



Universidade do Minho

UMinho

Mestrado Engenharia Informática  
Arquiteturas Aplicacionais  
Sistemas Interativos Confiáveis  
(2023/24)

DOCUMENTO ARQUITETURAL

COOLREADS v.1

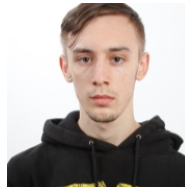
André Lucena   Carlos Machado  
Gonçalo Sousa   João Castro   José Barbosa



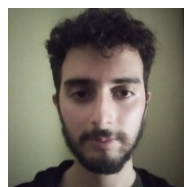
PG52672



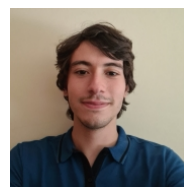
PG52675



PG52682



PG53929



PG52689

20 de junho de 2024

## **Resumo**

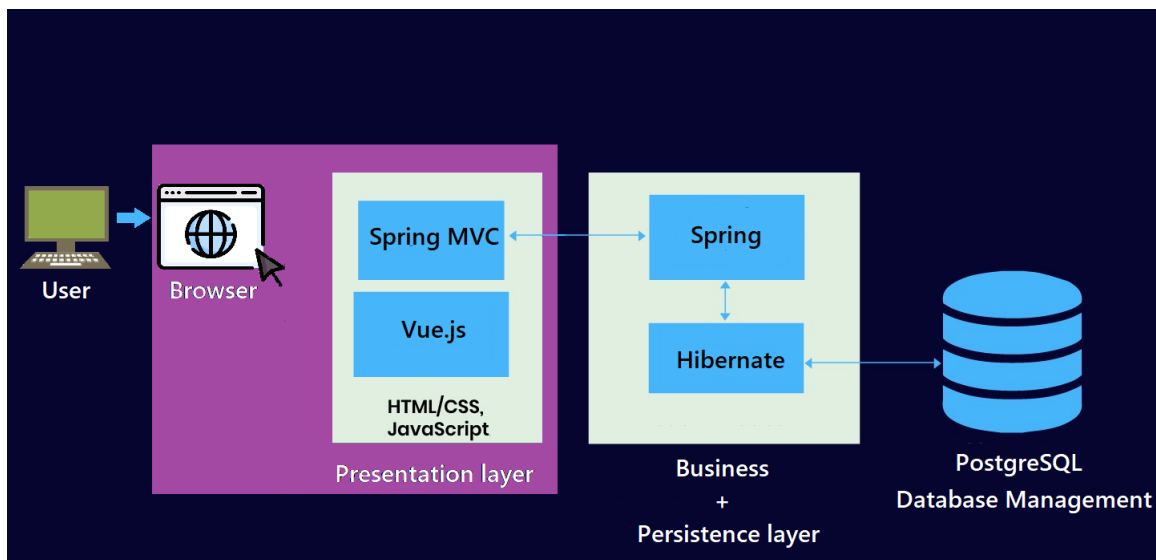
Este documento surge como resultado da proposta de desenvolvimento de uma *Web App* inspirada na plataforma *Goodreads*. Esta aplicação visa oferecer uma experiência focada no leitor, proporcionando uma biblioteca online onde os utilizadores podem monitorar o progresso das suas leituras, interagir com outros leitores, compartilhar interesses em comum e estar atualizados sobre os lançamentos e recomendações de livros.

Documento criado pelo Grupo 1.

# 1. Arquitetura Tipo

A nossa aplicação *web* é dividida em duas vertentes: *frontend* e *backend*, a primeira feita em Vue.js, uma *framework* de **Javascript** e a segunda em Spring Boot, uma *framework* de **Java**. Decidimos construir uma *stack* com as seguintes camadas:

- **Presentation:** *Vue.js(Client)* + *Spring MVC(Server)*
- **Business:** *Spring (Spring Boot)*
- **Integration:** *Hibernate (Spring Data JPA)*
- **Database:** *PostgreSQL*



**Figura 1:** Arquitetura base

Além disso, também escolhemos a base de dados *in-memory* **Redis** para servir como *cache* para a nossa aplicação, voltaremos a desenvolver mais sobre a mesma na parte de *deployment*.

## 2. Tecnologias Frontend

No que toca ao *Frontend* da aplicação utilizamos a framework Vue.js. Para nos ajudar com o design inicial da nossa aplicação, a prototipagem em mockups foi feita utilizando o Figma, que nos permitiu perceber como seria a apresentação e o esqueleto da Web App. Por outro lado, esta fase permitiu-nos extrair novos requisitos. Com base nesse design partimos para a implementação da interface.

O lado frontend da nossa aplicação encontra-se na pasta *coolreads* do nosso código. Para além das definições e dos ficheiros default e comuns da framework implementamos grande maioria do nosso código dentro da pasta *src*, sendo a estrutura é a seguinte:

- **assets:** Pasta com componentes default do Vue.js
- **components:** Pasta onde se definiram componentes individuais bastante importantes na definição do nosso projeto, que foram integrados nas views.
- **models:** Usada apenas para definir uma classe user (utilizador) para o processo de autenticação.
- **router:** Contém o ficheiro *index.js* que define as rotas de acesso a cada uma das views deste lado da implementação.
- **services:** Pasta onde estão os services utilizados, inclui autenticação e notificações.
- **store:** Pasta onde se definiram as stores que foram utilizadas e integradas na nossa Web App.
- **views:** Pasta onde se definiram todas as views utilizadas na nossa Web App.

Também existe uma pasta na diretoria *public*, fora da *src*, que contem todas as imagens estáticas do frontend que serão utilizadas nas páginas.

No que toca à lógica e soluções de implementação. A navegação dentro da nossa aplicação funciona através de um componente chamado *NavComponent*. Este gere a navegação permitindo o acesso a diferentes secções da aplicação ("Home", "Configurações", "Livros", "Estantes", "Notificações"). Ele também implementa a lógica para a pesquisa de livros, onde os resultados são exibidos dinamicamente à medida que o utilizador escreve o que pretende. Além disso, permite a navegação para os menus de autenticação, registo e para o utilizador se desconectar.

Em relação à autenticação e ao registo eles podem ser efetuados dentro dos respetivos menus, definidos dentro das Views *LoginPage.vue* e *SignupPage.vue*.

No *LoginPage.vue*, o utilizador introduz o seu nome de utilizador e palavra-passe para se autenticar. Se o utilizador já estiver autenticado, é redirecionado automaticamente para a última rota acessada. É possível alternar a visibilidade da palavra-passe para conforto do utilizador. A lógica de autenticação verifica se os campos estão preenchidos e, em caso de erro, exhibe mensagens apropriadas.

No *SignupPage.vue*, o utilizador pode criar uma conta fornecendo um email, nome de utilizador e palavra-passe. Tal como no login, a interface oferece permite mudar visibilidade da palavra-passe. A lógica de registo verifica se todos os campos estão preenchidos e, em caso de sucesso, redireciona o utilizador para a página de login.

Depois temos o *auth.module* e o *auth.service*. O módulo gere o estado do utilizador, as ações de login, logout e registo, e as mutações associadas para atualizar o estado conforme necessário. A store irá permitir-nos fazer a gestão se o utilizador está autenticado ou não, e utilizando a biblioteca *jwt-decode*, fizemos o decode do token e permiti-nos fazer a gestão de quando a token expira e consequentemente o utilizador terá de efetuar novo login caso pretenda. O serviço lida com as requisições para a API, como login, logout, registo e alteração de palavra-passe, utilizando *axios* para realizar as chamadas HTTP.

Falando da alteração da palavra-passe, esta é realizada no menu de definições. Neste os utilizadores têm a possibilidade de alterar a sua senha, definir a linguagem que preferem, entre Inglês e Português, e selecionar a privacidade da sua estante de livros, embora esta última opção seja puramente estética, pois não tivemos tempo de implementar essa lógica.

Carregar no botão de mudar a palavra passe faz com que surja um menu onde o utilizador tem de introduzir a sua palavra passe atual e aquela que quer agora. Se tudo tiver correto esta troca é realizada. Se a palavra-passe atual tiver avisada, o utilizador é avisado disso.

A mudança de linguagem permite escolher entre as duas línguas acima. Esta mudança afeta elementos que se mantêm consistentes na aplicação, ou seja, não afetam elementos que sejam provenientes do lado do backend. Esta funcionalidade é obtida com a ajuda de uma store que guarda as expressões traduzidas, sendo bastante simples o seu funcionamento e a maneira de adicionar palavras e expressões para traduzir.

Falando agora dos livros em mais detalhe e particularmente como funcionam fora da estante. A *BookMenuView.vue* funciona como uma página principal de navegação dos livros, onde os utilizadores podem explorar diferentes géneros de livros. Cada género é representado por um ícone e um título, como Aventura, Mistério, Biografia, entre outros. Ao clicar num género, os utilizadores são redirecionados para uma página específica desse género, onde podem ver os livros disponíveis nessa categoria. Essa página específica é a *BookCategoryView*. Aqui faz se fetch aos livros que pertencem à categoria, permitindo assim ao utilizador escolher e explorar os que gosta mais. Se carregar num, selecionando-o será direcionado para a página desse livro específico. que é definido na *BookDetailView.Vue*. Aqui o utilizador pode fazer muitas coisas, como ver o autor do livro, adicioná-lo às suas estantes, mudar o estado e ver estatísticas.

O Coolreads permite também aos seus utilizadores de terem as suas próprias bookshelves, estando tal definido na *BookshelfView.vue*. Aqui os utilizadores podem criar e gerir as suas próprias bookshelves virtuais personalizadas. Com ajuda do *ShelfSideBarComponent.vue*, que mostra ao utilizador todas as suas prateleiras, tornando possível visualizá-las apenas carregando na que ele pretende e também criar estantes novas. A View em si, permite aos utilizadores visualizarem todos os livros adicionados às suas bookshelves, organizados de acordo com as suas preferências de ordenação, de modo a tornar a User Experience mais completa. No fundo, esta View, permite ao utilizador visualizar e organizar de acordo com o seu gosto pessoal os seus livros, coleções e estantes.

O *HomeView.vue* acaba por ser o menu de início da nossa aplicação. Ele contém também um *ShelfSideBarComponent.vue*, que nos leva para a bookshelf que falamos no parágrafo anterior.

Existe também um ecrã de notificações, definido na *NotificationView.vue*. Quando se carrega na opção de notificações do menu de navegação aparece um minimenu dropdown que contém as duas notificações no topo do ecrã de notificações completo, que é acessível através do botão no fim deste dropdown menu. Cada notificação é apresentada de forma clara, incluindo detalhes como o tipo de notificação (pedido de amizade ou novo post, por exemplo), há quanto tempo foi recebida, e o nome do utilizador responsável pela ação.

Depois temos o *notifications.module* e o *notifications.service*. O módulo contém actions para buscar notificações do servidor. As mutations são responsáveis por modificar o estado das notificações conforme necessário, como adicionar novas notificações e atualizar a página atual durante a navegação. O service é implementado usando Axios para fazer pedidos HTTP, este serviço faz entre a aplicação Coolreads e o servidor. Ele oferece métodos para buscar notificações com base no nome de utilizador, número da página e tamanho da página, garantindo que apenas as notificações relevantes sejam recuperadas.

Resta-nos falar do perfil e da página de perfil do utilizador da Web App. Esta é definida da *ProfilePageView.vue* e centraliza informações como o nome do utilizador, idade, país, género, descrição pessoal, interesses e prateleira que o utilizador queria destacar. Os utilizadores têm a capacidade de editar esses detalhes diretamente na página, facilitando a atualização e personalização do seu perfil conforme necessário, podendo também adicionar uma imagem de perfil.

## 3. Avaliação de Usabilidade

### 3.1. Plano de Avaliação

O nosso plano para a avaliação da usabilidade da interface foi desenhado ao redor das duas vertentes a que tínhamos mais célere acesso: por Peritos e Empírica. Os testes de usabilidade foram todos conduzidos sobre o protótipo criado em Figma, principalmente devido aos constrangimentos técnicos experienciados pela equipa ao longo do desenvolvimento da aplicação concretamente, não sendo possível aproximá-la decentemente do protótipo para que a análise do *design* fosse válida.

Para esse efeito, foram conduzidas duas fases de avaliação semelhantes, uma após a primeira versão do protótipo e outra depois da segunda versão. A implementação foi envisionada para ser feita com base nesta segunda versão do protótipo. Em ambas as fases, o protótipo foi analisado por Peritos e por Utilizadores, tanto com *Cognitive Walkthrough*, *Heuristic Evaluation* e, com os utilizadores, observação e comentários sobre a prototipagem.

Para ambas as análises dos peritos, apresentamos o resultado da última análise, sobre o protótipo completo. No caso da avaliação das heurísticas, marcamos com "Incompleto" as funcionalidades que não estão presentes na interface final.

