

Engenharia Informática

Relatório de Base de Dados

Projeto realizado por:

Daniel Epifânio Lopes, 99104;

Flávio Miguel Freire Martins, 99360;

Guilherme Martins Nunes, 98662;

Vasco do Nascimento Paiva Araújo, 98654.

**Índice:**

Introdução…………………………………………………………………………………………… 3

Diagrama de Classes:

Catálogo Bibliográfico…………………………………………………………4, 5, 6

Utilização e Utentes:

Utentes…………………………………………………………………………………7

Empréstimos………………………………………………………………………..8

Arrumação…………………………………………………………………………..9

Listas de Leitura…………………………………………………………………10

Diagrama de Classes (completo)……………………………………………..11

Modelo Relacional…………………………………………………………………………12

**Introdução:**

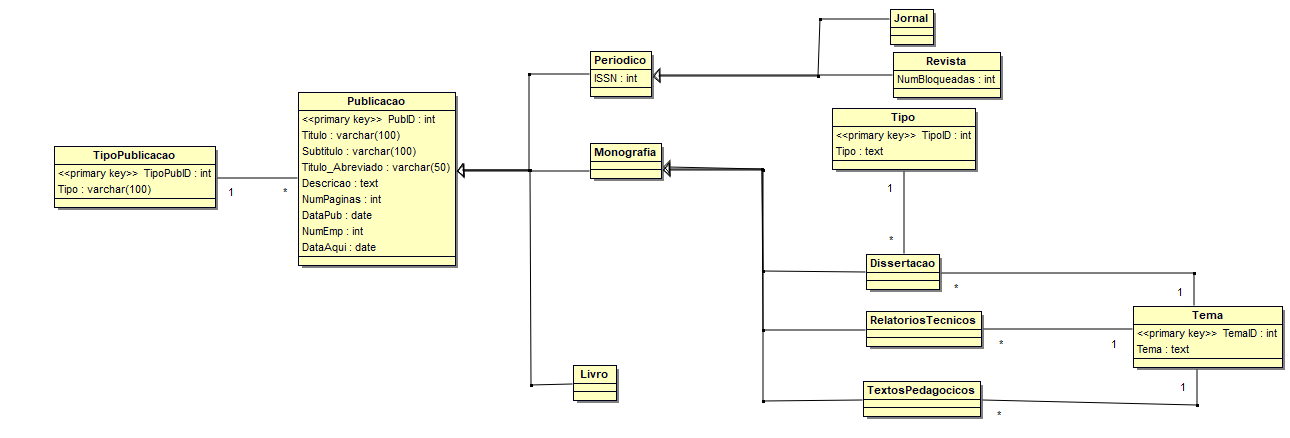
Este projeto tem como finalidade a construção e visualização de um sistema de gestão bibliográfica para o apoio ao funcionamento de uma biblioteca num ambiente universitário.

Primeiramente construímos o diagrama de classes a partir da interpretação do enunciado do projeto. Onde fizemos todas as ligações entre classes, atribuição de atributos, e definição de multiplicidades, tipos de dados e índices.

