

for recebida, a mesma música deverá tocar. O atributo **tom** é uma **String** porque se refere ao nome do tom que deverá ser tocado e o aparelho tem um conjunto de tons que podem ser selecionados, mas somente um para as chamadas recebidas, outro para as mensagens recebidas e outro para o despertar do alarme.

- **Tom** – Essa classe armazena os tons disponíveis no aparelho. Seus atributos são o nome do tom e o nome do arquivo que o celular deverá procurar para executar a música, ambos do tipo **String**. Os métodos dessa classe são **selecionarTom**, que apresenta ao usuário uma lista na qual este deverá escolher entre ligação, mensagem ou alarme. Se o proprietário do aparelho escolher uma das opções, o método chamará o método **listarTom** para visualizar os tons disponíveis e permitir que o usuário selecione um deles para a função previamente escolhida. Esta classe possui associações com as classes **Ligacao**, **Mensagem** e **Alarme** por meio das quais são definidos os sons que serão executados quando do recebimento de uma ligação, mensagem ou quando o alarme for ativado. Por meio dessas associações, pode-se verificar que um objeto da classe **Tom** pode estar associado à classe **Ligacao** ou à classe **Mensagem** ou à classe **Alarme**.

## 4.11 Exemplo de Diagrama de Classes – Sistema de Biblioteca

Nesta seção, apresentaremos o modelo de domínio representado pelo diagrama de classes do sistema de controle de biblioteca que iniciamos a modelar no capítulo sobre o diagrama de casos de uso, conforme ilustrado pela figura 4.54.

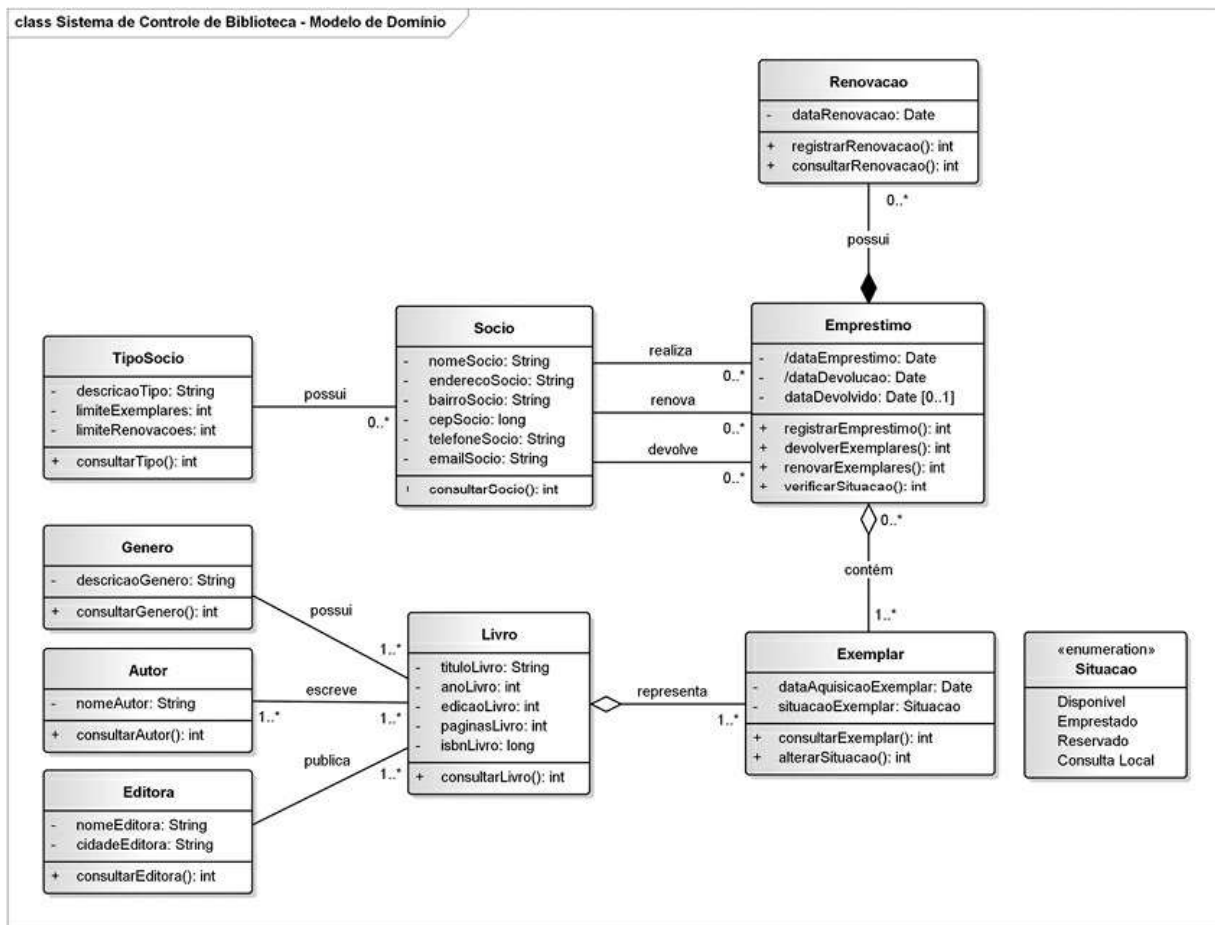


Figura 4.54 – Diagrama de Classes para o Sistema de Biblioteca – Modelo de Domínio.

Nesse modelo, explicitamos somente os métodos que consideramos importantes e são utilizados pelos casos de uso primários. Não consideramos necessário representar métodos cuja única função seria registrar ou excluir objetos em classes simples. Detalharemos a seguir as classes de entidade identificadas nesse modelo.

- **TipoSocio** – Essa classe representa os tipos de sócio aceitos pela biblioteca, como professores, alunos, funcionários e pessoas da comunidade externa. Essa classe é importante porque, além de determinar que tipo de sócio está realizando um empréstimo, estabelece qual o limite máximo de exemplares um sócio de um determinado tipo pode tomar emprestado por meio do atributo **limiteExemplares**, do tipo inteiro. Quando esse limite for atingido, o sócio não poderá locar mais livros até que devolva algum exemplar que tenha tomado

emprestado anteriormente. A classe também estabelece o número máximo de vezes que um sócio pode renovar um empréstimo, por meio do atributo **limiteRenovacoes**, do tipo inteiro. Essa classe poderia ser uma enumeração, porém preferimos deixá-la como uma classe de entidade para permitir que novos tipos de sócio sejam cadastrados sem que seja necessário dar manutenção ao sistema. A classe **TipoSocio** também contém um método para consultar um tipo de sócio.

- **Socio** – Essa classe armazena as informações dos sócios que podem retirar livros emprestados da biblioteca. Seus atributos e métodos são autoexplicativos. Observe que uma instância da classe **Socio** possui três associações com a classe **Emprestimo**, que informam que um sócio pode realizar nenhum ou muitos empréstimos, bem como devolver ou renovar nenhum ou vários empréstimos.
- **Gênero** – Essa classe tem como função armazenar os possíveis gêneros de livros possuídos pela biblioteca, como aventura, ficção científica, época, romance, terror, história, geografia, matemática, biologia, física etc. Essa classe tem como atributo a descrição do gênero do tipo **String**. Também representamos aqui um método para consultar um determinado gênero.
- **Autor** – Essa classe tem como função armazenar os autores que escreveram os livros possuídos pela biblioteca. Como pode se perceber pelas multiplicidades da associação entre esta classe e a classe Livro, um autor pode escrever muitos livros e um livro pode ser escrito por muitos autores. Esta classe tem como atributo o nome do autor do tipo **String** e um método para consultar um autor específico.
- **Editora** – Essa classe tem como função armazenar as editoras que publicaram os livros possuídos pela biblioteca. Essa classe possui os atributos para conter o nome e a cidade da editora do tipo **String**, além de um método para consultar uma editora específica.
- **Livro** – Esta classe armazena as informações relativas aos livros possuídos pela biblioteca. Seus atributos são título do tipo **String**, ano, edição, número de páginas do tipo **int** e ISBN do tipo **long**. Há, ainda, um método para consultar um determinado livro. Observe que a classe possui associações com diversas outras classes do modelo. Além da

associação já explicada com a classe **Autor**, pode-se perceber que uma instância da classe **Livro** deve estar associada a uma instância da classe **Genero** e a uma instância da classe **Editora**, posto que um livro precisa pertencer a um gênero e ser publicado por uma editora. Há, ainda, uma associação de agregação com a classe **Exemplar** que demonstra que as informações de um livro precisam ser complementadas pelas informações dos exemplares a ele associados.

- **Exemplar** – Essa classe armazena as informações referentes a cada exemplar de um livro. A classe **Livro** armazena as informações dos livros possuídos pela biblioteca, mas a classe **Exemplar** refere-se às cópias físicas existentes na biblioteca para empréstimo. Os atributos desta classe são a data em que o exemplar foi adquirido, do tipo **Date**, e a situação do exemplar, do tipo **Situacao**. **Situacao** é uma classe de enumeração cujos literais definem se o exemplar se encontra disponível, está emprestado, está reservado ou só pode ser consultado dentro da biblioteca. Esta classe possui ainda os métodos **consultarExemplar** e **alterarSituacao**: o primeiro serve para consultar um exemplar específico e o último, para alterar a situação de um determinado exemplar. A classe **Exemplar** possui ainda uma associação com a classe **Livro**, uma vez que cada exemplar representa fisicamente um livro, todavia um livro pode ser representado por muitos exemplares.
- **Emprestimo** – Essa classe tem por função registrar os empréstimos de exemplares feitos pelos sócios da biblioteca. Os atributos relevantes para essa classe são a data em que o empréstimo foi realizado, a data prevista de devolução e a possível data em que o empréstimo foi realmente devolvido, todos do tipo **Date**. Note que os dois primeiros atributos são derivados, uma vez que a primeira data é tomada do sistema e a segunda, calculada com base na primeira. Observe ainda que o atributo que representa a data em que o empréstimo foi efetivamente devolvido pode ter, no mínimo, nenhum e, no máximo, um valor, uma vez que essa data fica em aberto até todos os exemplares associados ao empréstimo terem sido devolvidos. A classe tem ainda os métodos **registrarEmprestimo**, por meio do qual um novo empréstimo é gerado e os exemplares envolvidos marcados como emprestados;

**devolverExemplares**, responsável por definir que exemplares do empréstimo foram devolvidos e, se todos foram devolvidos, que o empréstimo foi encerrado; **renovarExemplares**, que permite renovar o empréstimo de um ou mais exemplares; **verificarSituacao**, que verifica a atual situação do empréstimo, se está dentro do prazo, atrasado ou já foi devolvido.

Observe que a classe **Emprestimo** tem uma associação de agregação com a classe **Exemplar**, o que determina que os objetos da classe **Exemplar** são objetos-parte que complementam a informação dos objetos-todo da classe **Emprestimo**. Isto significa que ao consultar um empréstimo, deve-se apresentar também as informações dos exemplares locados nele. Observe ainda que um empréstimo pode conter, no mínimo, um e, no máximo, muitos exemplares e um objeto **Exemplar** pode estar relacionado a muitos objetos da classe **Emprestimo**.

- **Renovacao** – Esta classe tem a função de armazenar as datas em que um empréstimo foi renovado. Possui o atributo **dataRenovacao** do tipo **Date** e os métodos **registrarRenovacao** e **consultarRenovacao** que são autoexplicativos. Como se pode perceber pela associação de composição entre esta classe e a classe **Emprestimo**, um empréstimo pode ser renovado muitas vezes, porém a renovação não é obrigatória, assim um empréstimo pode nunca ser renovado. Além disso, a composição demonstra que as informações do empréstimo precisam ser complementadas pelas possíveis datas em que o empréstimo foi renovado. A quantidade de vezes que um empréstimo pode ser renovado dependerá do tipo de sócio que fez o empréstimo.

## 4.12 Exemplo de Diagrama de Classes – Sistema de Clínica Veterinária

Nesta seção, apresentaremos o diagrama de classes referente ao modelo de domínio para o sistema de clínica veterinária iniciado no capítulo sobre o diagrama de casos de uso. A figura 4.55 apresenta o modelo de domínio desse sistema.

Detalharemos a seguir as classes de entidade identificadas nesse modelo. O leitor notará que declaramos somente os métodos considerados importantes nesse modelo.