### 3.2. Cognitive Walkthrough

| Tarefa:  | Visualizar as Notificações no Sistema |              |       |  |
|--|---------------------------------------|--------------|-------|--|
| Utiliza.:  | Novato no sistema                     |              |       |  |
| Passos   | Questões                              | OK?<br>(S/N) | Risco | Problema/Sugestão/Justificação   |
| 1 O Utilizador navega até à página de notificações | 1                                     | S            |       | Na eventualidade de receber uma notificação nova, o botão de notificações encontra-se com um apontamento de uma cor diferente do fundo, que informa o utilizador dessa presença. |
|  | 2                                     | S            |       | O botão encontra-se sempre visível por fazer parte da barra de navegação standard no topo da página.   |
|  | 3                                     | S            |       | Para além do botão claramente estar registado com o texto "Notificação", o trocar de cor indica que é clicável.  |
|  | 4                                     | S            |       | A resposta à ação é imediata: apresentam-se as notificações ou navega-se para a página pretendida.   |
| 2 O Utilizador visualiza as notificações           | 1                                     | N/A          |       |  |
|  | 2                                     | N/A          |       |  |
|  | 3                                     | N/A          |       |  |
|  | 4                                     | N/A          |       |  |
| 3 O Utilizador sai da pagina de notificações       | 1                                     | S            |       | Após a leitura das notificações, não há mais ações a se tomar próprias desta página. Qualquer outra atividade depende de outras tarefas.   |
|  | 2                                     | N/A          |       | Depende da visibilidade dos botões das outras tarefas, isto é, se esses estiverem visíveis.  |
|  | 3                                     | N/A          |       |  |
|  | 4                                     | N/A          |       |  |



|   |  |              |              |   |
|---|--|--------------|--------------|---|
| Tarefa:   | Publicar Livro por um Autor                                    |              |              |   |
| Utiliza.:   | Autor, que não implica necessariamente conhecimento do sistema |              |              |   |
| Passos  | Questões   | OK?<br>(S/N) | Risco<br>(3) | 1Problema/Sugestão/Justificação   |
| 1<br><br>O Autor seleciona para adicionar um novo livro | 1  | S            |              | Toda a mecânica de adicionar/remover livros funciona a partir de botões e é o primeiro passo natural a se realizar nesta tarefa.  |
|   | 2  | S            |              | O botão faz parte da barra lateral sempre presente, apesar de também estar localizado na página de visualização de todos os livros publicados.  |
|   | 3  | S            |              | O botão tem texto escrito que indica que a sua função é exatamente a pretendida.  |
|   | 4  | S            |              | A resposta à ação é imediata: navega-se para a página de introdução dos detalhes do livro.  |
| 2<br><br>O Autor fornece os detalhes do livro           | 1  | S            |              | Os campos com os detalhes encontram-se diretamente apresentados na nova página com os seus campos de preenchimento vazios. Este é o passo natural no seguimento da tarefa.  |
|   | 2  | S            |              | Todos os campos estão intitulados com os valores que devem ser introduzidos.  |
|   | 3  | S            |              | Os campos de texto são standard para além de alguns indicarem quais os valores pretendidos, como é o caso da data.  |
|   | 4  | S            |              | O preenchimento dos campos mantém-se escrito mesmo após o Autor selecionar outro campo, o que indica que o sistema registou as alterações   |
| 3<br><br>O Autor confirma a criação do livro            | 1  | S            |              | Depois do preenchimento completo dos campos, o sistema não indica imediatamente que o livro foi introduzido com sucesso. O botão de guardar as alterações mantém-se sempre visível e indica que este é o passo final da tarefa. |
|   | 2  | S            |              | O botão de guardar os resultados encontra-se imediatamente visível no topo da página tal como no final de todos os campos de preenchimento.   |
|   | 3  | S            |              | O texto do botão indica exatamente aquilo que o Autor pretende.   |
|   | 4  | S            |              | O sistema pede uma confirmação da ação para além de indicar que o livros e encontra introduzido, alterando todas as interfaces necessárias para o efeito.   |

|   |                                   |              |              |  |
|---|-----------------------------------|--------------|--------------|--|
| Tarefa:   | Consultar Estatísticas dos Livros |              |              |  |
| Utiliza.:   | Novato no Sistema                 |              |              |  |
| Passos  | Questões                          | OK?<br>(S/N) | Risco<br>(3) | 1Problema/Sugestão/Justificação  |
| 1<br><br>O Utilizador seleciona o livro que pretende visualizar as estatísticas | 1                                 | S            |              | O Utilizador não se encontra em nenhuma página específica de um livro, para além da barra de pesquisa indicar que é por lá que se deve pesquisar por livros.   |
|   | 2                                 | S            |              | A barra de pesquisa encontra-se estática no topo do ecrã, mesmo se a página for deslizada. Os livros aparecem imediatamente inferiormente à barra.   |
|   | 3                                 | S            |              | A barra de pesquisa indica que os livros devem ser pesquisados em texto nela escrito. Os livros, quando aproximados pelo ponteiro, alteram o seu fundo para indicar que são clicáveis.   |
|   | 4                                 | S            |              | A resposta à ação é imediata: os livros apresentam-se imediatamente, clicar num navega para a página dos detalhes do livro imediatamente.  |
| 2<br><br>O Autor seleciona as estatísticas do livro                             | 1                                 | S            |              | Os campos com os detalhes do livro não são imediatamente as estatísticas pretendidas, mas sim apresentam as reviews dos utilizadores. O botão de statistics acizentado indica que existe outra aba com os detalhes pretendidos.  |
|   | 2                                 | S            |              | O botão para apresentar as estatísticas encontra-se imediatamente visível na nova página.  |
|   | 3                                 | S            |              | O texto e a diferença entre as cores da aba selecionada para essa indica que este botão deve ser o necessário para executar a troca.   |
|   | 4                                 | S            |              | A aba troca imediatamente para os dados pretendidos  |
| 3<br><br>O Autor seleciona quais as estatísticas que pretende visualizar        | 1                                 | S            |              | As estatísticas encontram-se bem detalhadas nos gráficos pretendidos. Se algum não for o que se entende visualizar, é claro que se devem trocar quais os dados ou os eixos necessários.  |
|   | 2                                 | S            |              | Os botões para alterar os dados que influenciam cada um dos gráficos encontram-se imediatamente antes de cada um deles, com a indicação visual de que se tratam de dropdowns com o título correspondente aos dados apresentados.   |
|   | 3                                 | S            |              | O texto de cada botão indica quais são os dados que são tratados tal como a indicação de ser dropdown implica que há outros valores possíveis e que este é o botão que se deve utilizar para lá chegar. O texto de cada dropdown indica quais os dados que vai demonstrar. |
|   | 4                                 | S            |              | O gráfico altera imediatamente para os dados selecionados.   |

|   |                   |              |              |   |
|---|-------------------|--------------|--------------|---|
| Tarefa:   | Pesquisar Livros  |              |              |   |
| Utiliza.:   | Novato no Sistema |              |              |   |
| Passos  | Questões          | OK?<br>(S/N) | Risco<br>(3) | 1Problema/Sugestão/Justificação   |
| 1<br><br>O Utilizador<br>accede à barra de<br>pesquisa          | 1                 | S            |              | Os Livros identificam-se, para o utilizador, pelo seu título. Por isso, é necessário introduzir os dados num campo que possa incluir texto, como uma barra de pesquisa.                             |
|   | 2                 | S            |              | A barra de pesquisa encontra-se estática no topo do ecrã, mesmo se a página for deslizada, então sempre visível   |
|   | 3                 | S            |              | A barra de pesquisa indica que os livros devem ser pesquisados em texto nela escrito.   |
|   | 4                 | S            |              | A resposta à ação é imediata: a barra apresenta um marcador de escrita à espera do input de texto.  |
| 2<br><br>O Utilizador<br>digita o título do<br>livro pretendido | 1                 | S            |              | A partir do momento que um ponteiro de escrita é demonstrado, o próximo passo é digitar qual o título.  |
|   | 2                 | S            |              | A barra de pesquisa encontra-se estática no topo do ecrã, mesmo se a página for deslizada, então sempre visível   |
|   | 3                 | S            |              | A partir do momento que um ponteiro de escrita é demonstrado, o próximo passo é digitar qual o título.  |
|   | 4                 | S            |              | Os livros são apresentados imediatamente após a escrita do título.  |
| 3<br><br>O Utilizador<br>seleciona o li- vro<br>que pretende    | 1                 | S            |              | O livro é representado textualmente pelo título mas o utilizador ainda precisa de navegar para a página específica do livro, apresentada depois de escrever, o que torna este seguimento intuitivo. |
|   | 2                 | S            |              | Os livros aparecem em dropdown a partir da barra de pesquisa a sobrepor-se aos restantes elementos da página.   |
|   | 3                 | S            |              | Sobrepôr o rato a qualquer destes elementos a representar os livros sobressai-os alterando a cor do fundo, o que indica que o elemento é clicável e que representa o livro em questão.              |
|   | 4                 | S            |              | O utilizador navega imediatamente para a página do livro selecionado  |

|   |                        |              |              |   |
|---|------------------------|--------------|--------------|---|
| Tarefa:   | Pesquisar Utilizadores |              |              |   |
| Utiliza.:   | Novato no Sistema      |              |              |   |
| Passos  | Questões               | OK?<br>(S/N) | Risco<br>(3) | 1Problema/Sugestão/Justificação   |
| 1<br><br>O Utilizador<br>accede à barra de<br>pesquisa                | 1                      | S            |              | Os Utilizadores identificam-se, para o utilizador, pelo seu nome. Por isso, é necessário introduzir os dados num campo que possa incluir texto, como uma barra de pesquisa.   |
|   | 2                      | S            |              | A barra de pesquisa encontra-se estática no topo do ecrã, mesmo se a página for deslizada, então sempre visível   |
|   | 3                      | S            |              | A barra de pesquisa indica que os utilizadores devem ser pesquisados em texto nela escrito.   |
|   | 4                      | S            |              | A resposta à ação é imediata: a barra apresenta um marcador de escrita à espera do input de texto.  |
| 2<br><br>O Utilizador<br>digita o nome do<br>utilizador<br>pretendido | 1                      | S            |              | A partir do momento que um ponteiro de escrita é demonstrado, o próximo passo é digitar qual o nome.  |
|   | 2                      | S            |              | A barra de pesquisa encontra-se estática no topo do ecrã, mesmo se a página for deslizada, então sempre visível   |
|   | 3                      | S            |              | A partir do momento que um ponteiro de escrita é demonstrado, o próximo passo é digitar qual o nome.  |
|   | 4                      | S            |              | Os utilizadores são apresentados imediatamente após a escrita do título.  |
| 3<br><br>O Utilizador<br>seleciona o<br>utilizador que<br>pretende    | 1                      | S            |              | O utilizador é representado textualmente pelo seu nome mas o utilizador ainda precisa de navegar para a página específica do perfil do utilizador, apresentada depois de escrever, o que torna este seguimento intuitivo. |
|   | 2                      | S            |              | Os utilizadores aparecem em dropdown a partir da barra de pesquisa a sobrepor-se aos restantes elementos da página.   |
|   | 3                      | S            |              | Sobrepôr o rato a qualquer destes elementos a representar os utilizadores sobressai-os alterando a cor do fundo, o que indica que o elemento é clicável e que representa o perfil em questão.                             |
|   | 4                      | S            |              | O utilizador navega imediatamente para a página do perfil selecionado   |

|  |                                 |              |              |  |
|--|---------------------------------|--------------|--------------|--|
| Tarefa:  | Gerir Estante (Adicionar Livro) |              |              |  |
| Utiliza.:  | Novato no Sistema               |              |              |  |
| Passos   | Questões                        | OK?<br>(S/N) | Risco<br>(3) | 1Problema/Sugestão/Justificação  |
| 1<br><br>O Utilizador seleciona a Estante pretendida                         | 1                               | S            |              | Cada estante identifica-se pelo seu nome, dado pelo utilizador. Para a gerir, é necessário encontrar-se na página dela.  |
|  | 2                               | S            |              | A barra lateral inclui todas as estantes do utilizador permanentemente, com cada nome a servir de uma ligação para a sua página.   |
|  | 3                               | S            |              | Cada nome na barra lateral é clicável e tal é expresso pelo alterar da sua cor e sublinhado.   |
|  | 4                               | S            |              | A resposta à ação é imediata: o utilizador navega para a página da estante   |
| 2<br><br>O Utilizador seleciona o botão de adicionar um novo livro à estante | 1                               | S            |              | A estante encontra-se estática sem alteração se não existirem interações pela parte do utilizador. Para adicionar um livro é necessário selecionar algum mecanismo que o possibilite.  |
|  | 2                               | S            |              | O botão de adicionar encontra-se explicitado e imediatamente no topo da página antes da estante em si.   |
|  | 3                               | S            |              | O texto do botão indica qual a sua funcionalidade.   |
|  | 4                               | S            |              | Uma caixa de texto é imediatamente apresentada para indicar qual o título do livro pretendido.   |
| 3<br><br>O Utilizador digita o título do livro pretendido                    | 1                               |              |              | A caixa de pesquisa aberta indica que se deve digitar o título do livro para se encontrar o livro pretendido.  |
|  | 2                               |              |              | A barra de pesquisa encontra-se no local onde a ação anterior foi desempenhada, imediatamente no local onde o ponteiro do rato se encontra.  |
|  | 3                               |              |              | A partir do momento que um ponteiro de escrita é demonstrado, o próximo passo é digitar qual o título.   |
|  | 4                               |              |              | Os livros são apresentados imediatamente após a escrita do título.   |
| 4<br><br>O Utilizador seleciona o livro que pretende                         | 1                               |              |              | O livro é representado textualmente pelo título mas o utilizador ainda precisa de selecionar o livro.  |
|  | 2                               |              |              | Os livros aparecem em dropdown a partir da barra de pesquisa a sobrepor-se aos restantes elementos da página.  |
|  | 3                               |              |              | Sobrepôr o rato a qualquer destes elementos a representar os livros sobressai-os alterando a cor do fundo, o que indica que o elemento é clicável e que representa o livro em questão. |
|  | 4                               |              |              | Uma caixa de confirmação é imediatamente apresentada para o utilizador confirmar a introdução do livro.  |