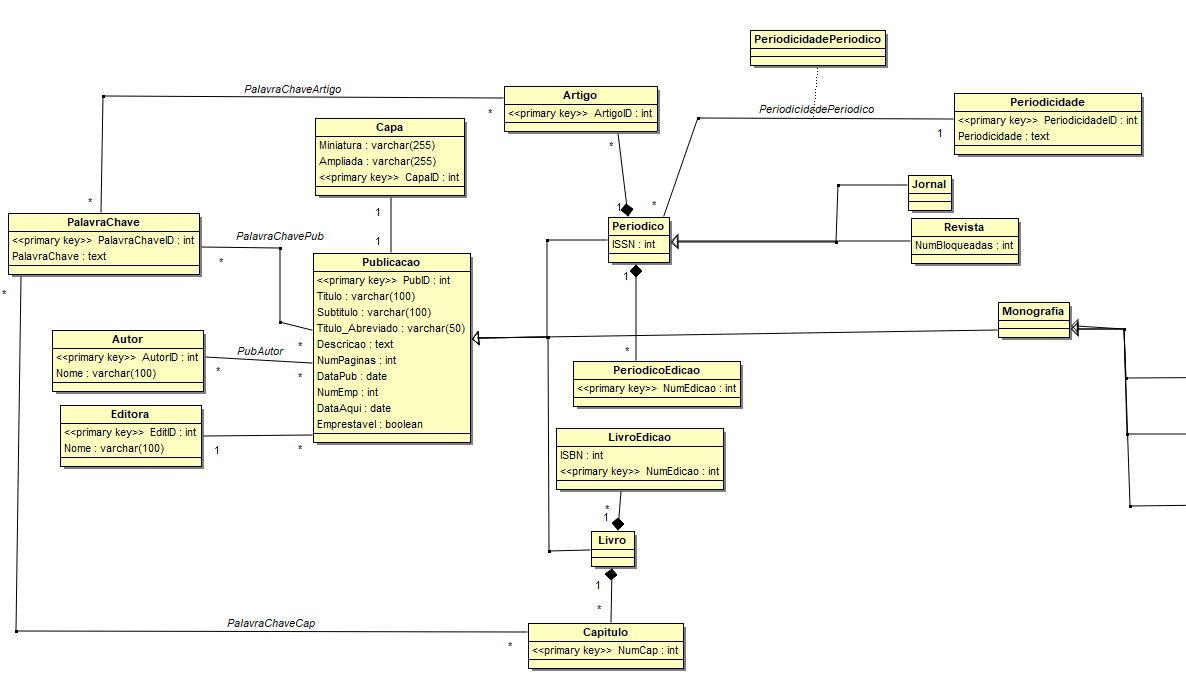
Seguidamente, convertemos através do UMLToRel, o diagrama de classes para código SQL, que por sua vez foi introduzido no PHPMyAdmin, onde o código foi convertido para o modelo relacional, onde acrescentámos definições importantes.

**Diagrama de Classes:**

1. Catálogo Bibliográfico:



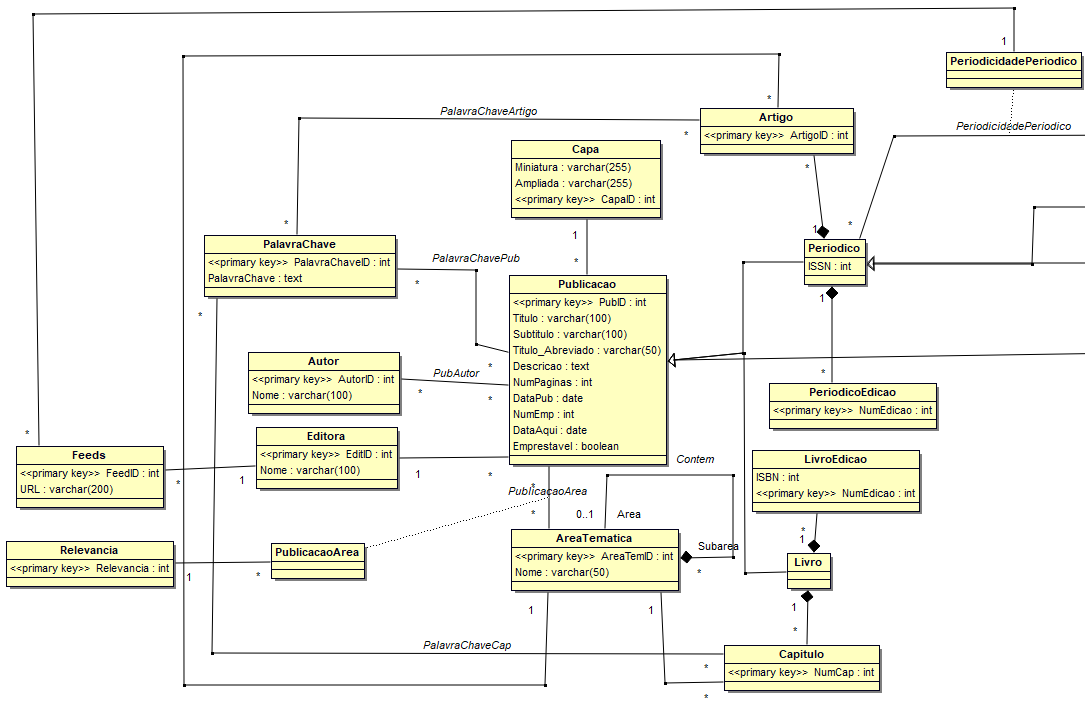
Começámos por criar a classe **Publicacao** com os seus respetivos atributos e particularizações (para **Livro**, **Monografia** e **Periodico**). Esta por sua vez, está associada a uma classe **TipoPublicacao** (que contem todas as particularizações), com o intuito de na base de dados, para além destes tipos estarem predefinidos e não puderem ser alterados pelo utilizador, ser mais fácil a identificação do respetivo tipo da publicação. A classe **Monografia** por sua vez, também tem as suas particularizações, nomeadamente **Dissertacao** (**Tipo** mestrado ou **Tipo** doutoramento), **RelatoriosTecnicos** e **TextosPedagocicos**. Estas três particularizações estão associadas, cada uma, a um **Tema**. A classe **Periodico** pode ser um **Jornal** ou uma **Revista**.



