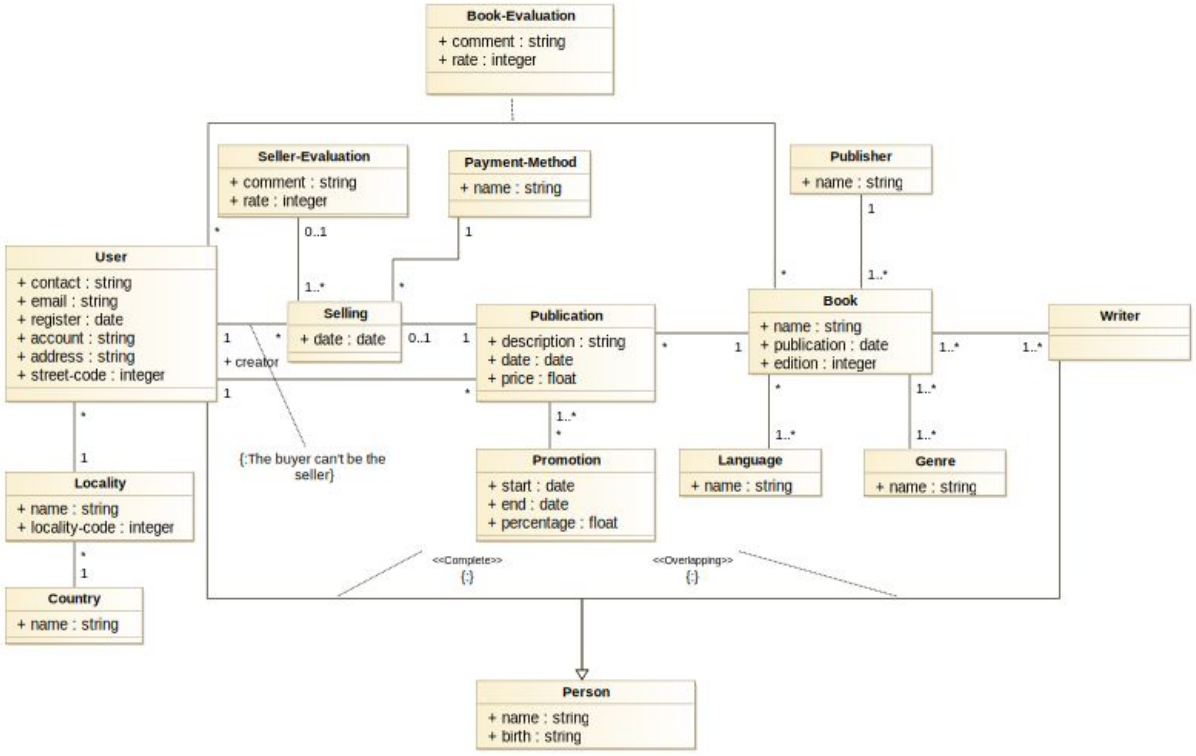
Projeto de BDAD

Sistema de compra e venda de livros

Grupo 404
Alexandre Almeida de Abreu Filho - up201800168
Juliane de Lima Marubayashi - up201800175
João Castro Pinto - up201806667

Porto, 2020

Modelo UML



Contexto

O projeto modela um sistema de compra e venda de livros online, onde os próprios utilizadores realizam as vendas e compras.

User

A classe user representa os vendedores e compradores de livros, tem como atributos o contacto, email, data de registo no sistema, conta bancária, e morada (que é representada por house-number, street, postal-code e Locality).

Selling

Selling representa uma compra dos itens de uma publicação. Quem faz a compra é o utilizador que está diretamente associado à classe “Selling”. O vendedor está indiretamente associado a selling através da publicação. O criador associado à publicação é quem está a fazer a venda.
Quando se faz uma compra, é possível avaliar o utilizador que vendeu o livro. Esta informação guarda-se no objeto “Seller-Valuation”, que tem como atributos uma nota de 1 a 5, e um comentário sobre o vendedor.
Tem como atributos a data de realização da compra e o método de pagamento (indicado pela classe Method)

Publication

Quando se coloca um livro à venda, faz-se através de uma publicação. Portanto, terá de estar associado a um utilizador quando é criado, ao livro a que se está a vender, e às promoções que são ou foram aplicadas a ele. Se um livro for vendido, ou seja, se alguém comprar o livro que está à venda na publicação, cria-se um objeto “Selling”.

Promotion

Uma publicação específica pode ter uma promoção inicial e mais tarde pode-se adicionar outra, portanto uma publicação pode ter várias promoções aplicadas e assim a relação é de um para muitos.

Book-Valuation

A fim de ter uma avaliação fiel a reputação do livro, um cliente não necessariamente precisa comprá-lo para o avaliar. Assim, mesmo que determinado livro nunca tenha sido vendido no site, este ainda pode possuir um histórico de avaliações e, portanto, compradores futuros irão possui um parâmetro de qualidade.

Book

A classe Book representa uma obra e tem como atributos o nome, a data de publicação a edição, o idioma, o gênero, o autor e a editora.

Person

Person é uma generalização da classe “User” e “Writer”, sendo “Writer” a classe que representa um escritor de um livro.

Definição do esquema relacional

Publication(id, description, date, price, idUser->User, ISBN->Book)
Publisher(id, name)
Selling(idPublication->Publication, idUser->User, date, idPayment->Payment-Method, evaluation->Seller-Evaluation)
Genre(id, name)
Language(code, name)
Person(id, name, birth)
Locality(locality-code, name, nameCountry->Country)
Country(code, name)
User(idPerson->Person, contact, email, register, account, address, street-code, locality-code->Locality)
Writer(idPerson->Person)
Book(ISBN, name, publication, edition, idPublisher->Publisher)
Promotion(id,start, end, percentage)
Payment-Method(id, name)
Seller-Evaluation(id, comment, rate)
Book-Evaluation(idPerson->User, ISBN->Book, comment, rate)

Relações de associação:

Book-Writer(idWriter->Writer, ISBN->Book)
Book-Genre(ISBN->Book, idGenre->Genre)
Book-Language(ISBN->Book,codeLanguage->Language)
Publication-Promotion(idPublication->Publication, idPromotion->Promotion)

Análise de Dependências Funcionais e Formas Normais

Publication(id, description, date, price, idUser, ISBN)
    id -> description, date, price, idUser, ISBN
Publisher(id, name)
    id -> name
Selling(idPublication, idUser, date, idPayment, evaluation)
    idPublication -> idUser, date, idPayment, evaluation
Genre(id, name)
    id -> name
    name -> id
Language(code, name)
    code -> name
    name -> code
Person(id, name, birth)
    id -> name, birth
Locality(locality-code, name, nameCountry)
    locality-code -> name, nameCountry
Country(code, name)
    code -> name
    name -> code
User(idPerson, contact, email, register, account, address, street-code, locality-code)
    idPerson -> contact, email, register, account, address, street-code, locality-code
    email -> idPerson, contact, register, account, address, street-code, locality-code
    contact -> idPerson, email, register, account, address, street-code, locality-code
    account -> idPerson, contact, email, register, address, street-code, locality-code
    address -> locality-code, street-code
Writer(idPerson)
    (nothing)
Book(ISBN, name, publication, edition, idPublisher)
    ISBN -> name, publication, edition, idPublisher
Promotion(id, start, end, percentage)
    id -> start, end, percentage
    start, end, percentage -> id
Payment-Method(id, name)
    id -> name
    name -> id
Seller-Evaluation(id, comment, rate)
    id -> comment, rate
    comment, rate -> id
Book-Evaluation(idPerson, ISBN, comment, rate)
    idPerson, ISBN -> comment, rate
Book-Writer(idWriter, ISBN)
    (nothing)

Book-Genre(ISBN, idGenre)
(nothing)
Book-Language(ISBN,codeLanguage)
(nothing)
Publication-Promotion(idPublication, idPromotion)
(nothing)

Conclusão

Atributos que seguem as formas normais

3 Forma Normal:
Todas as dependências pertencem a 3 forma normal, exceto o **User.**
BCNF:
Todas as dependências pertencem a BCNF, exceto o **User.**