|  |                               |              |              |   |
|--|-------------------------------|--------------|--------------|---|
| Tarefa:  | Gerir Estante (Remover Livro) |              |              |   |
| Utiliza.:  | Novato no Sistema             |              |              |   |
| Passos   | Questões                      | OK?<br>(S/N) | Risco<br>(3) | 1Problema/Sugestão/Justificação   |
| 1<br><br>O Utilizador<br>seleciona a Estante<br>pretendida | 1                             | S            |              | Cada estante identifica-se pelo seu nome, dado pelo utilizador. Para a gerir, é necessário encontrar-se na página dela.                       |
|  | 2                             | S            |              | A barra lateral inclui todas as estantes do utilizador permanentemente, com cada nome a servir de uma ligação para a sua página.              |
|  | 3                             | S            |              | Cada nome na barra lateral é clicável e tal é expresso pelo alterar da sua cor e sublinhado.  |
|  | 4                             | S            |              | A resposta à ação é imediata: o utilizador navega para a página da estante  |
| 2<br><br>O Utilizador<br>procura o livro na<br>estante     | 1                             | S            |              | Todos os livros da estante encontram-se na mesma, com a possibilidade de os ordenar ou de pesquisar   |
|  | 2                             | S            |              | Os livros encontram-se claramente apresentados na estante. A barra de pesquisa da estante está imediatamente em cima da mesma.                |
|  | 3                             | S            |              | O texto da barra de pesquisa indica qual a sua funcionalidade.  |
|  | 4                             | S            |              | A Estante altera-se para demonstrar apenas os livros que correspondem à pesquisa  |
| 3<br><br>O Utilizador<br>remove o livro<br>pretendido      | 1                             |              |              | O livro deve ser removido da estante, tal como indica o pequeno botão vermelho com um sinal de menos no canto superior direito de cada livro. |
|  | 2                             |              |              | Todos os livros contém este botão visivelmente junto da sua capa, na estante.   |
|  | 3                             |              |              | Baseia-se na natureza standard do botão e também no facto que lhe dar hover apresenta uma tooltip que indica essa funcionalidade.             |
|  | 4                             |              |              | Uma caixa de confirmação é imediatamente apresentada para o utilizador confirmar a remoção do livro.  |

|   |   |              |              |   |
|---|---|--------------|--------------|---|
| Tarefa:   | Gerir Estante (Alterar Estante de um Livro) |              |              |   |
| Utiliza.:   | Novato no Sistema                           |              |              |   |
| Passos  | Questões                                    | OK?<br>(S/N) | Risco<br>(3) | 1Problema/Sugestão/Justificação   |
| 1<br><br>O Utilizador seleciona a Estante pretendida            | 1   | S            |              | Cada estante identifica-se pelo seu nome, dado pelo utilizador. Para a gerir, é necessário encontrar-se na página dela.   |
|   | 2   | S            |              | A barra lateral inclui todas as estantes do utilizador permanentemente, com cada nome a servir de uma ligação para a sua página.  |
|   | 3   | S            |              | Cada nome na barra lateral é clicável e tal é expresso pelo alterar da sua cor e sublinhado.  |
|   | 4   | S            |              | A resposta à ação é imediata: o utilizador navega para a página da estante  |
| 2<br><br>O Utilizador procura o livro na estante                | 1   | S            |              | Todos os livros da estante encontram-se na mesma, com a possibilidade de os ordenar ou de pesquisar   |
|   | 2   | S            |              | Os livros encontram-se claramente apresentados na estante. A barra de pesquisa da estante está imediatamente em cima da mesma.  |
|   | 3   | S            |              | O texto da barra de pesquisa indica qual a sua funcionalidade.  |
|   | 4   | S            |              | A Estante altera-se para demonstrar apenas os livros que correspondem à pesquisa  |
| 3<br><br>O Utilizador visualiza os detalhes do livro pretendido | 1   | S            |              | Cada livro terá detalhes particulares do utilizador, como a sua review ou as estantes em que se encontra. Para além de se encontrarem na sua página de livro na biblioteca geral, o utilizador quererá vê-los na sua estante. |
|   | 2   | S            |              | Os livros encontram-se na estante. Mater o rato sobre eles além de os ampliar apresenta um texto a indicar que clicar apresenta os detalhes.  |
|   | 3   | S            |              | O texto é claro a indicar que clicar sobre o livro irá apresentar os seus detalhes.   |
|   | 4   | S            |              | Os detalhes do livro são apresentados exatamente sobre o livro selecionado.   |
| 4<br><br>Alterar a estante do livro                             | 1   | S            |              | Com os detalhes do livro selecionados, onde constam as estantes desse mesmo livro, basta alterar quais as estantes em que se encontra.  |
|   | 2   | S            |              | Os detalhes do livro são apresentados exatamente sobre o livro selecionado. As estantes do livro estão apresentados neles tal como o botão de acrescentar o livro a uma nova estante.   |
|   | 3   | S            |              | O texto do botão da nova estante é consistente com os outros locais onde se encontra  |
|   | 4   | S            |              | Uma nova estante é adicionada aos detalhes do livro, após a confirmação explícita pelo utilizador.  |

|  |   |              |              |  |
|--|---|--------------|--------------|--|
| Tarefa:  | Gerir Estante (Alterar Visibilidade da Estante) |              |              |  |
| Utiliza.:  | Novato no Sistema                               |              |              |  |
| Passos   | Questões  | OK?<br>(S/N) | Risco<br>(3) | 1Problema/Sugestão/Justificação  |
| 1<br><br>O Utilizador seleciona a Estante pretendida   | 1   | S            |              | Cada estante identifica-se pelo seu nome, dado pelo utilizador. Para a gerir, é necessário encontrar-se na página dela.  |
|  | 2   | S            |              | A barra lateral inclui todas as estantes do utilizador permanentemente, com cada nome a servir de uma ligação para a sua página.   |
|  | 3   | S            |              | Cada nome na barra lateral é clicável e tal é expresso pelo alterar da sua cor e sublinhado.   |
|  | 4   | S            |              | A resposta à ação é imediata: o utilizador navega para a página da estante   |
| 2<br><br>O Utilizador altera a visibilidade da estante | 1   | S            |              | O Utilizador selecionou a estante para este mesmo propósito. Agora que está selecionada basta apenas alterar a visibilidade.   |
|  | 2   | S            |              | O botão da visibilidade da estante encontra-se imediatamente em cima da mesma, com um botão dropdown para as possibilidades.   |
|  | 3   | S            |              | Tanto o título do botão que indica que se trata da visibilidade da estante como o facto do botão indicar visualmente que se trata de um dropdown tal como também apresenta a visibilidade corrente da estante. |
|  | 4   | S            |              | Uma caixa de confirmação é apresentada ao utilizador para garantir a pretenção para a alteração da visibilidade da mesma.  |



|  |                             |              |              |  |
|--|-----------------------------|--------------|--------------|--|
| Tarefa:  | Escrever Review de um Livro |              |              |  |
| Utiliza.:  | Novato no Sistema           |              |              |  |
| Passos   | Questões                    | OK?<br>(S/N) | Risco<br>(3) | 1Problema/Sugestão/Justificação  |
| 1<br><br>O Utilizador<br>accede à barra de<br>pesquisa   | 1                           | S            |              | Os Livros identificam-se, para o utilizador, pelo seu título. Por isso, é necessário introduzir os dados num campo que possa incluir texto, como uma barra de pesquisa.  |
|  | 2                           | S            |              | A barra de pesquisa encontra-se estática no topo do ecrã, mesmo se a página for deslizada, então sempre visível  |
|  | 3                           | S            |              | A barra de pesquisa indica que os livros devem ser pesquisados em texto nela escrito.  |
|  | 4                           | S            |              | A resposta à ação é imediata: a barra apresenta um marcador de escrita à espera do input de texto.   |
| 2<br><br>O Utilizador<br>digita o título do<br>livro pretendido  | 1                           | S            |              | A partir do momento que um ponteiro de escrita é demonstrado, o próximo passo é digitar qual o título.   |
|  | 2                           | S            |              | A barra de pesquisa encontra-se estática no topo do ecrã, mesmo se a página for deslizada, então sempre visível  |
|  | 3                           | S            |              | A partir do momento que um ponteiro de escrita é demonstrado, o próximo passo é digitar qual o título.   |
|  | 4                           | S            |              | Os livros são apresentados imediatamente após a escrita do título.   |
| 3<br><br>O Utilizador<br>seleciona o li- vro<br>que pretende   | 1                           | S            |              | O livro é representado textualmente pelo título mas o utilizador ainda precisa de navegar para a página específica do livro, apresentada depois de escrever, o que torna este seguimento intuitivo.                        |
|  | 2                           | S            |              | Os livros aparecem em dropdown a partir da barra de pesquisa a sobrepor-se aos restantes elementos da página.  |
|  | 3                           | S            |              | Sobrepôr o rato a qualquer destes elementos a representar os livros sobressai-os alterando a cor do fundo, o que indica que o elemento é clicável e que representa o livro em questão.                                     |
|  | 4                           | S            |              | O utilizador navega imediatamente para a página do livro selecionado   |
| 4<br><br>O Utilizador<br>escreve uma<br>review para o livro<br>(com ou sem<br>avaliação<br>quantitativa) | 1                           |              |              | Lendo a atividade do livro, o utilizador encontra-se na aba de reviews, que também apresenta nos detalhes do livro a sua avaliação total. Nesta página, então, o utilizador conclui que pode escrever também a sua review. |
|  | 2                           |              |              | A caixa de texto (tal como as estrelas quantitativas) encontram-se imediatamente abaixo da separação da aba mas antes de todas as outras reviews do livro.   |
|  | 3                           |              |              | O botão de publicar review está devidamente identificado tal como a caixa de texto indica claramente que é nela que se deve escrever a review.   |
|  | 4                           |              |              | Depois de publicada, a review aparece permanente junto das restantes, após confirmação do utilizador.  |

### 3.3. Heuristic Evaluation

| Heurística   | Dificuldades  | Oportunidades de melhoria  |
|--|---|--|
| <p>Visibility of system status</p> <p>The system should always keep users informed about what is going on, through appropriate feedback within reasonable time</p>   | <p>1. Não é possível ver qual a ordem escolhida para os livros na biblioteca geral. 2. Não é possível ver quais as bookshelves a que um dado livro está atribuído na sua página particular. 3. Não é possível ver a própria avaliação junto das restantes para um dado livro.</p> | <p>1. Manter a escolha da ordenação sempre visível. (Inacabado) 2. Mostrar as estantes na página do livro, tendo em conta as classes de exclusividade (Inacabado) 3. Manter a avaliação do utilizador sempre no mesmo lugar, onde foi publicada originalmente.</p> |
| <p>Match between system and the real world</p> <p>The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow realworld conventions, making information appear in a natural and logical order.</p> | <p>1. Ordem pouco natural e lógica dos itens na barra de navegação.</p>   | <p>1. Reordenar para manter os itens do foco da aplicação de um lado e os itens específicos ao utilizador do outro. (Inacabado)</p>  |
| <p>User control and freedom</p> <p>Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.</p>  | <p>1. Impossível de recuperar de erroneamente pressionar o botão de Login/Signup. 2. Perda de uma avaliação feita por alguém que ainda não estava LoggedIn.</p>   | <p>1. Possibilitar voltar à interface inicial através de algum botão existente nestas duas páginas. 2. Manter tanto a rota como a avaliação para as recuperar depois do Login/Signup.</p>  |

|   |  |  |
|---|--|--|
| <p>Consistency and standards</p> <p>Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.</p>   | <p>1. Falta de visibilidade da interface inteira devido a problemas de resolução.</p>  | <p>1. Construir os componentes relativamente uns aos outros para conseguir adaptar-se a qualquer resolução de dispositivo. (Inacabado)</p>   |
| <p>Error prevention</p> <p>Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate errorprone conditions or check for them and present users with a confirmation option before they commit to the action.</p>                               | <p>1. A avaliação quantitativa está a ser imediatamente atribuída após a seleção das estrelas.</p>   | <p>1. Apenas atribuir a avaliação depois do pressionar da publicação, não imediatamente no pressionar das estrelas.</p>  |
| <p>Recognition rather than recall</p> <p>Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.</p> | <p>1. Os botões de navegação das bookshelves são indistinguíveis de texto natural.</p>   | <p>1. Tornar o texto sublinhado quando sobrevoado com o rato. (Inacabado)</p>  |
| <p>Flexibility and efficiency of use</p> <p>Accelerators unseen by the novice user may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.</p>   | <p>1. Moroso escolher uma estante para ser a destacada no perfil, ou para mudar a sua visibilidade. 2. Igualmente moroso alterar a privacidade e a exclusividade de várias bookshelves de uma vez. 3. Os detalhes de cada livro para um utilizador são apenas alteráveis na página do mesmo.</p> | <p>1. Botão de atalho junto da estante para a tornar destacada/mudar visibilidade. (Inacabado) 2. Introduzir uma página de edição das definições das estantes. (Inacabado) 3. Permitir alterar os detalhes pessoais de um livro nas estantes (Inacabado)</p> |

|   |  |  |
|---|--|--|
| <p>Aesthetic and minimalist design</p> <p>Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.</p>  | <p>1. Sobrecarga de informação pela presença permanente das barras laterais das estantes e dos livros a serem lidos.</p> | <p>1. Não demonstrar estas entradas em páginas em que a sua presença não é diretamente envolvida (definições, perfil, ...)</p> |
| <p>Help users recognize, diagnose, and recover from errors</p> <p>Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.</p>   | <p>1. Nem a página de Signup/Login demonstram erros de introdução de dados.</p>  | <p>1. Criar erros para todas as combinações, como campos em branco.</p>  |
| <p>Help and documentation</p> <p>Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.</p> | <p>1. Classes de exclusividade complexas de se compreender para novatos.</p>   | <p>1. Introduzir uma tooltip de ajuda na página de edição das classes de exclusividade (Inacabado).</p>                        |

### 3.4. Human-Centred Design

Nesta fase da avaliação, recorreram-se a utilizadores reais com experiência na plataforma em que nos baseamos, *Goodreads*. Com base em dois utilizadores distintos, efetuamos efetivamente passos de observação da utilização tanto da plataforma original como do nosso protótipo, para obter opiniões detalhadas sobre os diferentes mecanismos de usabilidade.

Por um lado, os pontos fulcrais notados na aplicação original notam a dificuldade de conseguir adicionar livros às diferentes estantes, tal como a edição prática e recorrente das mesmas, em locais pouco acessíveis em diferentes páginas. Para além disso, pretende-se a implementação de novas funcionalidades, nomeadamente mais uma privacidade, *friends-only*, mais uma estante padrão, *Did Not Finish* e o que é chamado de **Classes de Exclusividade**, onde duas estantes dentro de uma dada classe não podem partilhar nenhum livro. Os utilizadores demarcaram a sua insatisfação com o facto de apenas haver uma classe, e essa ser utilizada para os estados dos livros de leitura, quando poderia ser possível criar classes personalizadas com estantes que não as base. Por fim, também existem queixas da falta de possibilidade de fazer atualizações ao progresso dos livros que não estejam na estante "Currently Reading" e noutras interfaces que não a principal.