Uma **Publicacao** tem associada uma única **Capa**, sendo que a capa ampliada e miniatura, são do tipo varchar pois são a localização do ficheiro da imagem. A **Publicacao** pode ter um ou vários autores (classe **Autor**) dos registados na base de dados, assim como para uma **Editora**.

A classe **Publicacao** pode estar associada a várias **PalavraChave** ou *tags,* tal como qualquer **Capitulo** ou **Artigo**.

A classe **Livro** é composta por vários capítulos (classe **Capitulo**) e por várias edições (classe **LivroEdicao**), o mesmo para a classe **Periodico** que é composta por vários artigos (classe **Artigo**) e por várias edições (classe **PeriodoEdicao**). Cada **Periodico** tem uma **Periodicidade** associada, podendo ser diária, semanal, mensal, etc.



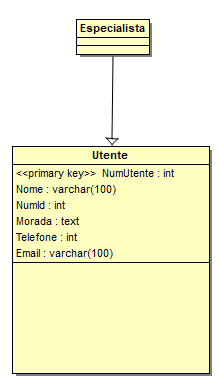
A **Publicacao** pode estar associada a várias áreas temáticas (classe **AreaTematica**). Para saber quais as publicações mais relevantes numa dada área (escaladas de 1 a 5), associou-se a classe **Relevancia** à classe associativa **PublicacaoArea**. Os capítulos e os artigos, estão associados a uma área temática.

As áreas temáticas estão associadas reflexivamente, pois cada área é composta por uma ou mais subáreas, e cada subárea só está contida no máximo numa área. Sendo também de relatar que uma área mãe, não está associada a nenhuma outra área.

Cada feed (classe **Feeds**) está associado a uma editora e tem como atributo o URL ao qual a biblioteca visita consoante a periodicidade do periódico em questão.

1. Utilização e Utentes:

2.1) Utentes:

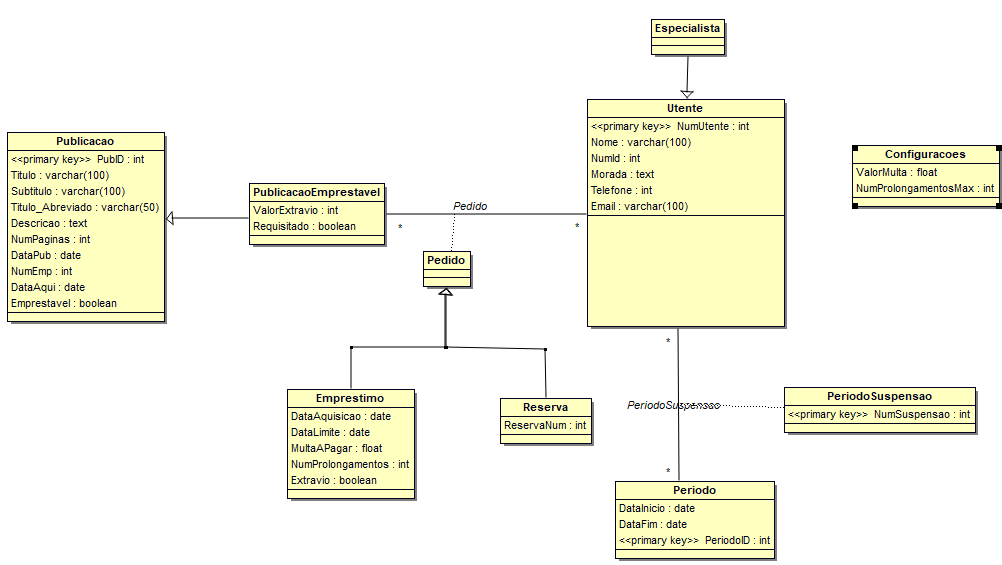


Nesta fase, criámos a classe Utente, para representar os frequentadores da biblioteca. Estes são identificados pelo nome, Numid (cartão de cidadão ou passaporte), morada, número de telefone e email.

