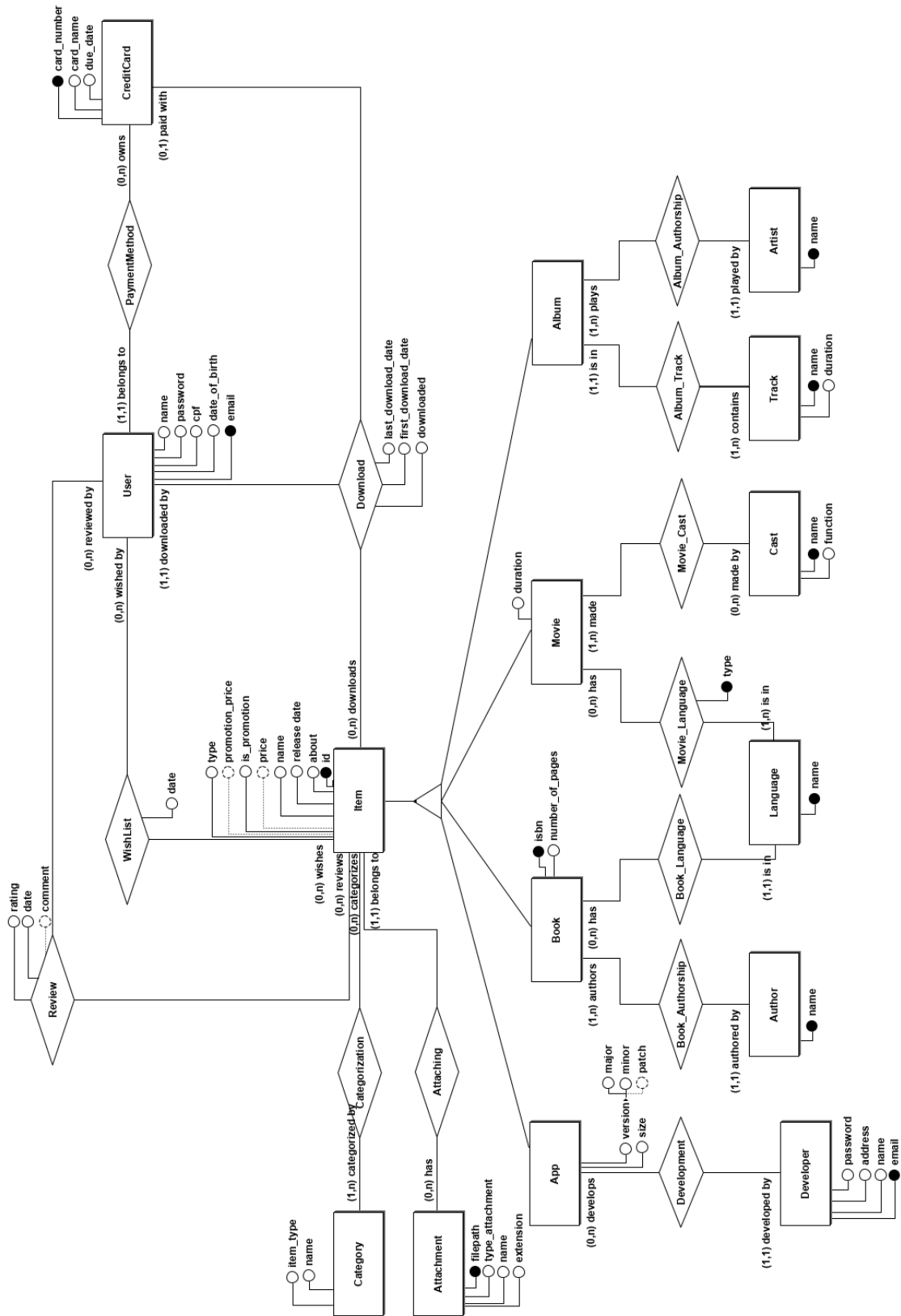
Linguagens			
Português	Inglês	Espanhol	Alemão
Italiano	Francês	Russo	Japones
Islândes	Norueguês	Sueco	Coreano