Destas considerações decidimos que deveriam existir várias classes de exclusividade possíveis, com uma página facilmente acessível de qualquer ponto da interface que permitisse editar mais do que uma estante de cada vez, em todas as características mencionadas. Também pretendemos que exista uma forma de atualizar o progresso na leitura dos livros dentro das estantes a que pertencem sem, recorrer à página inicial.

Por outro lado, a análise do protótipo final por parte dos utilizadores culminou na decisão de alterar a ordem da barra de navegação superior, por não ser tão intuitiva, o alterar de alguns dos títulos dados aos botões, por não serem sugestivos o suficiente, e a introdução de mais botões nas abas laterais que permeiam a interface inteira, para permitir fácil acesso mas sem sobrecarregar o utilizador por estar visivelmente separado do alvo principal de atenção. Também se introduziu o botão de "Move to Read" nos detalhes dos livros para permitir rápida mudança entre os estados de leitura e também conseguir introduzir uma data de leitura ao livro.

## 4. Tecnologias Backend

Para um rápido desenvolvimento do *backend*, optamos pela utilização da ferramenta *Spring Boot* que agiliza o desenvolvimento de aplicações *Spring*.

Para começar um projeto *Spring Boot* podemos optar por uma inicialização rápida através do *Spring Initializr* que nos permite escolher um conjunto de dependências essenciais para o desenvolvimento da nossa aplicação, nomeadamente um conjunto de *starters dependencies*, sendo elas:

- **Spring Boot Starter Web** - Desenvolver aplicações Web, no nosso caso estamos a usar *RESTful* com recurso ao *Spring MVC*;
- **Spring Boot Starter Cache** - Permite gestão de *Cache* com anotações específicas;
- **Spring Boot Starter Data Redis** - Permite integração com a base de dados **Redis**;
- **Spring Boot Starter Data JPA** - Adiciona suporte para **Java Persistence API** (JPA) e integra-a com o **Spring Boot**;
- **Spring Boot Starter Data REST** - Providencia um *RESTful frontend* para *Spring Data repositories*;
- **Spring Boot Starter Security** - Adiciona o **Spring Security**, oferecendo serviços de autenticação e autorização.
- **Spring Boot Starter Mail** - Tem como função simplificar os processo de configurar e usar a API do *JavaMail*, responsável para mandar *emails*.

## 5. Modelos

### Modelos de Persistência

Para o mapeamento de classes **Java** para tabelas da base de dados, recorreremos à *framework ORM* **Hibernate**, pelos seguintes motivos:

- Remove a necessidade de gerir transações;
- Evita ter que instanciar conexões, fazer as interrogações *sql* e fechar as conexões, ou seja, gerir as conexões e ter que escrever código *sql* explícito;
- Oferece uma maior transparência relativamente à base de dados relacional usada, o que facilita a migração para outra base de dados.

Além disso, usamos a especificação **JPA** e a extensão **Spring Data JPA** da *framework* **Spring**, a primeira oferece uma *interface* padronizada para o **Hibernate**, pois define uma especificação de como os dados devem ser persistidos e procurados; a segunda promove ainda mais a simplificação, ao reduzir o código *boilerplate* necessário para operações comuns como *CRUD* e paginações.

Inicialmente, definimos um diagrama **PIM** que mais tarde foi convertido para um **PSM Hibernate**, ou seja, tentamos seguir o processo *MDA*, no sentido em que começamos por definir diagramas abstratos independentes das tecnologias com o *core* dos componentes da lógica de negócio, e, depois, refinamos esses diagramas para uma tecnologia específica como o **Hibernate**. Não tiramos totalmente proveito da geração automática do código *xml* **Hibernate**, tomando como alternativa a escrita do mapeamento das entidades por anotações no código **Java** pela visibilidade que elas apresentam, uma vez que ficam adjacentes ao código, tendo como contrapartida o facto de “prendermos” o nosso código às anotações o que poderá dificultar uma transição de tecnologias, como, por exemplo, se em vez de usar **Hibernate**, decidíssemos voltar a usar **JDBC** diretamente. Portanto, a seguir aos diagramas, também explicaremos o propósito de cada anotação.

O diagrama **PIM** localiza-se nos anexos, sobre o **PSM**, o completo também está nos anexos, mas vamos falar de cada parte.

### Utilizador

Para a vertente do utilizador, isto é, *Customer* e *author*, persistimos muita informação sobre o mesmo uma vez que a nossa aplicação tem uma vertente social e por isso os requisitos relativamente às páginas de perfil eram diversos. Como um *author* é um utilizador da aplicação, mas com a capacidade de adicionar livros, ele herda diretamente do *customer*, detalhes mais técnicos, nomeadamente anotações são falados mais no final.

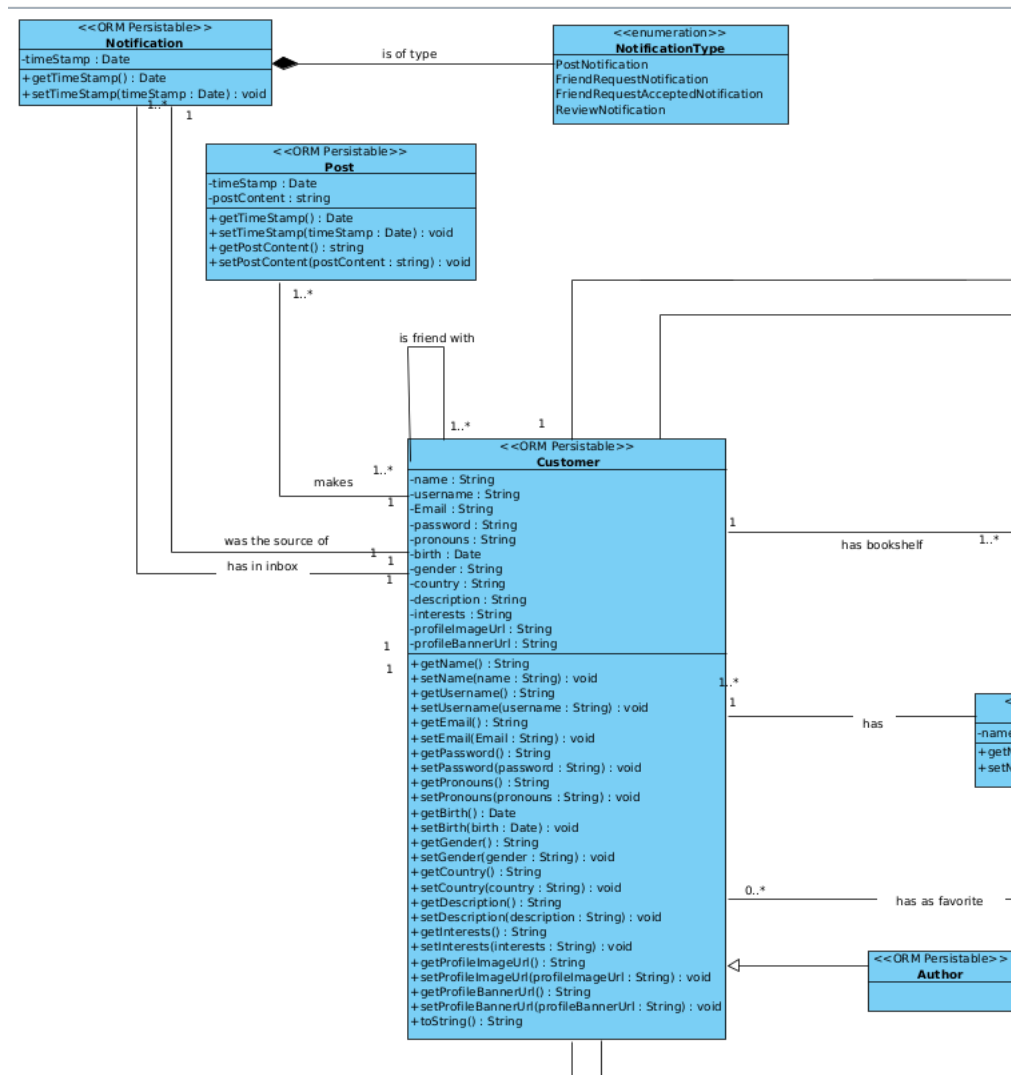
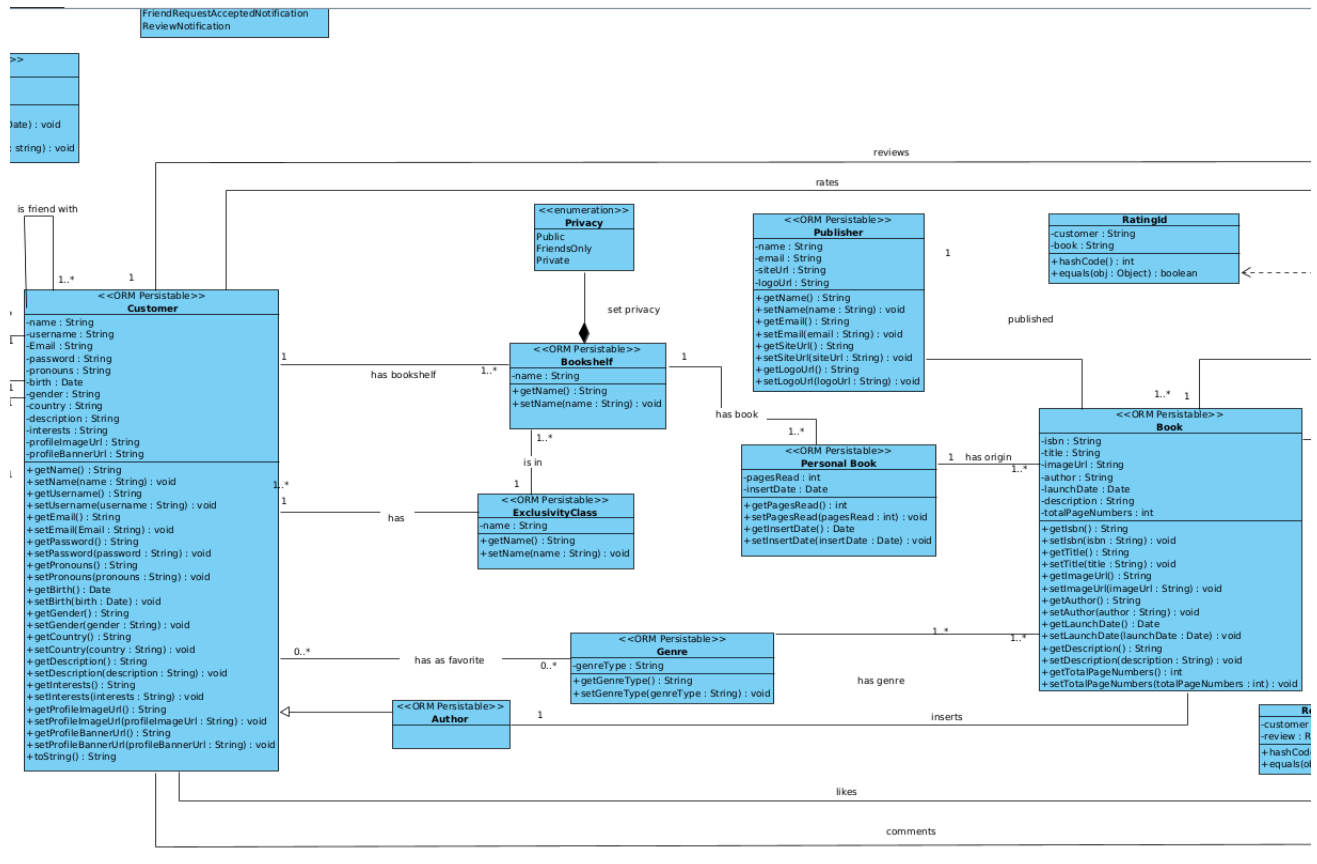


Figura 2: Vertente Utilizador PSM Hibernate

Além disso, o *customer* guarda um conjunto de posts que faz e notificações, para as notificações, se forem de 1 para 1 *customer*, guardamos diretamente, como, por exemplo, pedidos de amizade e confirmações de que foram aceites. No caso de serem para vários, como a notificação de que um amigo fez review de um livro, utilizamos o "padrão observer" oferecido pelo *spring* que utilizada a ideia de *publisher-subscriber*, que permite desacoplar preocupações e criar assincronia, dessa forma quando alguém faz uma review não precisa de ficar à espera que todos os seus amigos recebam a mensagem, isto é, seja persistida na base de dados, pois temos a funcionalidade de eventos e *listeners*.