Para a identificação dos utentes na biblioteca, é lhes atribuído um NumUtente, que é usado para que todas a interações destes, sejam registadas.

Um utente para puder atribuir relevância a uma dada publicação numa certa área temática, precisa de ser um utente especialista.

2.2) Empréstimos:

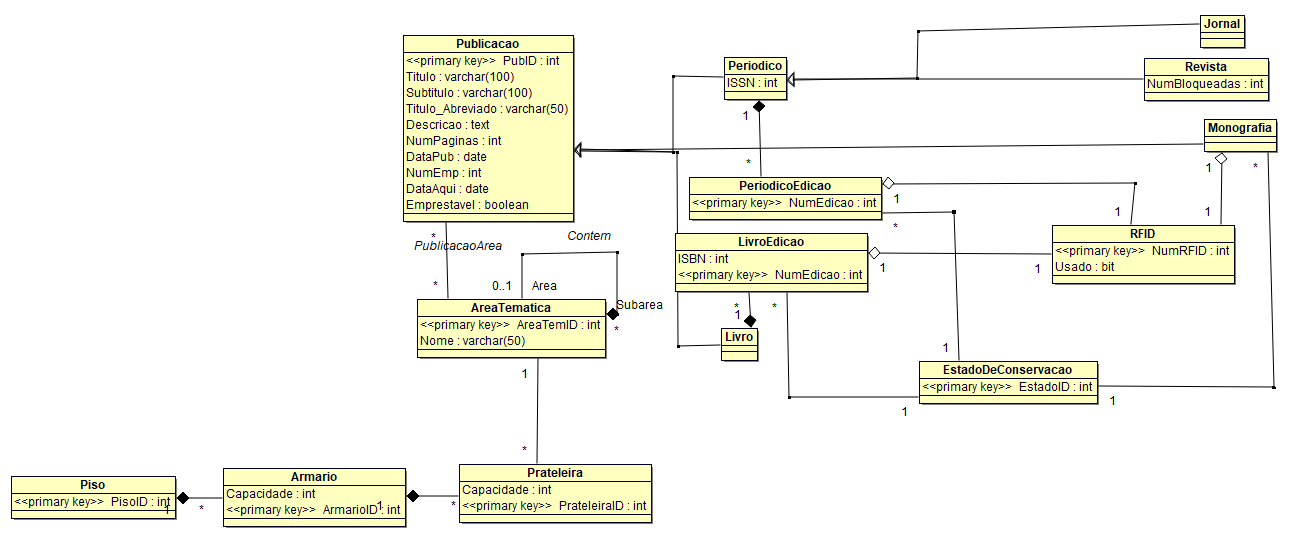


Uma publicação pode ser emprestável (classe **PublicacaoEmprestavel**) consoante o valor do atributo boolean Emprestavel. Caso a publicação seja emprestável, contém um valor de extravio associado e o atributo Requisitado, que permite saber se a publicação se encontra disponível na biblioteca ou se já foi requisitado por um utente.

Um utente pode realizar vários pedidos de requisição, caso a publicação já esteja requisitada por outro utente, esse pedido torna-se numa **Reserva** para essa publicação ao qual será atribuída um número de reserva. Se a publicação estiver disponível, é efetuado o **Emprestimo**, o qual contém o valor Extravio, que regista se a publicação foi furtada ou estragada pelo respetivo utente; o valor MultaAPagar, que é calculado multiplicando o número de dias em atraso pela constante guardada na classe **Configuracoes** no campo ValorMulta; e o NumProlongamentos, que é um contador que não pode exceder a constante guardada na classe **Configuracoes** no campo NumProlongamentosMax.

O utente está associado a um ou mais períodos (classe **Periodo**) pela associação *PeriodoSuspensao* que garante que o mesmo utente não pode estar associado ao mesmo período duas vezes.