Tabela 1: Linguagens possíveis no sistema

Categorias			
<i>App</i>	<i>Filme</i>	<i>Livro</i>	<i>Álbum</i>
Arte e Design	Ação e Aventura	Arte	Alternativa
Carro e Veículos	Animação	Terror	Blues
Beleza	Comédia	Biografias	MPB
Esportes	Crime	Finanças	Infantil
Negócio	Documentário	Cozinha	Classica
Tempo	Drama	Educação	Country
Comunicação	Família	Ficção	Eletrônica
Namoro	Terror	História	Folk
Educação	Mistério	Humor	Rap
Entretenimento	Suspense	Psicologia	Jazz
Eventos	Ficção Científica	Religião	Metal
Família	Fantasia	Romance	Pop
Finanças	Esportes	Fantasia	Reggae
Social		Auto-ajuda	Rock
Viagens			Sertanejo
Saúde e Ginástica			Samba
Mapas e Navegação			Rock Nacional
Fotografia			Indie

Tabela 2: Categorias (com seus respectivos tipos) possíveis no sistema

Modelo Conceitual



Dicionário de Dados

O dicionário de dados será itemizado abaixo, com cada tabela representando uma entidade (ou relacionamento com atributos) presente na modelagem. Nele já transferimos o modelo conceitual para lógico, com as tabelas geradas e os atributos colocados nas posições escolhidas pelos integrantes do grupo.

Atributos em **negrito** definem que eles representam as chaves primárias das tabelas, sendo que se houver mais do que um atributo em **negrito**, esses atributos formam uma chave primária composta. Atributos únicos que não são definidos como chave primária (chaves candidatas) estão sublinhados. Atributos com um asterisco indicam que eles são opcionais. Todos os tipos descritos abaixo são referentes aos tipos que o PostgreSQL implementa em seu SGBD.

Item			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
id	uuid	Identificador único para o item	704ef290-...
price*	numeric(5,2)	Valor a ser cobrado por um item, em reais, sendo o valor limitado em R\$ 999,99. Caso esse campo esteja nulo, entendemos que o produto está de graça.	9.99
name	varchar(50)	Nome do item, com no máximo 50 caracteres	Super SGBD
release_date	date	Data de lançamento do item	31/12/2018
about	varchar(1000)	Descrição sobre o item, podendo conter até no máximo 1000 caracteres	Lorem ipsum...
is_promotion	boolean	Informa se o item está ou não em promoção. Se campo for true, o item é vendido pelo valor descrito em promotion_price, caso contrário, é vendido pelo valor descrito em price.	false
promotion_price*	numeric(5,2)	Valor a ser cobrado por um item, em reais, caso o atributo is_promotion seja true, sendo o valor limitado em R\$ 999,99. Caso esse campo esteja nulo, enquanto is_promotion esteja em true, entendemos que o produto está de graça.	
type	char(5)	Referencia qual tipo de Item esse Item realmente é. É utilizado para sabermos em qual outra tabela da especificação se encontram o resto dos dados (já que a generalização/especificação da tabela Item é total). Pode assumir os valores { app, movie, book, album }	app

Tabela 3: Tabela Item e a descrição textual de seus atributos

Attachment			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
filepath	varchar(100)	Caminho para onde o anexo está guardado localmente	./downloads/attach_001.jpg
app_id	uuid	Referencia o atributo id da tabela Item, para demonstrar a qual Item esse Attachment pertence	704ef290-...
type_attachment	char(6)	Tipo de anexo, sendo que os possíveis valores são {imagem, video}	imagem
name	varchar(50)	Nome dado ao anexo, para ser utilizado na hora de mostrarmos o mesmo na UI.	Imagem 01
extension	char(5)	Extensão dada para o arquivo	jpg

Tabela 4: Tabela Attachment e a descrição textual de seus atributos

User			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
email	varchar(50)	E-mail do usuário, válido de acordo com a RFC 5322[3]	example@example.com
name	varchar(80)	Nome do usuário, com no máximo 80 caracteres	John Doe
password	varchar	Senha do usuário, guardada com criptografia SHA512[4]	1253AB...
cpf	char(11)	CPF do usuário, armazenado sem os caracteres separadores	99999999999
birthdate	date	Data de nascimento do usuário	02/01/1998

Tabela 5: Tabela User e a descrição textual de seus atributos

Credit Card			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
card_number	char(16)	Identificador do cartão de crédito	1111 1111 1111 1111
user_email	varchar(50)	Referencia o atributo email da tabela User, para demonstrar qual User é o responsável por esse cartão	example@example.com
card_name	varchar(80)	Nome do dono do cartão, que pode ser diferente do User relacionado a esse Credit Card	John Doe
due_date	date	Data de vencimento do cartão	31/12/2025

Tabela 6: Tabela CreditCard e a descrição textual de seus atributos

WishList			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
user_email	varchar(50)	Referencia o atributo email da tabela User para demonstrar na WishList de qual User o Item foi adicionado	example@example.com
app_id	uuid	Referencia o atributo id da tabela Item, para demonstrar qual Item o User adicionou em sua WishList	704ef290-...
date	date	Data na qual o Item foi adicionado à WishList do User	12/01/2019

Tabela 7: Tabela WishList e a descrição textual de seus atributos

Review			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
user_email	varchar(50)	Referencia o atributo email da tabela User para sabermos qual User realizou essa Review	example@example.com
app_id	uuid	Referencia o atributo id da tabela Item, para sabermos qual Item foi avaliado	704ef290-...
rating	smallint	Nota atribuída pelo User ao Item. Seus valores possíveis são {0, 1, 2, 3, 4, 5}	4
date	date	Data que a review aconteceu	22/02/2019
comment*	varchar(256)	Comentário opcional do User referente ao Item	Top

Tabela 8: Tabela Review e a descrição textual de seus atributos

Download			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
item_id	uuid	Referencia o atributo id da tabela Item, para demonstrar à qual Item esse download se refere	704ef290-...
user_email	varchar(50)	Referencia o atributo email da tabela User, para demonstrar qual User realizou esse download	example@example.com
credit_card_number*	char(16)	Referencia o atributo card_number da tabela Credit Card, para demonstrar qual Credit Card o User utilizou. Esse campo só está preenchido se quando o User realizou o download do Item, ele possuía price ou promotion_price (caso em promoção) diferente de null ou 0.	1111 1111 1111 1111
first_download_date	date	Data na qual o item foi baixado pela primeira vez	18/01/2019
last_download_date	date	Data na qual o item foi baixado pela última vez. É sempre maior ou igual a first_download_date.	18/01/2019
downloaded	boolean	Informa se o item está instalado (true) ou desinstalado (false) no momento	false

Tabela 9: Tabela Download e a descrição textual de seus atributos

Language			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
name	varchar(20)	Nome da linguagem, com no máximo 20 caracteres	italiano

Tabela 10: Tabela Language e a descrição textual de seus atributos

Developer			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
email	varchar(50)	E-mail do(a) desenvolvedor(a), validado de acordo com a RFC 5322[3]	developer@example.com
name	varchar(80)	Nome do(a) desenvolvedor(a), com no máximo 80 caracteres	Developer Doe
address	varchar	Endereço do(a) desenvolvedor(a) contendo rua, número, bairro, cidade e país, ou conforme for o padrão do país de origem do(a) desenvolvedor(a)	..., Havana, Cuba
password	varchar	Senha do developer, guardada com criptografia SHA512[4]	8269bf0a...

Tabela 11: Tabela Developer e a descrição textual de seus atributos

Artist			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
name	varchar(80)	Representa o nome de um artista que pode possuir algum Album. Nome pode possuir no máximo 80 caracteres	Artist Doe

Tabela 12: Tabela Artist e a descrição textual de seus atributos

Author			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
name	varchar(80)	Nome do autor, com no máximo 80 caracteres	Author Doe

Tabela 13: Tabela Author e a descrição textual de seus atributos

Cast			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
name	varchar(80)	Nome da pessoa que pertence ao Cast de algum filme	Cast Doe
function	char(8)	Função que a pessoa desempenhou no filme. É um dos valores entre {actor, director, producer, writer}	writer

Tabela 14: Tabela Cast e a descrição textual de seus atributos

Category			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
name	varchar(25)	Nome da categoria, sendo único dentro do mesmo item_type	Ficção Científica
item_type	char(5)	Tipo de item à qual essa categoria se aplica. Os valores possíveis são {app, livro, filme, album}. Para o mesmo item_type, não podem haver 2 categorias com o mesmo nome. Isso significa que item_type e name juntos , identificam uma categoria.	filme

Tabela 15: Tabela Category e a descrição textual de seus atributos. Mais informações sobre ela na Tabela 2.

Categorization			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
category_name	varchar(25)	Referencia o atributo name da tabela Category, associando essa Categoria a um Item	Ficção Científica
category_item_type	char(5)	Referencia o atributo item_type da tabela Category, informando a qual tipo de Item essa categoria pode se relacionar. O Item referenciado por essa Categorization precisa ser do mesmo tipo que o descrito nesse campo.	filme
item_id	uuid	Referencia o atributo id da tabela Item, fazendo com que uma Category seja associada a um Item.	704ef290-...

Tabela 16: Tabela Categorization e a descrição textual de seus atributos

App (que especifica a tabela Item)			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
app_id	uuid	Referencia o atributo id da tabela Item, para encontramos os dados adicionais referentes ao App	704ef290-...
developer_email	varchar(50)	Referencia o atributo email da tabela Developer, para sabermos quem foi o(a) desenvolvedor(a) desse App	developer@example.com
size	bigint	Tamanho do aplicativo, em bytes	1048576
version	varchar	Texto demonstrando a versão do app no formato "MAJOR.MINOR.PATCH", sendo que MAJOR, MINOR e PATCH são números inteiros, sem vírgulas, e PATCH pode ser omitido juntamente com o ponto anterior à ele	2.1.12

Tabela 17: Tabela App (que especifica a tabela Item) e a descrição textual de seus atributos

Album (que especifica a tabela Item)			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
app_id	uuid	Referencia o atributo id da tabela Item, para encontramos os dados adicionais referentes ao Album	704ef290-...
artist_name	varchar(80)	Referencia o atributo name da tabela Arist, para encontramos o cantor/produtor desse album	Artist Doe

Tabela 18: Tabela Album (que especifica a tabela Item) e a descrição textual de seus atributos

Track			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
album_id	uuid	Referencia o atributo app_id da tabela Album, para sabermos de qual Album essa música pertence	704ef290-...
name	varchar(50)	Nome da música, sendo única dentro de um álbum. Possui no máximo 50 caracteres	Juntos e Shallow Now
duration	int	Duração da música em segundos	205

Tabela 19: Tabela Track e a descrição textual de seus atributos

Book (que especifica a tabela Item)			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
item_id	uuid	Referencia o atributo id da tabela Item, para encontramos os dados adicionais referentes ao Book	704ef290-...
author_name	varchar(80)	Referencia o atributo name da tabela Author, para sabermos quem foi o(a) autor(a) desse Book	Author Doe
language_name	varchar(80)	Referencia o atributo name da tabela Language, para sabermos em qual linguagem o livro está escrito	Italiano
number_of_pages	smallint	Quantidade de páginas no livro	425
ISBN	char(13)	ISBN (código identificador) do livro[5]	9788533302273

Tabela 20: Tabela Book (que especifica a tabela Item) e a descrição textual de seus atributos

Movie (que especifica a tabela Item)			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
app_id	uuid	Referencia o atributo id da tabela Item, para encontramos os dados adicionais referentes ao Movie	704ef290-...
duration	int	Quantidade de minutos do filme	123

Tabela 21: Tabela Movie (que especifica a tabela Item) e a descrição textual de seus atributos

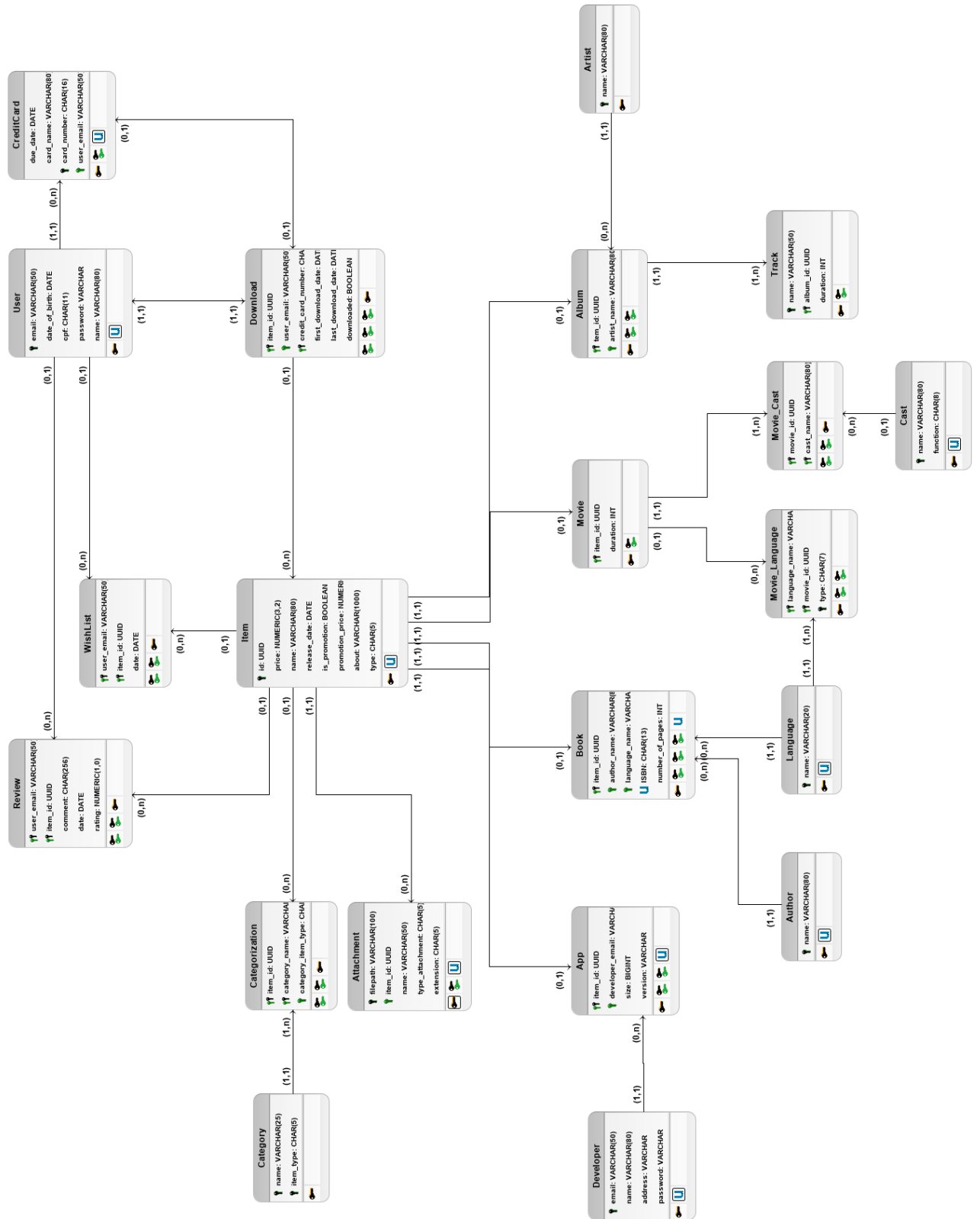
Movie_Language			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
movie_id	uuid	Referencia o atributo app_id da tabela Movie, para sabermos qual filme possui essa linguagem	704ef290-...
language_name	varchar(80)	Referencia o atributo name da tabela Language, para sabermos qual linguagem o filme indicado em movie_id possui	Italiano
type	char(7)	Representa se essa linguagem para esse filme é relacionada à legenda ou ao áudio. Os únicos valores permitidos são {audio, legenda}	legenda

Tabela 22: Tabela Movie_Language e a descrição textual de seus atributos

Movie_Cast			
<i>Atributo</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Exemplo</i>
movie_id	uuid	Referencia o atributo app_id da tabela Movie, para sabermos qual filme possui essa pessoa no Cast	704ef290-...
cast_name	varchar(80)	Referencia o atributo name da tabela Cast, para sabermos qual pessoa do cast participou do filme indicado em movie_id	Cast Doe

Tabela 23: Tabela Movie_Cast e a descrição textual de seus atributos

Modelo Lógico



Descrição do Mapeamento (Conceitual para Lógico)

Segue uma lista de mapeamentos implementados, onde os atributos, entidades e relacionamentos que estão escritos com *itálico*, são campos adicionais que não estavam no modelo conceitual e os que são definidos como "Mapeamento direto"querem dizer que estão da mesma maneira como foram definidos no conceitual.

Entidades:

- User: Mapeamento direto para **User**.
 - name: Mapeamento direto para **User.name**, obrigatório.
 - email: Mapeamento direto para **User.email**, obrigatório e chave primária.
 - date_of_birth: Mapeamento direto para **User.date_of_birth**, obrigatório.
 - password: Mapeamento direto para **User.password**, obrigatório.
- CreditCard: Mapeamento direto para **CreditCard**.
 - due_date: Mapeamento direto para **CreditCard.due_date**, obrigatório.
 - card_name: Mapeamento direto para **CreditCard.card_name**, obrigatório.
 - card_number: Mapeamento direto para **CreditCard.card_number**, obrigatório e chave primária.
 - user_email: Identificador do usuário, por conta do mapeamento de relacionamento por colunas adicionais no relacionamento PaymentMethod. Mapeado para **CreditCard.user_email**, obrigatório e chave estrangeira.
- Item: Como é uma generalização total e exclusiva, o mapeamento foi feito seguindo a alternativa 1 (segundo os slides de mapeamento). Onde foi usada como chave estrangeira (única) na entidade especializada a chave primária da entidade generalizada, fazendo com que sempre que ocorre uma entidade especializada, exista uma entidade generalizada que ainda não foi relacionada. Mapeada para **Item**.
 - id: É um índice criado para ser a chave primária do item, pois como não existe nenhum atributo sozinho que é único, e ele tem que ser usado como chave estrangeira na entidade especializada, foi decidido que seria lógico usar um índice, por conta do menor espaço ocupado e facilidade. Mapeado para **Item.id**, obrigatório e chave primária, sendo do tipo UUID, automaticamente gerado pelo banco de dados.
 - name: Mapeamento direto para **Item.name**, obrigatório.
 - price: Mapeamento direto para **Item.price**, opcional.
 - release_date: Mapeamento direto para **Item.release_date**, obrigatório.
 - is_promotion: Mapeamento direto para **Item.is_promotion**, obrigatório.
 - promotion_price: Mapeamento direto para **Item.promotion_price**, opcional.
 - about: Mapeamento direto para **Item.about**, obrigatório.
- App: Entidade especializada de item, onde contém como chave primária, uma chave estrangeira para item, fazendo com que seja obrigatória a existência do mesmo para sua criação. Mapeada para **App**.

- *item_id*: É uma chave estrangeira para Item, ao mesmo tempo em que é chave primária, fazendo com que seja único o relacionamento com Item e não exista nenhum outro App relacionado com o mesmo. Mapeamento direto para **App.item_id**, obrigatório e chave primária.
- *developer_email*: É uma chave estrangeira para Developer. Mapeado para **App.developer_email**, obrigatório.
- *size*: Mapeamento direto para **App.size**, obrigatório.
- *version*: Mapeamento direto para **App.version**, obrigatório.
- **Book**: Entidade especializada de item, onde contém como chave primária, uma chave estrangeira para item, fazendo com que seja obrigatória a existência do mesmo para sua criação. Mapeada para **Book**.
 - *item_id*: É uma chave estrangeira para Item, ao mesmo tempo em que é chave primária, fazendo com que seja único o relacionamento com Item e não exista nenhum outro App relacionado com o mesmo. Mapeamento direto para **Book.item_id**, obrigatório e chave primária.
 - *author_name*: Chave estrangeira para a tabela Author. Mapeado para **Book.author_name**, obrigatório.
 - *language_name*: Chave estrangeira para a tabela Language. Mapeado para **Book.language_name**, obrigatório.
 - *isbn*: Mapeamento direto para **Book.isbn**, obrigatório e único.
 - *number_of_pages*: Mapeamento direto para **Book.number_of_pages**, obrigatório.
- **Movie**: Entidade especializada de item, onde contém como chave primária, uma chave estrangeira para item, fazendo com que seja obrigatória a existência do mesmo para sua criação. Mapeada para **Movie**.
 - *item_id*: É uma chave estrangeira para Item, ao mesmo tempo em que é chave primária, fazendo com que seja único o relacionamento com Item e não exista nenhum outro App relacionado com o mesmo. Mapeamento direto para **Movie.item_id**, obrigatório e chave primária.
 - *duration*: Mapeamento direto para **Movie.duration**, obrigatório.
- **Album**: Entidade especializada de item, onde contém como chave primária, uma chave estrangeira para item, fazendo com que seja obrigatória a existência do mesmo para sua criação. Mapeada para **Album**.
 - *item_id*: É uma chave estrangeira para Item, ao mesmo tempo em que é chave primária, fazendo com que seja único o relacionamento com Item e não exista nenhum outro App relacionado com o mesmo. Mapeamento direto para **Album.item_id**, obrigatório e chave primária.
 - *artist_name*: É uma chave estrangeira para Album. Mapeado para **Album.artist_name**, obrigatório.
- **Developer**: Mapeamento direto para **Developer**.
 - *email*: Mapeamento direto para **Developer.email**, obrigatório e chave primária, pois é a única chave candidata.

- name: Mapeamento direto para **developer.name**, obrigatório.
- address: Mapeamento direto para **Developer.address**, obrigatório.
- password: Mapeamento direto para **Developer.password**, obrigatório.
- Author: Mapeamento direto para **Author**.
 - name: Mapeamento direto para **Author.name**, obrigatório e chave primária, pois é a única chave candidata.
- Language: Mapeamento direto para **Language**.
 - name: Mapeamento direto para **Language.name**, obrigatório e chave primária, pois é a única chave candidata.
- Cast: Mapeamento direto para **Cast**.
 - name: Mapeamento direto para **Cast.name**, obrigatório e chave primária, pois é a única chave candidata.
 - function: Mapeamento direto para **Cast.function**, obrigatório.
- Track: Mapeamento direto para **Track**.
 - name: Mapeamento direto para **Track.name**, obrigatório e chave primária juntamente com **Track.album_id**, já que **Track** é uma entidade fraca.
 - *album_id*: Mapeamento direto para **Track.album_id**, obrigatório e chave primária juntamente com **Track.name**, já que **Track** é uma entidade fraca em relação a **Album**.
 - duration: Mapeamento direto para **Track.duration**, obrigatório e não nulo.
- Attachment: Mapeamento direto para **Attachment**
 - filepath: Mapeamento direto para **Attachment.filepath**, obrigatório e não nulo, é a chave primária da tabela por ser a única chave candidata.
 - *item_id*: Mapeamento direto para **Attachment.item_id**, obrigatório e não nulo, é uma chave estrangeira para a tabela Item, representando o relacionamento Attaching.
 - name: Mapeamento direto para **Attachment.name**, obrigatório e não nulo.
 - type_attachment: Mapeamento direto para **Attachment.type_attachment**, obrigatório e não nulo.
 - extension: Mapeamento direto para **Attachment.extension**, obrigatório e não nulo.
- Category: Mapeamento direto para **Category**
 - name: Mapeamento direto para **Category.name**, obrigatório e não nulo, é unico juntamente com **Category.item_type**, assim como chave primária juntamente com ele.
 - item_type: Mapeamento direto para **Category.item_type**, obrigatório e não nulo, é unico juntamente com **Category.name**, assim como chave primária, juntamente com ele.

Relacionamentos:

- **PaymentMethod**: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1: N . Com a chave estrangeira em **CreditCard**.
- **Review**: Mapeamento de relacionamento por tabela própria, pois é um relacionamento $N:M$.
 - *user_email*: Chave primária do usuário, por ser uma tabela de relação que liga usuário a itens. Mapeado para **Review.user_email**, obrigatório e chave estrangeira.
 - *item_id*: Chave primária do item, pelo mesmo motivo do atributo *user_email*. Mapeado para **Review.item_id**, obrigatório e chave estrangeira.
 - *comment*: Mapeamento direto para **Review.comment**, opcional.
 - *date*: Mapeamento direto para **Review.date**, obrigatório.
 - *rating*: Mapeamento direto para **Review.rating**, obrigatório.
- **WishList**: Mapeamento de relacionamento por tabela própria, pois é um relacionamento $N:M$.
 - *user_email*: Chave primária do usuário, por ser uma tabela de relação que liga usuário a itens. Mapeado para **WishList.user_email**, obrigatório e chave estrangeira.
 - *item_id*: Chave primária do item, pelo mesmo motivo do atributo *user_email*. Mapeado para **WishList.item_id**, obrigatório e chave estrangeira.
 - *date*: Mapeamento direto para **WishList.date**, obrigatório.
- **Download**: Mapeamento de relacionamento por tabela própria, pois é um relacionamento ternário. Foi implementada uma tabela com referência (chave estrangeira) às chaves primárias de **User**, **CreditCard** e **Item**.
 - *item_id*: Chave primária do item, que está fazendo parte da chave composta que identifica o próprio **Download**. Mapeado para **Download.item_id**, obrigatório e chave estrangeira.
 - *user_email*: Chave primária do usuário, mapeado para **WishList.user_email**, obrigatório e chave estrangeira.
 -
- **Development**: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1: N . Com a chave estrangeira em **App**.
- **Categorization**: Mapeamento de relacionamento por tabela própria, pois é um relacionamento $N:M$.
 - *item_id*: Chave estrangeira para **Item** e que está fazendo parte da chave primária composta. Mapeada para **Categorization.item_id**, obrigatória.
 - *category_name* e *category_item_type*: Chave estrangeira composta de **Category**, onde *category_name* faz parte da chave composta junto com *item_id* em **Categorization**. Mapeados respectivamente para **Categorization.category_name** e **Categorization.category_item_type**.
- **Attaching**: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1: N . Com a chave estrangeira em **Attachment**.

- **Book_Authorship**: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1: N . Com a chave estrangeira em **Book**.
- **Book_Language**: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1: N . Com a chave estrangeira em **Book**.
- **Movie_Language**: Mapeamento de relacionamento por tabela própria, pois é um relacionamento $N:M$.
 - **type**: Mapeamento direto para **Movie_Language.type**, obrigatório e parte da chave primária composta.
 - **language_name**: Chave estrangeira para **Language** e que está fazendo parte da chave primária composta. Mapeada para **Movie_Language.language_name**, obrigatória.
 - **movie_id**: Chave estrangeira para **Movie** e que está fazendo parte da chave primária composta. Mapeada para **Movie_Language.movie_id**, obrigatória.
- **Movie_Cast**: Mapeamento de relacionamento por tabela própria, pois é um relacionamento $N:M$.
 - **cast_name**: Chave estrangeira para **Language** e que está fazendo parte da chave primária composta. Mapeada para **Movie_Cast.cast_name**, obrigatória.
 - **movie_id**: Chave estrangeira para **Movie** e que está fazendo parte da chave primária composta. Mapeada para **Movie_Cast.movie_id**, obrigatória.
- **Album_Track**: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1: N . Com a chave estrangeira em **Track**.
- **Album_Authorship**: Mapeamento de relacionamento por colunas adicionais, pois é um relacionamento 1: N . Com a chave estrangeira em **Album**.

Referências

- [1] From Google owned by Alphabet Inc., all rights reserved, 2015-2019. Available from World Wide Web: (<https://play.google.com/store>)
- [2] Owned by Alphabet Inc., all rights reserved, 2015-2019. Available from World Wide Web (<https://abc.xyz/>)
- [3] Defined in Email address (Wikipedia). Accessed in 31/05/2019. Available from World Wide Web (https://en.wikipedia.org/wiki/Email_address)
- [4] SHA-2 cryptography, as defined at Wikipedia. Accessed in 31/05/2019. Available from World Wide Web (<https://en.wikipedia.org/wiki/SHA-2>)
- [5] International Standard Book Number. Accessed in 31/05/2019. Available from World Wide Web (<http://www.isbn.bn.br/website/>)