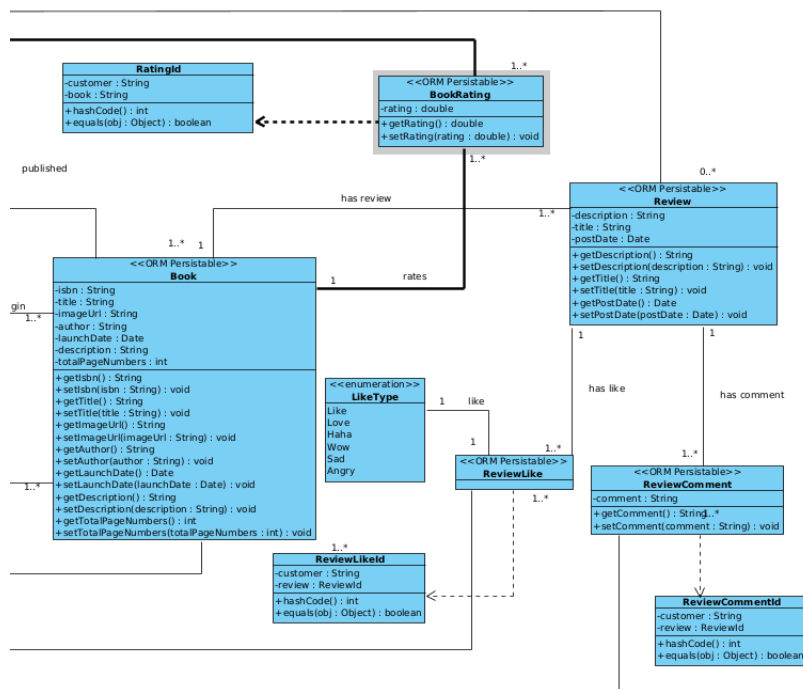
## Estante e Livros



**Figura 3:** Vertente Estante PSM Hibernate

De forma a atender a complexidade dos requisitos no que toca à estante, para além dos livros persistidos na base de dados, preciso da ideia de um livro pessoal que atua como contador de páginas lidas e que pode ser atualizado e exposto a outros utilizados se o tipo de privacidade da estante que o mesmo está for adequada, para níveis de privacidade temos: publico, privado e apenas-amigos. Além disso, é necessário guardar o tipo de estante, se ela tem classe de exclusividade ou não, ou seja, se aquela estante tiver classe, temos de implementar lógica para as outras estantes não partilharem o livro pessoal.

Para os livros, guardamos todos os livros criados pelos autores com quem faz *publish* do livro. Para além de ter as informações do livro que serão apresentadas, é necessário guardar reviews sobre esse livro, classificações, e comentários e likes das reviews.



**Figura 4:** Vertente Social do Livro PSM Hibernate

## Anotações

Para as anotações, em vez de destacar todas as anotações e a sua funcionalidade, vamos destacar as mais importantes e diferentes, uma vez que as restantes são repetitivas.

- **Inheritance(strategy = InheritanceType.SINGLE\_TABLE)** - Define a estratégia de herança onde todas as subclasses são mapeadas para uma única tabela, ou seja, apenas temos uma tabela Customer;
- **DiscriminatorColumn(name = "role", discriminatorType = DiscriminatorType.STRING)**  
- Especifica a role, serve para diferenciar as diferentes entidades na mesma tabela, principalmente porque partilham a tabela devido à estratégia de herança, essa diferenciação é marcada com **DiscriminatorValue("author")**;
- Temos relações muitos para muitos, um para muitos, muitos para um, **OneToMany(mappedBy = "customer", cascade = CascadeType.ALL, orphanRemoval = true)**, tomemos esta como exemplo, identificamos o proprietário da relação, **cascadeType.ALL** faz com que todas as operações de persistência sejam propagadas para outro elemento (no caso os elementos) da relação e **orphanRemoval** como **true** força entidades que não estejam ligadas a serem removidas.
- **Embedded** permite que separemos alguns atributos noutra classe para simplificar a gestão das classes.

## Modelos PSM Spring Boot

Para a parte da lógica de negócio, devido ao seu tamanho, decidimos partir em partes lógicas e não vamos referenciar diretamente as entidades, mas cada *repository* tem como dependência a entidade no início do seu nome, ou seja, um *BookRepository* tem como entidade *Book*. Os diagramas completos estão na pasta do *visual paradigm*, junto dos ficheiros com o código na pasta denominada VPP, o nome do ficheiro é "ClassDiagram".

### Diagrama PSM da Estante

Seguimos a divisão tradicional, temos uma camada *controller* que faz a gestão das rotas, não implementa lógica de negócio, para mover dados, usamos o *design pattern* DTOs, desse modo conseguimos mover uma menor quantidade de dados e desacoplamos a entidades da informação que é preciso enviar, além disso, recorremos a mappers criados por nós, existem bibliotecas que fazem o mapping, mas tomamos como preferência ter o controlo total do *mapping*, que desacopla ainda mais os dois, pois não fazemos a conversão diretamente em nenhuma das classes. Os serviços implementam a lógica de negócio e para acessar objetos persistidos recorremos aos *repositories* que abstraem a lógica de usar *DAOs* e simplificam o processo de gestão, criação e eliminação de objetos persistidos. Além disso, também tiramos proveito das exceções do Java para conseguir diferenciar o melhor possível os diferentes possíveis erros da nossa aplicação e conseguir usá-los como forma de mensagem para a camada de apresentação.

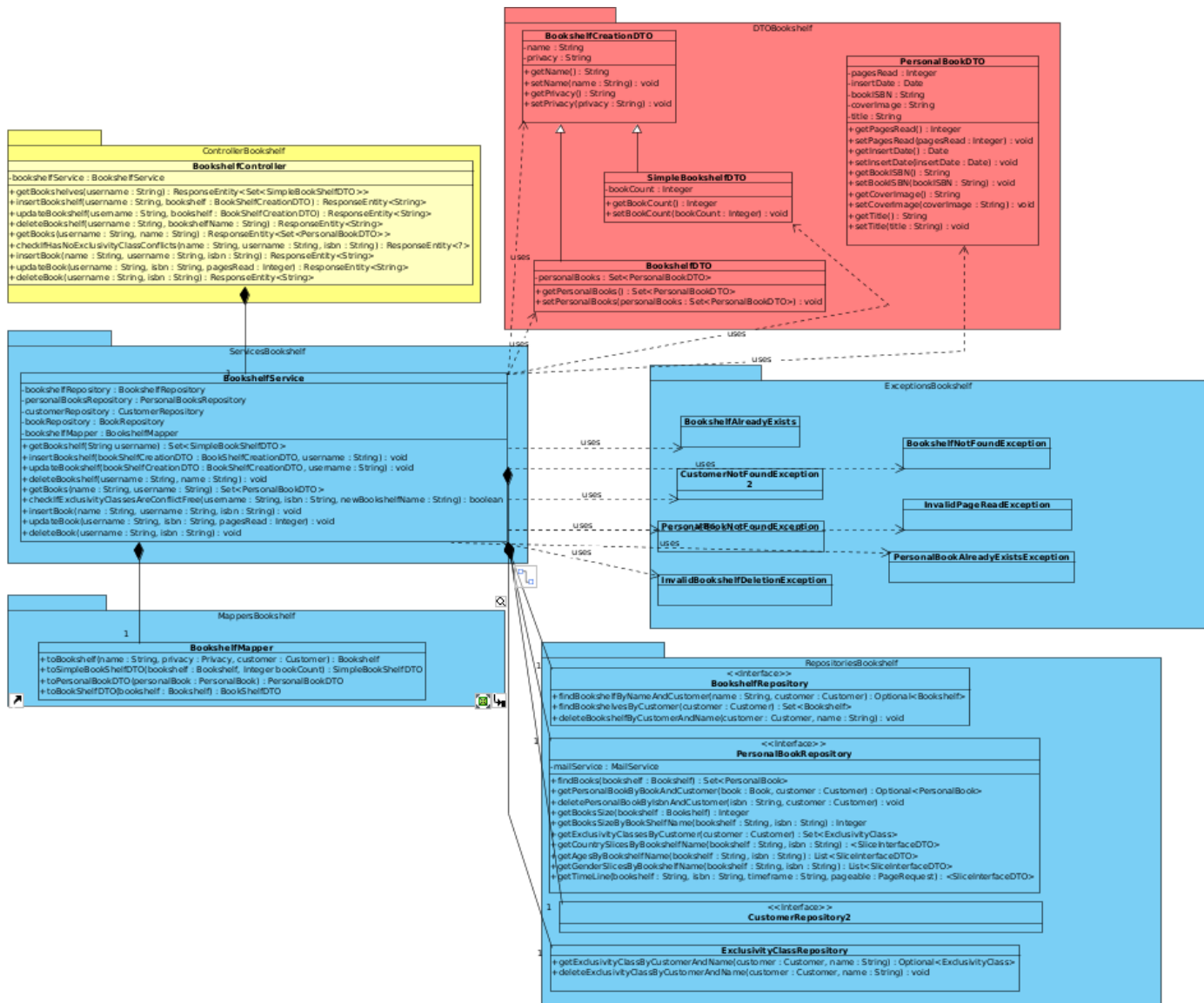


Figura 5: PSM completo da estante

## Diagrama PSM Customer

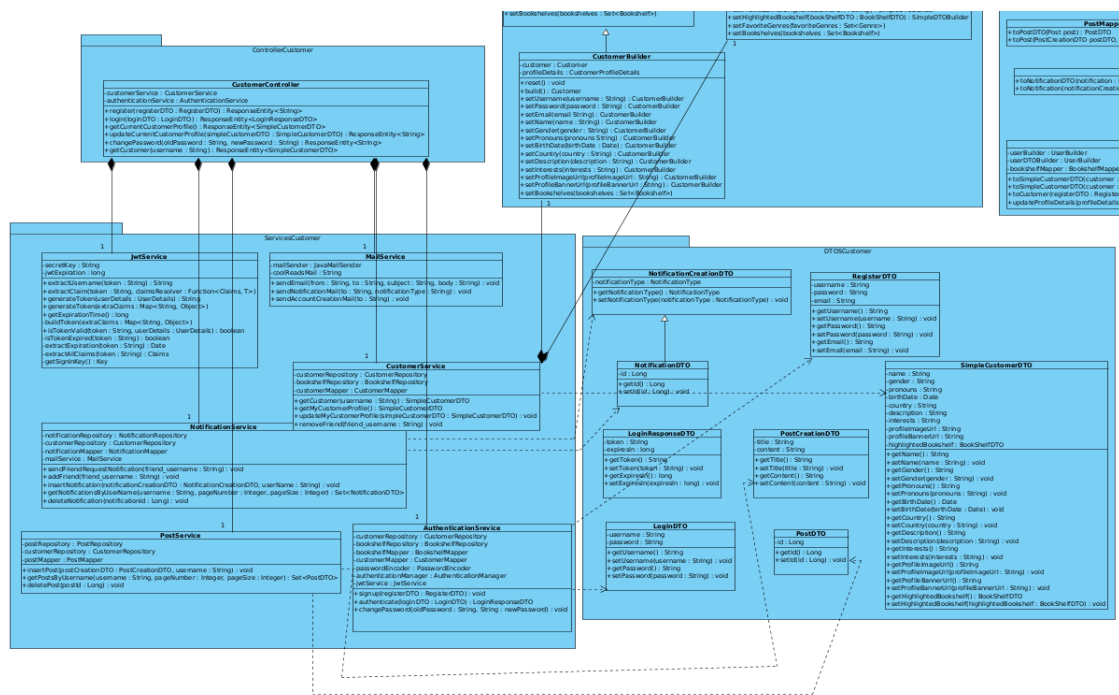


Figura 6: PSM Customer Parte 1

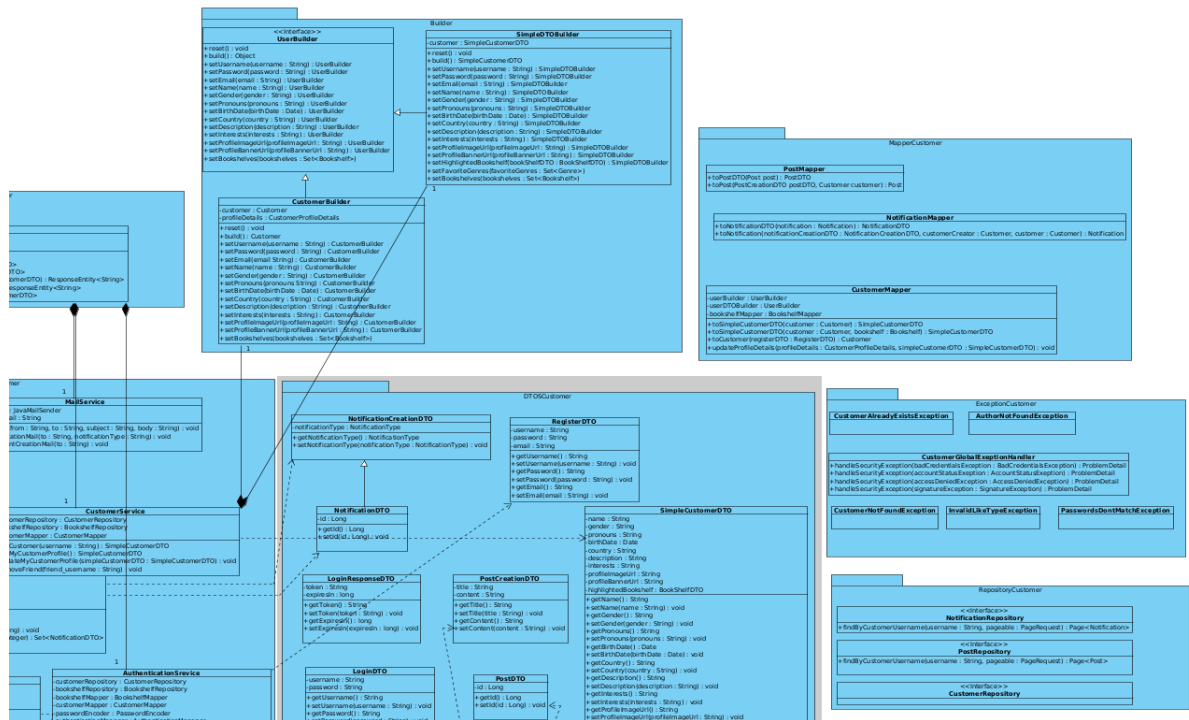


Figura 7: PSM Customer Parte 2

## PSM Publisher

O *Publisher*, pela sua simplicidade, apenas tem a operação de *insert*, seguindo a mesmas regras definidas nos diagramas anteriores.

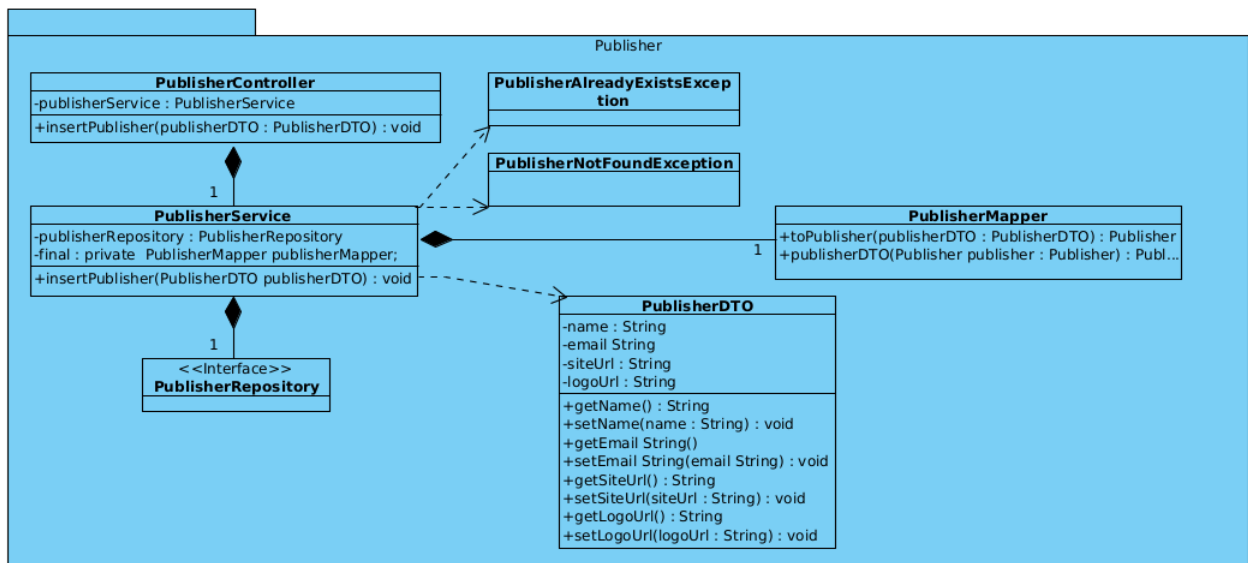


Figura 8: PSM Publisher

## PSM Book Review

Para a review, seguimos o padrão, é só importante referir que a classe `ReviewMessage` serve como contendor para o evento das notificações de review, desse modo, podemos usar o padrão publish-subscribe para notificar os amigos do utilizador.

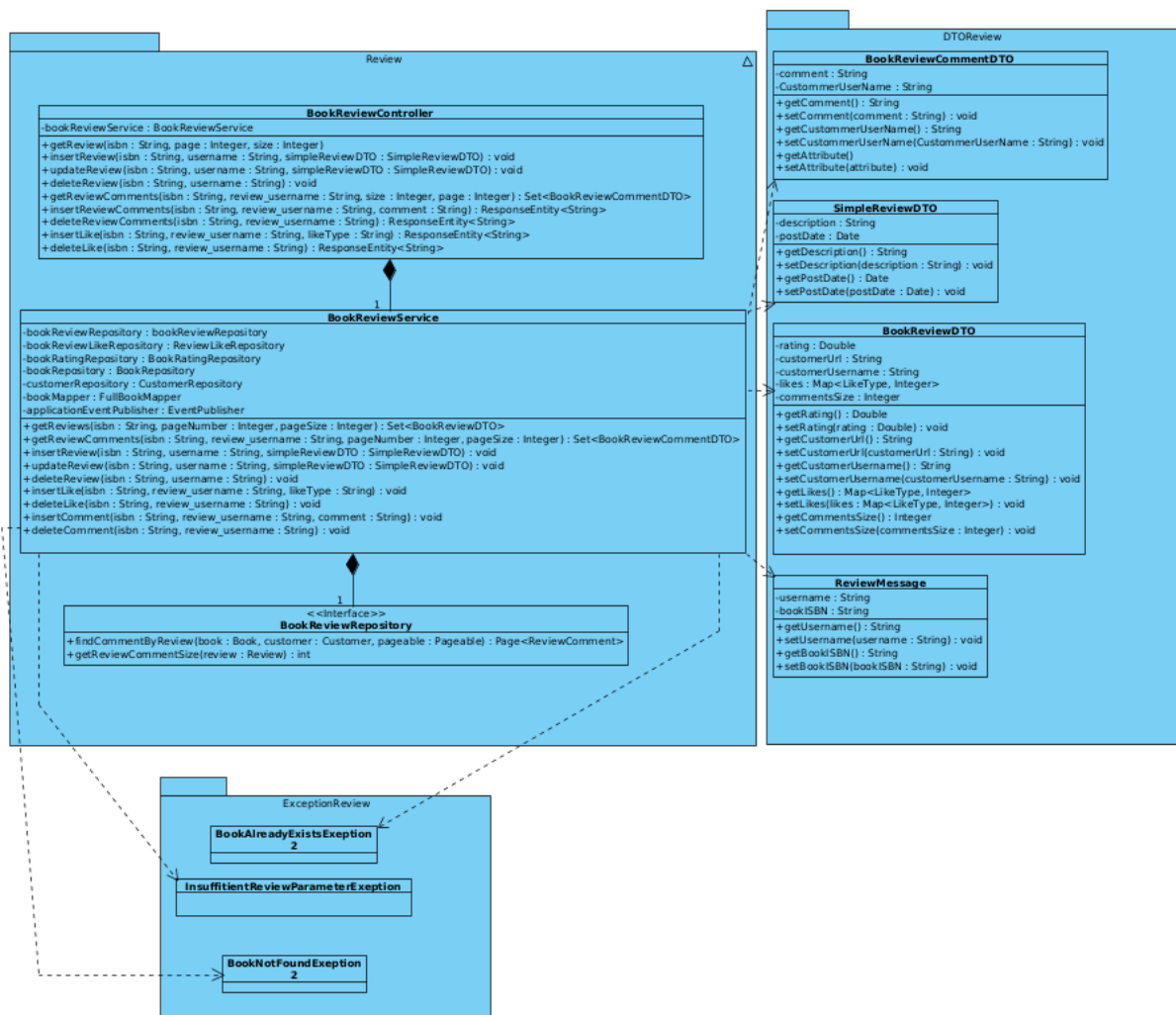


Figura 9: PSM Review

## PSM Book

Para o livro, é importante referir que para além do mapeamento feito entre os dtos, fazemos uso de cache com o *redis* uma vez que é um ponto crítico da aplicação.

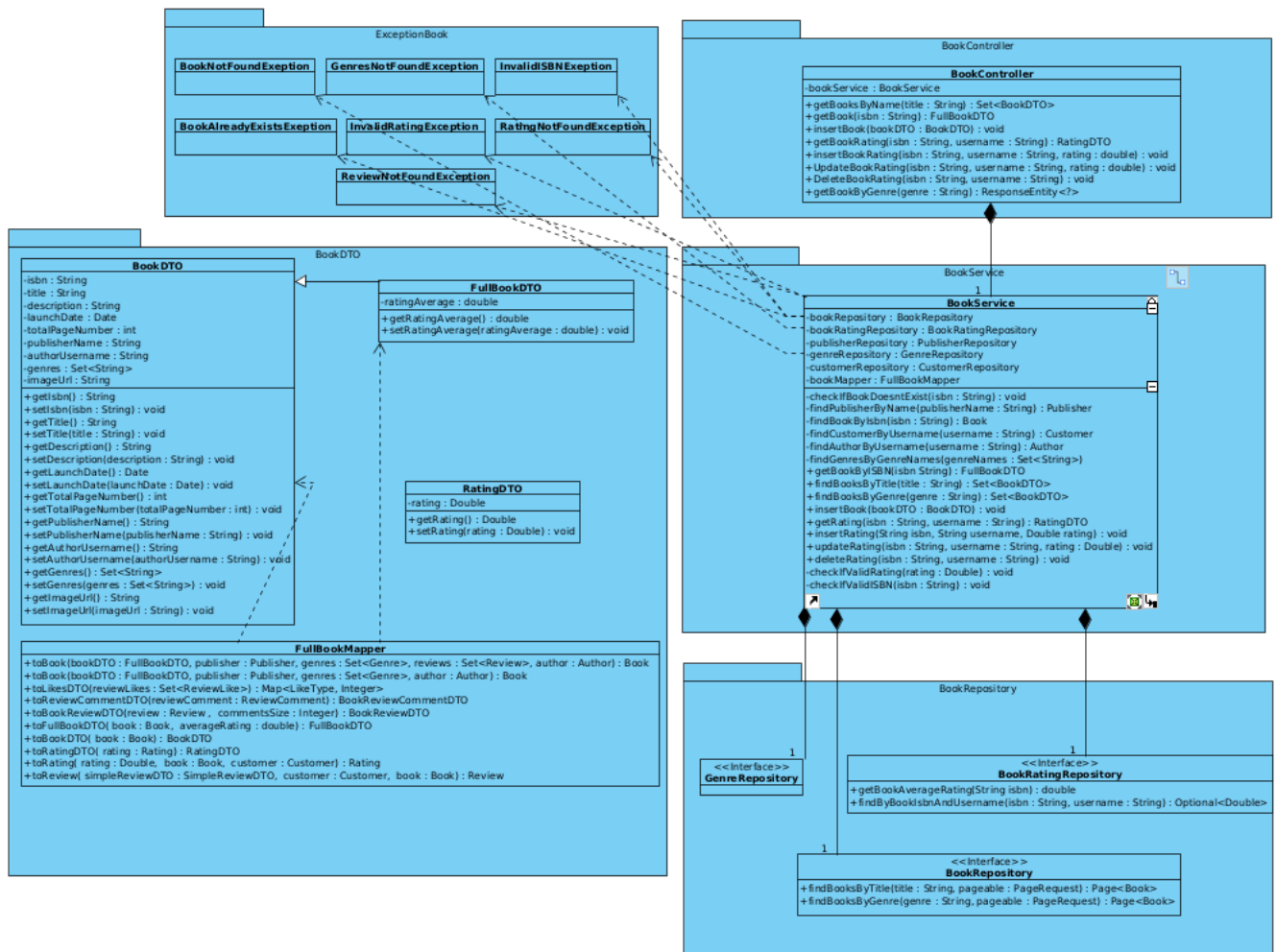


Figura 10: PSM Livro

## PSM das Estatísticas

Para as estatísticas é importante referir as diferenças de comportamento entre as categorias do gráfico circular, além de que esta diferença foi implementada utilizando o *Strategy pattern*, as categorias país e género funcionam de forma semelhante realizando um **COUNT(\*)**, no entanto, a categoria idade necessita que seja feito *binning* pelas idades e que estas sejam comparadas com as categorias definidas numa tabela de configuração, para tal teve que ser usada uma *query* nativa do *Postgresql*.



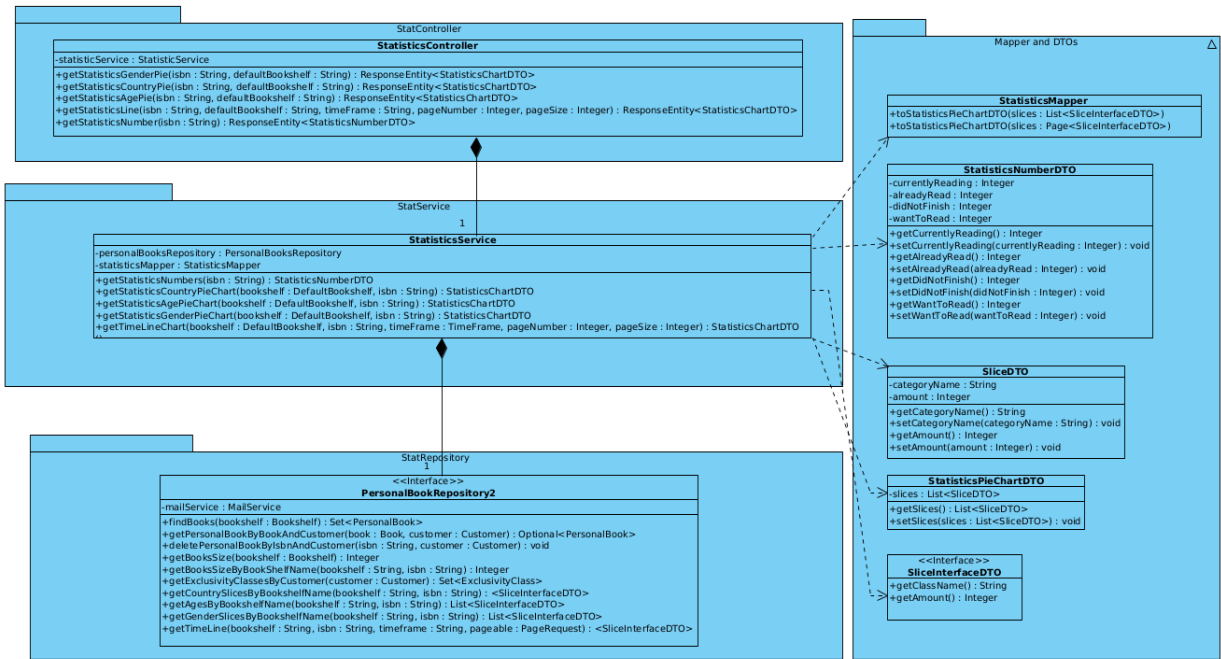


Figura 11: PSM Estadísticas

## 6. Documentação da API BackEnd

Entrando na esfera da implementação, decidimos elaborar uma documentação simples para a API do *backend*. Isto proporcionará um ponto de partida claro e fundamentado para os desenvolvedores, estabelecendo as operações *HTTP* essenciais de *POST*, *GET*, *PUT* e *DELETE*. Tendo em conta as diferentes vertentes da nossa aplicação, documentamos as possíveis ações em partes lógicas:

## Gestão de Livros

| Método | Rota  | Descrição   | Autorização |
|--------|---|---|-------------|
| POST   | /api/book                                   | Inserir um livro no Sistema                         | Author      |
| GET    | /api/book/name?title                        | Fazer pesquisa por livros por um certo título       | Customer    |
| GET    | /api/book/<isbn>                            | Buscar um livro pelo seu identificador único (ISBN) | Customer    |
| GET    | /api/book/name?title                        | Fazer pesquisa por livros por um certo título       | Customer    |
| GET    | /api/book/name?title                        | Fazer pesquisa por livros por um certo título       | Customer    |
| GET    | /api/book/name?title                        | Fazer pesquisa por livros por um certo título       | Customer    |
| GET    | /api/book/genre?genre?page?size             | Retorna livros por <i>genre</i>                     | Customer    |
| GET    | /api/book/<isbn>                            | Retorna a informação de um livro <i>genre</i>       | Customer    |
| POST   | /api/book                                   | Insere um livro no sistema                          | Author      |
| GET    | /api/book/<isbn>/rate?username              | Retorna a avaliação de um livro de um customer      | Customer    |
| POST   | /api/book/<isbn>/rate?username              | Insere uma avaliação de um livro                    | Customer    |
| PATCH  | /api/book/<isbn>/rate?username              | Atualiza uma avaliação de um livro                  | Customer    |
| DELETE | /api/book/<isbn>/rate?username              | Elimina uma avaliação de um livro                   | Customer    |
| GET    | /api/book/<isbn>/review?username            | Retorna as reviews sobre um livro                   | Customer    |
| POST   | /api/book/<isbn>/review?username            | Insere uma review relativa a um livro               | Customer    |
| PATCH  | /api/book/<isbn>/review?username            | Atualiza uma review relativa a um livro             | Customer    |
| DELETE | /api/book/<isbn>/review?username            | Remove uma review relativa a um livro               | Customer    |
| GET    | /api/book/<isbn>/review/username/<username> | Buscar a review de um utilizador sobre um livro     | Customer    |

|        |  |   |          |
|--------|--|---|----------|
| GET    | /api/book/<isbn>/review<br>/<review_username>/comment?page?size                | Retorna os comentários feitos sobre uma <i>review</i>                             | Customer |
| POST   | /api/book/<isbn>/review/<review_username>/comment                              | Insere um comentário sobre uma <i>review</i>                                      | Customer |
| DELETE | /api/book/<isbn>/review/<review_username>/comment                              | Elimina um comentário sobre uma <i>review</i>                                     | Customer |
| POST   | /api/book/<isbn>/review/<br><review_username>/like?likeType                    | Insere uma reação sobre uma <i>review</i>   | Customer |
| DELETE | /api/book/<isbn>/review/<review_username>/like                                 | Remove uma reação sobre uma <i>review</i>   | Customer |
| GET    | /api/book/<isbn>/stats/pie/gender?defaultBookshelf                             | Buscar a descrição de um livro numa <i>bookshelf</i> por gênero                   | None     |
| GET    | /api/book/<isbn>/stats/pie/age?defaultBookshelf                                | Buscar a descrição de um livro numa <i>bookshelf</i> por idade                    | None     |
| GET    | /api/book/<isbn>/stats/pie/country?defaultBookshelf                            | Buscar a descrição de um livro numa <i>bookshelf</i> por país                     | None     |
| GET    | /api/book/<isbn>/stats/line<br>?defaultBookshelf?timeFrame?pageNumber?pageSize | Buscar as adições de um livro a uma <i>bookshelf</i> ao longo do tempo            | None     |
| GET    | /api/book/<isbn>/stats/numbers   | Buscar o número total de vezes que um livro aparece nas <i>bookshelf's</i> padrão | None     |

## Gestão de Utilizadores

| Método | Rota   | Descrição  | Autorização |
|--------|--|--|-------------|
| POST   | /api/customer/register?isAuthor                        | Regista um utilizador no sistema   | None        |
| POST   | /api/customer/login                                    | Verifica as credenciais de um utilizador                                 | None        |
| GET    | /api/customer/me                                       | Retorna os detalhes do utilizador que realiza o pedido                   | Customer    |
| PUT    | /api/customer/me                                       | Atualiza os detalhes do utilizador que realiza o pedido                  | Customer    |
| DELETE | /api/customer/me                                       | Remove um utilizador que realiza o pedido                                | Customer    |
| PUT    | /api/customer/me/password?newPassword                  | Modifica a <i>password</i> do utilizador que efetua o pedido             | Customer    |
| GET    | /api/customer/username/<username>                      | Retorna os detalhes de um utilizador                                     | Customer    |
| GET    | /api/customer/me/friends                               | Retorna os amigos do utilizador que realizou o pedido                    | Customer    |
| POST   | /api/customer/me/friends?friendUsername                | Envia um pedido de amizade aos amigos do utilizador que efetuou o pedido | Customer    |
| PUT    | /api/customer/me/friends?friendUsername                | Aceita o pedido de amigo de um utilizador                                | Customer    |
| DELETE | /api/customer/me/friends?friendUsername                | Remove um utilizador da lista de amigos                                  | Customer    |
| GET    | /api/customer/me/notificationa<br>?pageNumber?pageSize | Retorna as notificações de um utilizador ordenadas por data de inserção  | Customer    |
| DELETE | /api/customer/me/notificationa<br>?notificationId      | Remove uma notificação de um utilizador                                  | Customer    |
| GET    | /api/customer/<username>/bookshelf                     | Retorna as <i>bookshelf's</i> de um utilizador                           | Customer    |
| POST   | /api/customer/<username>/bookshelf                     | Insere uma <i>bookshelf's</i> no sistema                                 | Customer    |
| PUT    | /api/customer/<username>/bookshelf                     | Atualiza uma <i>bookshelf's</i> no sistema                               | Customer    |

|        |  |  |          |
|--------|--|--|----------|
| DELETE | /api/customer/<username>/bookshelf                         | Remove uma <i>bookshelf</i> 's do sistema  | Customer |
| GET    | /api/customer/<username>/bookshelf/<name>                  | Retorna os livros de uma <i>bookshelf</i>  | Customer |
| PUT    | /api/customer/<username>/bookshelf/<name>/exclusivityClass | Associa uma classe de exclusividade a uma <i>bookshelf</i>   | Customer |
| DELETE | /api/customer/<username>/bookshelf/<name>/exclusivityClass | Remove a associação de uma classe de exclusividade a uma <i>bookshelf</i>                                    | Customer |
| GET    | /api/customer/<username>/bookshelf/<name>/checkConflicts   | Verifica se um livro pode ser adicionado a uma <i>bookshelf</i> tendo em conta a sua classe de exclusividade | Customer |
| POST   | /api/customer/<username>/bookshelf/<name>                  | Inserir um livro numa <i>bookshelf</i>   | Customer |
| PUT    | /api/customer/<username>/bookshelf/personalBook            | Atualiza as páginas lidas de um livro  | Customer |
| DELETE | /api/customer/<username>/bookshelf/personalBook            | Atualiza as páginas lidas de um livro  | Customer |
| GET    | /api/customer/<username>/bookshelf/exclusivityClass        | Retorna a classe de exclusividade de uma <i>bookshelf</i>  | Customer |
| POST   | /api/customer/<username>/bookshelf/exclusivityClass        | Cria uma classe de exclusividade   | Customer |
| PUT    | /api/customer/<username>/bookshelf/exclusivityClass/<name> | Altera o nome de uma classe de exclusividade   | Customer |
| DELETE | /api/customer/<username>/bookshelf/exclusivityClass        | Elimina uma classe de exclusividade  | Customer |

## Gestão de *Publishers*

| Método | Rota           | Descrição                  | Autorização |
|--------|----------------|----------------------------|-------------|
| POST   | /api/publisher | Insere um <i>publisher</i> | Author      |

Notas:

- Para mais informação detalhada sobre a documentação da API, nomeadamente o que cada *request* precisa de enviar no *body*, criamos um swagger: <https://app.swaggerhub.com/apis/CMACHAD00032/CoolReads/v0>, para além de te-lo nos ficheiros do backend (caso seja preciso correr localmente).

## 7. Testes e *Deployment*

### Testes

Um fator que ajudou com a construção da nossa aplicação foi a adição de testes ao mesmo tempo que desenvolvíamos, desse modo, conseguimos detetar *bugs* mais facilmente e fortificamos a nossa *app* para num caso de *refactor*, conseguíamos assegurar com mais certeza que as funcionalidades não foram perdidas, ou novos *bugs* foram introduzidos. Desse modo, embora não tenhamos seguido a *approach* **Test-Driven Development**, no sentido em que escrevíamos os testes antes, e a escrita do código era minimizada ao máximo para passar nesses testes, tentamos adotar a ideia de incorporar testes com o desenvolvimento. Por isso, a nossa metodologia aproxima-se à **Shift-left testing**, no sentido em que puxamos os testes o mais para a esquerda possível, isto é, trazemos os testes mais perto do desenvolvimento, permitindo, como dito anteriormente, testes o mais cedo possível e automatizá-los para terem uma presença contínua no desenvolvimento da aplicação. É importante referir que não fazemos testes na vertente de segurança, o que seria também uma mais-valia para o projeto, pois da mesma maneira que trazer testes para o desenvolvimento evita um maior risco para aplicação, pois só são feitos no fim e por isso os requisitos poderão estar mal implementados, a vertente de segurança também poderá trazer complicações para o projeto, pois poderá também implicar mudanças na arquitetura que podiam ter sido detetadas mais cedo.

Para efetuar os testes, utilizamos tanto o **Postman** para testes mais simples e menos automatizados e para automatizar os testes e fazê-los de forma mais contínua utilizamos o **JUnit**.

### Deployment

Para o *deployment* da aplicação, decidimos utilizar sistema de orquestração de *containers* **kubernetes** com o Google cloud, assim tiramos partido do facto de termos uma infraestrutura dada (*IaaS*) invés de sermos nós a montar o *cluster* e arcar com o peso de instalar e manter a infraestrutura.

Recorremos à ferramenta **Ansible** para o *provisioning* e *deployment* de cada uma das componentes da nossa aplicação. Ao automatizar o processo com o **Ansible**, ganhamos diversas vantagens como evitar o processo repetitivo de fazer o aprovisionamento e *deployment* da *app*, aumentar a heterogeneidade entre vários sistemas e evitamos o fator humano de fazer iterativamente o processo.

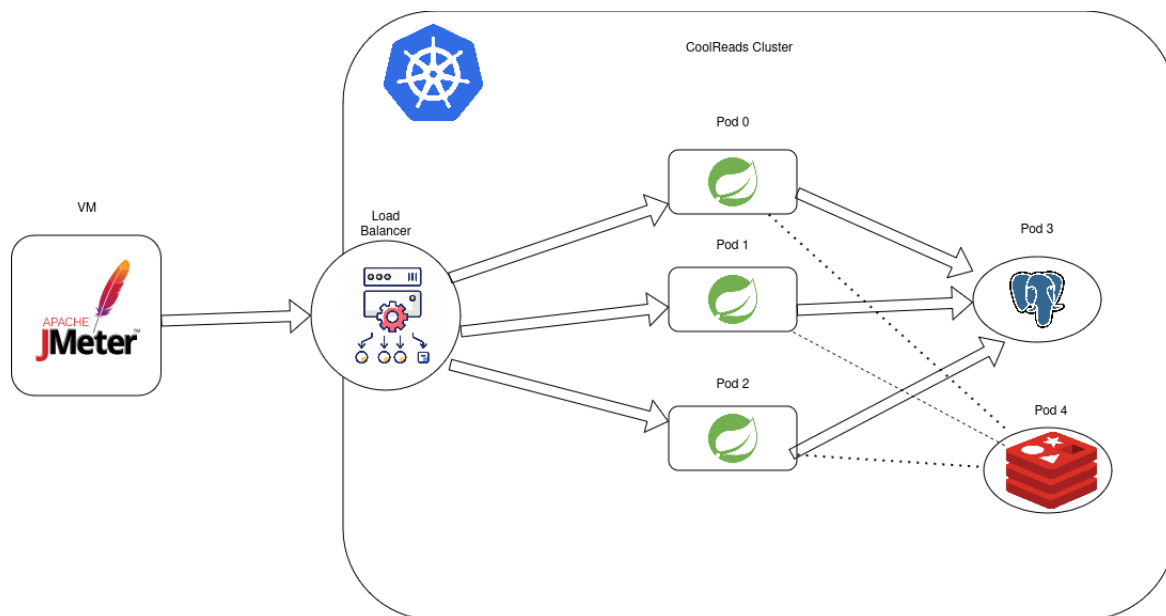
### Testes de Carga

Para entender como a nossa aplicação lida com certas situações, resolvemos realizar testes de carga com a ferramenta **Jmeter** que com algumas configurações, consegue simular diversos clientes a fazer vários pedidos, com isso conseguimos testar a aplicação com carga e entender em que ponto a desempenho da aplicação diminui.

De forma a aproximar uma utilização de um utilizador, tentamos fazer testes numa *pipeline* ações que utilizadores fariam como registar/fazer login, ver uma página de um livro, fazer uma análise



do livro e/ou adicionar o livro a uma estante. Para uma maior credibilidade, seria preciso fazer testes que stresssem a aplicação em outras situações e, além disso, tentar replicar ainda mais a utilização "normal" de um utilizador, pondo possivelmente a aplicação à prova para um conjunto menor de utilizadores, ou até se possível, usar um *trace* real de uma aplicação semelhante. Como setup, decidimos não replicar a Base de dados nem a Cache. Replicamos apenas a aplicação em 3 réplicas onde a carga é balanceada com o load balancer oferecido pela google cloud. Portanto, o setup será:



**Figura 12:** Setup Google Cloud

## Testes de carga para um

| Label        | # Samples | Average | Min | Max  | Std. Dev. | Error % | Throughput | Received KB/... | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|-----|------|-----------|---------|------------|-----------------|-------------|------------|
| HTTP Request | 100       | 2042    | 247 | 3476 | 859.01    | 0.00%   | 23.2/sec   | 10.12           | 6.67        | 446.0      |
| TOTAL        | 100       | 2042    | 247 | 3476 | 859.01    | 0.00%   | 23.2/sec   | 10.12           | 6.67        | 446.0      |

**Figura 13:** Registo para 100 clientes

| Label        | # Samples | Average | Min | Max   | Std. Dev. | Error % | Throughput | Received KB/... | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|-----|-------|-----------|---------|------------|-----------------|-------------|------------|
| HTTP Request | 1000      | 18241   | 480 | 32457 | 8740.66   | 0.00%   | 26.1/sec   | 11.37           | 7.49        | 446.0      |
| TOTAL        | 1000      | 18241   | 480 | 32457 | 8740.66   | 0.00%   | 26.1/sec   | 11.37           | 7.49        | 446.0      |

**Figura 14:** Registo para 1000 clientes

| Label          | # Samples | Average | Min | Max  | Std. Dev. | Error % | Throughput | Received KB/... | Sent KB/sec | Avg. Bytes |
|----------------|-----------|---------|-----|------|-----------|---------|------------|-----------------|-------------|------------|
| Register       | 100       | 1779    | 337 | 5455 | 1116.32   | 0.00%   | 16.6/sec   | 7.22            | 7.29        | 446.0      |
| Login          | 100       | 2316    | 620 | 4767 | 941.05    | 0.00%   | 15.4/sec   | 9.37            | 6.01        | 623.0      |
| Get Book       | 100       | 1069    | 6   | 3692 | 740.67    | 0.00%   | 17.4/sec   | 12.54           | 5.53        | 739.0      |
| Insert Review  | 100       | 642     | 27  | 1686 | 377.12    | 0.00%   | 33.2/sec   | 12.32           | 16.53       | 380.0      |
| Create Book... | 100       | 228     | 10  | 936  | 213.91    | 0.00%   | 50.8/sec   | 18.84           | 21.61       | 380.0      |
| TOTAL          | 500       | 1207    | 6   | 5455 | 1069.70   | 0.00%   | 60.1/sec   | 30.15           | 24.91       | 513.6      |

**Figura 15:** 100 clientes

| Label          | # Samples | Average | Min  | Max   | Std. Dev. | Error % | Throughput | Received KB/... | Sent KB/sec | Avg. Bytes |
|----------------|-----------|---------|------|-------|-----------|---------|------------|-----------------|-------------|------------|
| Register       | 200       | 3879    | 365  | 10966 | 2460.87   | 0.00%   | 17.0/sec   | 7.41            | 7.48        | 446.0      |
| Login          | 200       | 5035    | 1740 | 11457 | 2033.30   | 0.00%   | 14.2/sec   | 8.63            | 5.54        | 623.0      |
| Get Book       | 200       | 2373    | 8    | 7238  | 1292.98   | 0.00%   | 16.9/sec   | 12.17           | 5.37        | 739.0      |
| Insert Review  | 200       | 1748    | 101  | 4304  | 969.85    | 0.00%   | 27.3/sec   | 10.12           | 13.58       | 380.0      |
| Create Book... | 200       | 749     | 11   | 2527  | 568.03    | 0.00%   | 40.3/sec   | 14.96           | 17.17       | 380.0      |
| TOTAL          | 1000      | 2757    | 8    | 11457 | 2225.88   | 0.00%   | 60.4/sec   | 30.31           | 25.05       | 513.6      |

**Figura 16:** 200 clientes

| Label          | # Samples | Average | Min  | Max   | Std. Dev. | Error % | Throughput | Received KB/... | Sent KB/sec | Avg. Bytes |
|----------------|-----------|---------|------|-------|-----------|---------|------------|-----------------|-------------|------------|
| Register       | 500       | 8461    | 491  | 22158 | 4794.93   | 0.00%   | 21.6/sec   | 9.41            | 9.49        | 446.0      |
| Login          | 500       | 14930   | 9167 | 24241 | 2373.37   | 0.00%   | 15.5/sec   | 9.45            | 6.07        | 623.0      |
| Get Book       | 500       | 7357    | 1919 | 14849 | 3073.50   | 0.00%   | 18.2/sec   | 13.13           | 5.79        | 739.0      |
| Insert Review  | 500       | 3931    | 1534 | 7604  | 989.36    | 0.00%   | 32.0/sec   | 11.88           | 15.94       | 380.0      |
| Create Book... | 500       | 2937    | 10   | 5485  | 1175.08   | 0.00%   | 57.3/sec   | 21.28           | 24.42       | 380.0      |
| TOTAL          | 2500      | 7523    | 10   | 24241 | 5101.59   | 0.00%   | 61.1/sec   | 30.64           | 25.32       | 513.6      |

**Figura 17:** 500 clientes

| Label          | # Samples | Average | Min   | Max   | Std. Dev. | Error % | Throughput | Received KB/... | Sent KB/sec | Avg. Bytes |
|----------------|-----------|---------|-------|-------|-----------|---------|------------|-----------------|-------------|------------|
| Register       | 1000      | 16510   | 627   | 39600 | 9144.73   | 0.00%   | 23.6/sec   | 10.30           | 10.39       | 445.9      |
| Login          | 1000      | 30896   | 22561 | 43613 | 2579.99   | 0.00%   | 15.4/sec   | 9.38            | 6.02        | 623.0      |
| Get Book       | 1000      | 15970   | 3279  | 30518 | 7320.47   | 0.00%   | 18.8/sec   | 13.57           | 5.98        | 739.0      |
| Insert Review  | 1000      | 7762    | 3417  | 13712 | 1994.57   | 0.00%   | 33.8/sec   | 12.56           | 16.85       | 380.0      |
| Create Book... | 1000      | 7220    | 12    | 13142 | 2337.57   | 0.00%   | 59.2/sec   | 21.98           | 25.22       | 380.0      |
| TOTAL          | 5000      | 15672   | 12    | 43613 | 10196.96  | 0.00%   | 60.5/sec   | 30.32           | 25.06       | 513.6      |

**Figura 18:** 1000 clientes

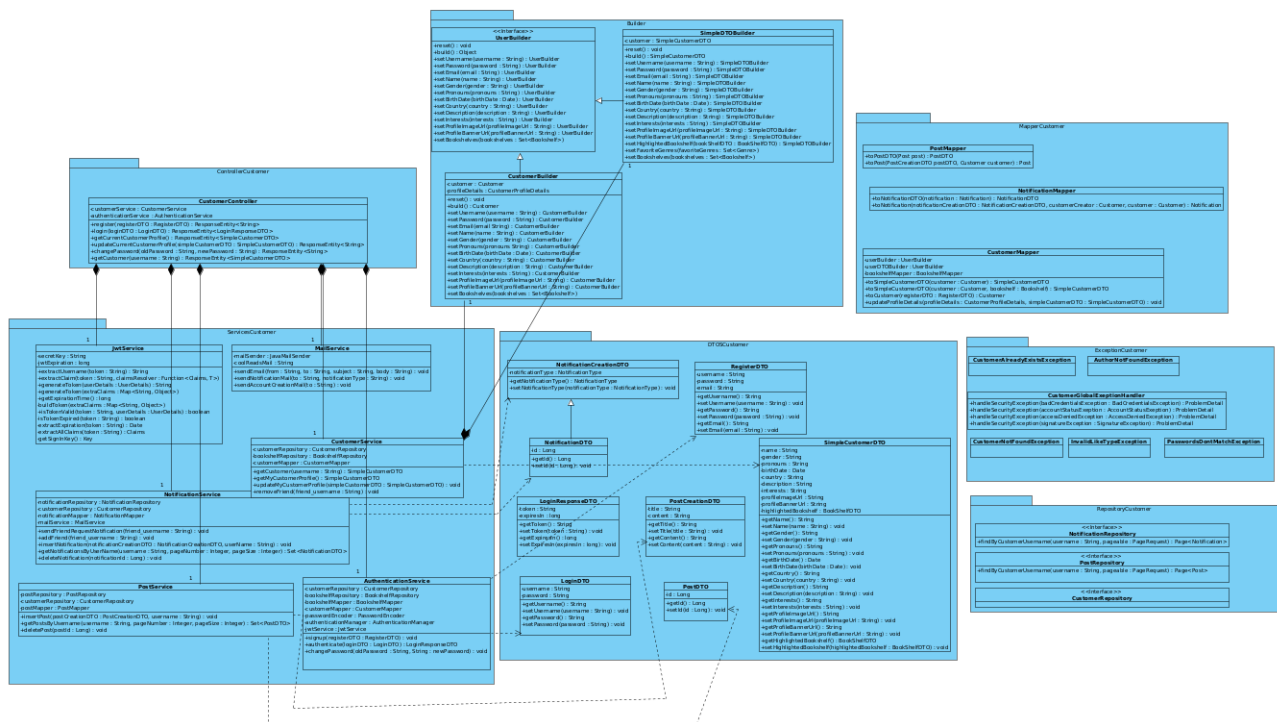
| Label          | # Samples | Average | Min   | Max   | Std. Dev. | Error % | Throughput | Received KB/... | Sent KB/sec | Avg. Bytes |
|----------------|-----------|---------|-------|-------|-----------|---------|------------|-----------------|-------------|------------|
| Register       | 2000      | 33006   | 462   | 71181 | 18190.29  | 0.00%   | 24.0/sec   | 10.45           | 10.54       | 446.0      |
| Login          | 2000      | 58269   | 22360 | 78018 | 8371.01   | 0.00%   | 15.0/sec   | 9.11            | 5.85        | 623.0      |
| Get Book       | 2000      | 35834   | 10503 | 68088 | 14936.67  | 0.00%   | 15.7/sec   | 11.36           | 5.01        | 739.0      |
| Insert Review  | 2000      | 21079   | 10987 | 46365 | 6515.86   | 0.00%   | 23.0/sec   | 8.55            | 11.47       | 380.0      |
| Create Book... | 2000      | 18338   | 3409  | 29401 | 5828.68   | 0.00%   | 44.2/sec   | 16.41           | 18.82       | 380.0      |
| TOTAL          | 10000     | 33305   | 462   | 78018 | 18459.07  | 0.00%   | 56.0/sec   | 28.06           | 23.19       | 513.6      |

**Figura 19:** 2000 clientes

| Label          | # Samples | Average | Min   | Max    | Std. Dev. | Error % | Throughput | Received KB/... | Sent KB/sec | Avg. Bytes |
|----------------|-----------|---------|-------|--------|-----------|---------|------------|-----------------|-------------|------------|
| Register       | 5000      | 87149   | 349   | 176282 | 48631.09  | 0.00%   | 22.3/sec   | 9.71            | 9.80        | 446.0      |
| Login          | 5000      | 134053  | 16347 | 176714 | 33213.69  | 15.54%  | 13.8/sec   | 12.45           | 4.55        | 925.1      |
| Get Book       | 5000      | 76815   | 13297 | 170880 | 44802.29  | 9.84%   | 13.1/sec   | 11.75           | 3.75        | 920.1      |
| Insert Review  | 5000      | 44742   | 13297 | 152821 | 23508.96  | 4.28%   | 16.1/sec   | 7.45            | 7.67        | 474.3      |
| Create Book... | 5000      | 34043   | 832   | 114724 | 10975.03  | 0.32%   | 26.2/sec   | 9.91            | 11.12       | 387.1      |
| TOTAL          | 25000     | 75361   | 349   | 176714 | 49757.53  | 6.00%   | 58.9/sec   | 36.26           | 23.05       | 630.5      |

**Figura 20:** 5000 clientes

# Anexos



**Figura 21:** PSM Customer Completo

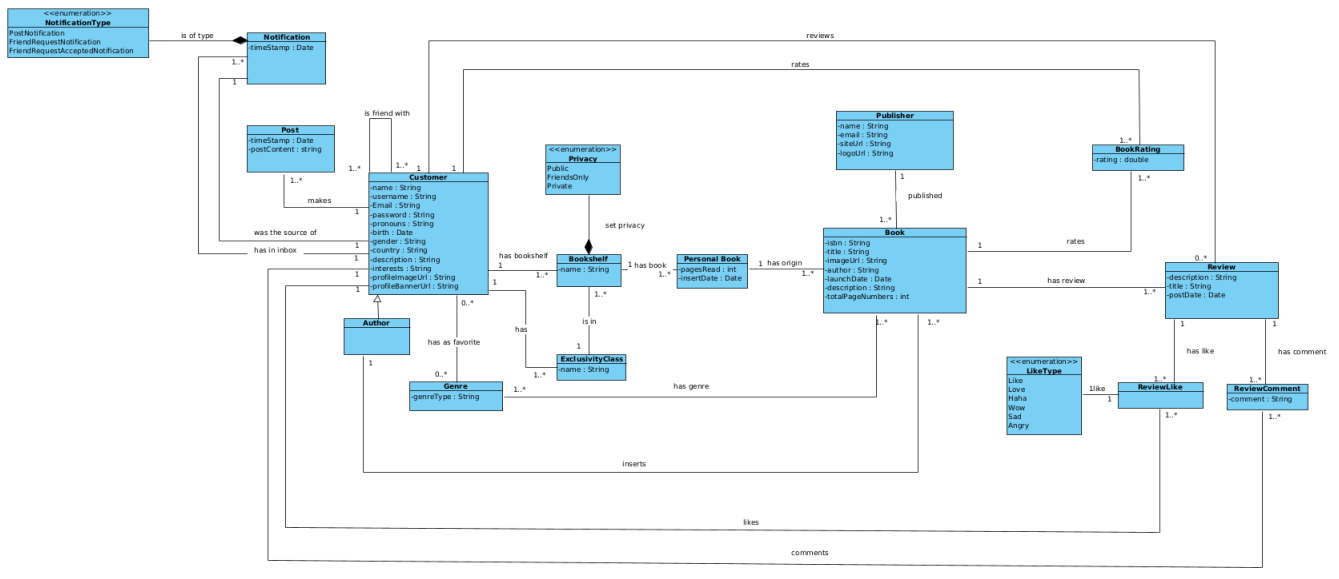


Figura 22: Diagram PIM Hibernate Completo

## Bibliografia

- [1] Spring Boot, <https://spring.io/projects/spring-boot>