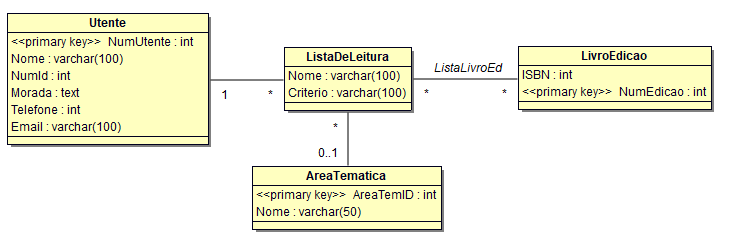
2.3) Arrumação:



Um **Piso** é composto por vários armários (classe **Armario**), que por sua vez, são compostos por prateleiras (classe **Prateleira**). Cada **Prateleira** e cada **Armario** possuem um atributo Capacidade. Cada prateleira está associada a uma área temática e a várias publicações.

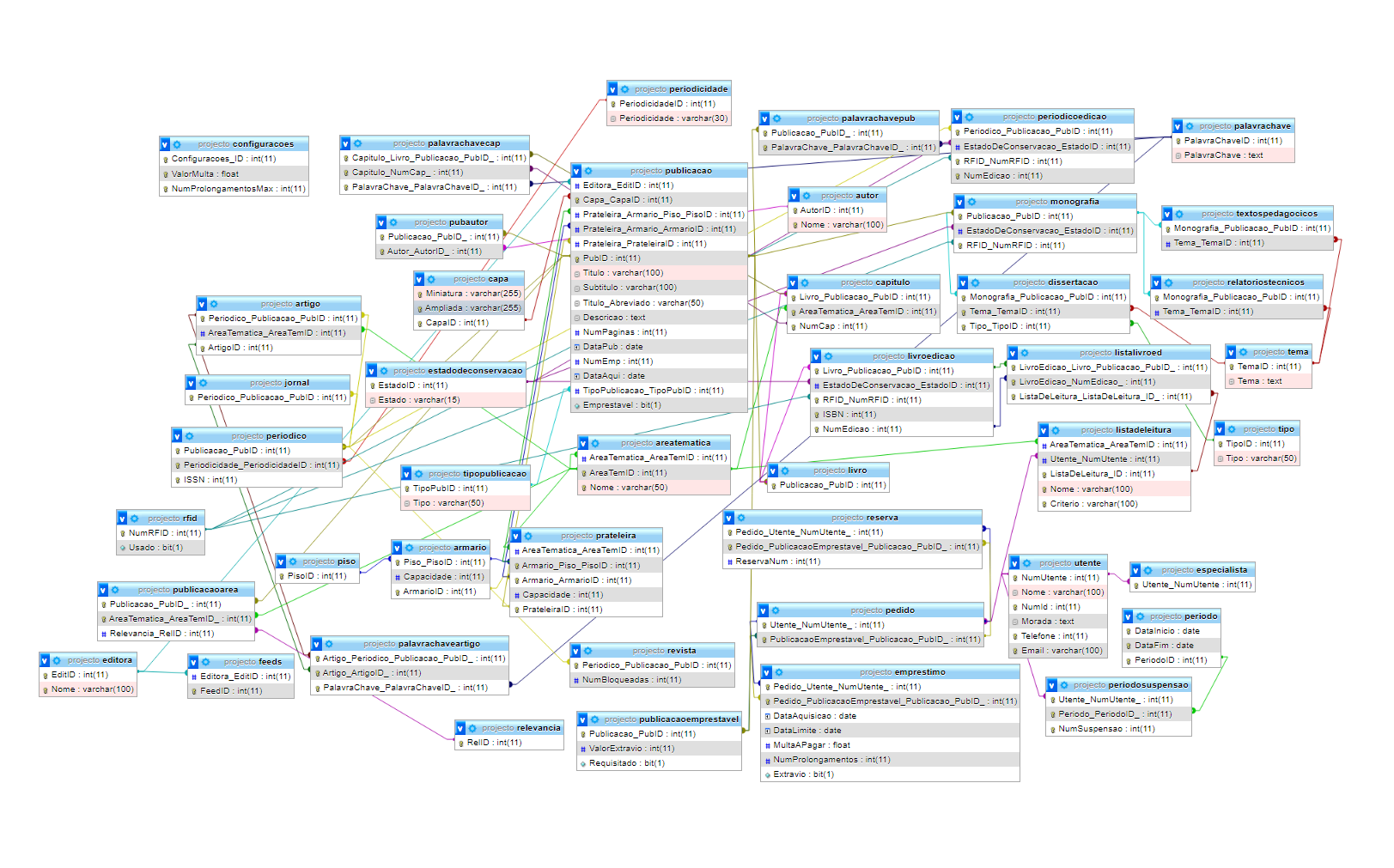
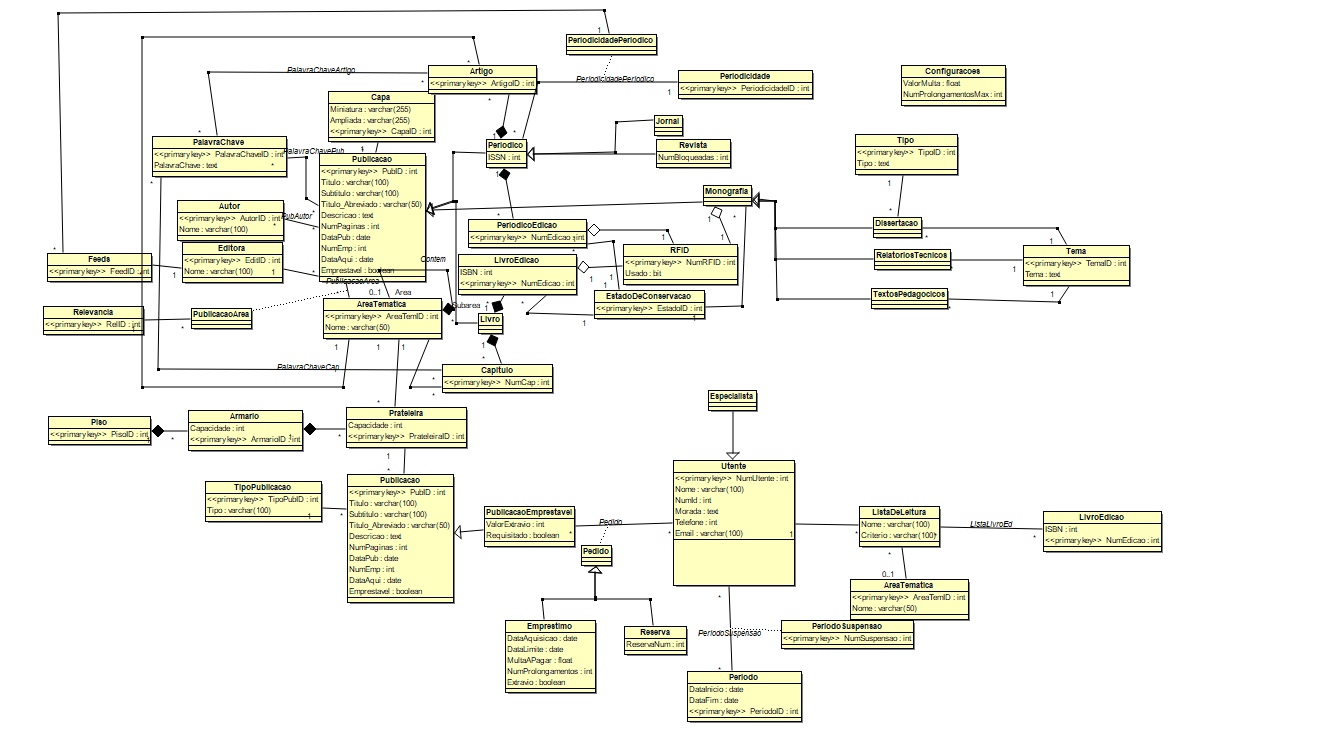
Cada exemplar tem uma etiqueta **RFID** e um estado de conservação (novo, gasto, inutilizado e extraviado) associados. O atributo Usado da classe **RFID**, é um bit que permite saber se um RFID está a ser ou não utilizado (desta forma por via de programação, podemos evitar que o mesmo RFID seja atribuído a várias exemplares).

2.4) Listas de Leitura:



Um utente pode criar várias listas de leitura (classe **ListaDeLeitura**), com os livros e a respetiva edição que lhe interessam, dando-lhe um nome e uma área temática ou outro critério que indique

Diagrama de Classes (Completo):

Modelo Relacional: