

Faculdade de Ciências e Tecnologias
Departamento de Engenharia Informática

Plataforma para publicação e subscrição de conteúdo específico de uma área

Platform for publishing and subscribing to domain specific content over written publication

Dinis Coelho Marques
dcm@student.dei.uc.pt

Relatório de Estágio no âmbito do Mestrado em Engenharia Informática, especialização em
Sistemas Inteligentes orientado pelo Prof. João Fernandes e Eng. Pedro Cioga e apresentada à
Faculdade de Ciências e Tecnologia / Departamento de Engenharia Informática.

Julho 2019

1 2  9 0

UNIVERSIDADE D
COIMBRA

Agradecimentos

Em primeiro lugar quero agradecer à Redlight Software pelo estágio e por me ter acolhido durante estes dois semestres. Um agradecimento especial às pessoas ligadas ao projeto que estiveram sempre dispostas a ajudar.

Quero também agradecer ao orientador do DEI, Prof. João Fernandes, e ao orientador da RLS, Pedro Cioga, pelo acompanhamento ao longo do estágio.

Para finalizar quero também agradecer à minha família, amigos e colegas que me acompanharam neste percurso académico.

Resumo

O Anotado é uma linha de produtos de software móveis desenvolvido e comercializado pela RedLight Software, cujo objetivo é permitir a qualquer profissional a criação de conteúdo específico de um domínio (anotações) sobre publicações pré-existentes e acessíveis gratuitamente ao público.

As anotações são comentários, ideias, sugestões ou explicações de um conteúdo específico. Por exemplo uma anotação de um artigo de uma lei seria uma explicação do mesmo ou até uma sugestão de como este pode ser interpretado.

Pessoas na área da advocacia utilizam essas anotações ou outras sobre um conteúdo específico quando têm dúvidas sobre o mesmo. O acesso a estas anotações normalmente é feito através de livros, que por norma costumam ter uma custo elevado associado, este acesso pode ser facilitado recorrendo ao uso da tecnologia.

Contudo, atualmente, são poucas as aplicações que facilitam todo este processo de partilha, acesso e criação de anotações. Os utilizadores por vezes têm que recorrer a várias aplicações para puderem usufruir de múltiplas funcionalidades.

Durante o estágio foi desenvolvido uma aplicação web em que os utilizadores podem criar as suas próprias anotações, com ou sem colaboração de outros utilizadores, e depois as podem partilhar de forma gratuita ou até mesmo as vender, tudo isto sem saírem da plataforma.

Palavras-Chave

anotações, aplicação web, colaboração, partilha, venda, compra

Abstract

Anotado is a line of mobile software products developed and marketed by RedLight Software, which aims to allow any professional to create domain-specific content (annotations) about pre-existing publications and accessible to the public free of charge.

Annotations are comments, ideas, suggestions, or explanations of a specific content. For example an annotation of an article of a law would be an explanation of the same or even a suggestion of how it can be interpreted.

People in the advocacy field use those annotations or others about specific content when they have doubts about it. Access to these annotations is usually done through books, which usually have a high associated cost, this access can be facilitated using technology.

However, there are currently very few applications that facilitate this whole process of sharing, accessing and annotating. Users sometimes have to resort to multiple applications to take advantage of multiple features.

During the internship was developed a web application where users can create their own annotations, with or without the collaboration of other users, and then share them for free or even sell them, all without leaving the platform.

Keywords

annotations, web application, collaboration, share, sell, buy

Conteúdo

1	Introdução	1
1.1	Contexto	1
1.2	Motivação	2
1.3	Objetivos	2
1.4	Estrutura do documento	2
2	Estado da Arte	5
2.1	Produto atual - aplicação móvel	6
2.2	Análise a plataformas web	8
2.2.1	Annotation Studio	8
2.2.2	DocumentCloud	12
2.2.3	Genius	14
2.3	Comparação das aplicações	17
3	Descrição do processo	19
3.1	Abordagem ao conceito	19
3.2	Diagrama de Gantt	20
4	Especificações do projeto	23
4.1	Modelo de desenvolvimento de software	23
4.2	Requisitos	23
4.2.1	Requisitos funcionais	24
4.2.2	Requisitos não-funcionais	24
4.3	Arquiteturas do sistema	26
4.3.1	Arquitetura do servidor	26
4.3.2	Arquitetura da aplicação web	33
4.4	Tecnologias e ferramentas	37
4.4.1	Servidor	37
4.4.2	Aplicação web	38
4.4.3	Ferramentas	39
4.5	Desafios tecnológicos	39
4.6	Riscos e estratégias de mitigação	42
5	Desenvolvimento	47
5.1	Sprints	47
5.2	Metodologia	49
6	Verificação e validação	53
6.1	Indentação de Código	53
6.2	Testes	54
6.3	<i>Code Review</i>	57
6.4	<i>Quality Assurance</i>	58

6.5	<i>Sentry</i>	59
7	Resultados e métricas do estágio	61
7.1	Estado final da aplicação	61
7.2	Objetivos e métricas	67
7.2.1	Requisitos Funcionais	67
7.2.2	Requisitos Não Funcionais	68
8	Conclusão	73
8.1	Futuro do Projeto	74
	Referências	75
	Anexos	77
A	Requisitos Funcionais	81

Glossário

Application Programming Interface Uma Application Programming Interface (API) é um grupo de métodos que um software fornece para que aplicações externas possam fazer uso desses métodos sem necessidade de conhecer os detalhes de implementação. viii, ix, 27

Framework Abstração de código que oferece funcionalidades genéricas que o programador pode usar para implementar funcionalidades específicas. 37, 40

Gamification Aplicação de técnicas de *game-design* em produtos não relacionados com jogos. 18

gem Uma biblioteca ou aplicação que adiciona um conjunto de funcionalidades à linguagem de programação Ruby. 37

Acrónimos

API Application Programming Interface. viii, 27, 37

CPC Código de Processo Civil Brasileiro. 1, 5, 6, 8, 49, 51

QA Quality Assurance. 3

REST Representational State Transfer. 27, 40, 41

RLS Redlight Software[1]. 1, 2, 5, 20, 43

RoR Ruby On Rails. 39, 40, 49

Lista de Figuras

2.1	CPC Anotado- Listar conteúdo	6
2.2	CPC Anotado - Listar artigos	6
2.3	CPC Anotado - Pesquisar	7
2.4	CPC Anotado - Visualizar Anotações com Premium	7
2.5	CPC Anotado - Visualizar Anotações sem Premium	7
2.6	Annotation Studio - Dashboard	8
2.7	Annotation Studio - Listar conteúdos	9
2.8	Annotation Studio - Listar anotações	9
2.9	Annotation Studio - Adicionar conteúdo	10
2.10	Annotation Studio - Visualizar Anotações no texto	10
2.11	Annotation Studio - Visualizar Anotações na lateral	11
2.12	Annotation Studio - Criar Anotação	11
2.13	Annotation Studio - Escrever nova Anotação	12
2.14	DocumentCloud - Listar conteúdos	13
2.15	DocumentCloud - Visualizar Anotações	13
2.16	Genius - Listar/Pesquisar conteúdos	14
2.17	Genius - Visualizar Anotações	15
2.18	Genius - Criar Anotação	15
2.19	Genius - Escrever nova Anotação	16
3.1	Diagrama de Gantt - Fases do projeto	20
4.1	Diagrama de contexto	26
4.2	Diagrama de <i>containers</i>	27
4.3	Diagrama de componentes	28
4.4	Diagrama de relacionamento de entidades	30
4.5	Diagrama da aplicação web	34
4.6	REST vs GraphQL	41
5.1	Diagrama de Gantt dos <i>Sprints</i>	49
5.2	Exemplo de visualização de uma tarefa no JIRA	49
5.3	JIRA - Exemplo da ferramenta de gestão de tarefas	50
5.4	Diagrama do processo de desenvolvimento	51
6.1	Exemplo do resultado do comando <i>rubocop</i>	54
6.2	Exemplo do resultado do comando <i>yarn lint</i>	54
6.3	Exemplo do resultado do comando <i>rspec</i>	55
6.4	Exemplo do resultado do comando <i>yarn test</i>	55
6.5	<i>Coverage</i> dos testes aos modelos	56
6.6	<i>Coverage</i> dos testes às <i>actions</i>	56
6.7	<i>Coverage</i> dos testes às páginas da aplicação	57
6.8	Exemplo de uma discussão resultante de <i>Code Review (CR)</i>	58

6.9	Exemplo de uma mensagem caso a tarefa falhe no <i>Quality Assurance (QA)</i> .	59
6.10	Exemplo da página de erros do <i>Sentry</i>	59
6.11	Exemplo da visualização de um erro no <i>Sentry</i>	60
7.1	Página de login	61
7.2	Página de listagem de conteúdos	62
7.3	Página de listagem de secções	62
7.4	Página de uma secção - listagem de anotações	63
7.5	Página de uma secção - criar nova anotação privada	63
7.6	Página dos favoritos	64
7.7	Página de pesquisa	64
7.8	Página da loja premium	65
7.9	Página de uma compra	65
7.10	Página de criação de bundle colaborativo	66
7.11	Página de um bundle colaborativo	66
7.12	Página de um bundle colaborativo	67
7.13	Exemplo do <i>Can I Use</i>	68
7.14	Comando <i>ab</i> e as suas configurações	69
7.15	Exemplo do resultado do comando <i>ab</i>	69
7.16	Exemplo do Chrome Dev Tools - Performance	70

Lista de Tabelas

2.1	Tabela comparativa de características - Plataformas Web	17
2.2	Tabela comparativa de características - Aplicações de Leitura e Extensões .	18
4.1	Lista e descrição das de tecnologias do servidor	37
4.2	Lista e descrição das tecnologias da aplicação web	38
4.3	Lista e descrição das de ferramentas	39
4.4	Tabela inicial de riscos e suas características	44
4.5	Tabela final de riscos e suas características	44
7.1	Métricas dos requisitos funcionais	67
7.2	Resultados dos testes de performance ao servidor	70
7.3	Resultados dos testes de performance à aplicação web	71

Capítulo 1

Introdução

Este documento descreve todo o trabalho desenvolvido ao longo do estágio que decorreu na empresa Redlight Software[1] (RLS) .

Neste capítulo estão detalhados os conteúdos do projeto a desenvolver e os conteúdos deste relatório. Começando por descrever o contexto no qual o projeto se enquadra e qual a motivação por trás do mesmo. São também listados os objetivos do estágio e do projeto. Para finalizar é feita uma breve descrição de toda a estrutura do documento.

1.1 Contexto

O **Anotado** é uma linha de produtos de software móveis desenvolvido e comercializado pela RLS, cujo objetivo é permitir a qualquer profissional a criação de conteúdo adicional a um domínio (anotações) sobre publicações pré-existentes e acessíveis gratuitamente ao público. Atualmente encontra-se disponível ao público o primeiro produto Anotado, focado no Código de Processo Civil Brasileiro (CPC), sendo utilizado diariamente por centenas de profissionais para consulta de anotações de profissionais e comparação entre diferentes versões do CPC.

As anotações são informações complementares a um conteúdo específico, normalmente criadas por pessoas experientes na área e depois utilizadas por outras com o intuito de expandir o conhecimento sobre esse mesmo conteúdo.

Estas anotações são usadas em diversas áreas desde o ensino, como por exemplo ajudar no estudo de uma obra (*e.g.* Mensagem de Fernando Pessoa), à advocacia, como por exemplo ajudar na perceção do significado de uma lei.

Este primeiro produto surge de uma já consolidada parceria entre a RLS e uma entidade Brasileira da área da justiça/advocacia, a qual organizou um conjunto de profissionais para criar as anotações que são vendidas na aplicação.

Numa fase inicial esta aplicação foi lançada com o nome **CPC Comparado** e apenas tinha a funcionalidade de comparar versões antigas de artigos com versões mais atuais, ou seja, mostrava quais os artigos do CPC de 1973 que correspondiam ao artigo do CPC de 2015, o atualmente em vigor.

Foi numa segunda iteração desta aplicação que nasceu o atual **Anotado**, mantendo a funcionalidade de comparação mas adicionando a possibilidade de consultar anotações de profissionais por um custo reduzido.

1.2 Motivação

A forma comum para aceder e partilhar anotações de conteúdo específico é normalmente em formato físico, mais propriamente o papel. Com o crescimento do uso de tecnologia no dia-a-dia faz sentido haver uma solução digital que facilite o processo de acesso e partilha dessas anotações.

Os principais problemas relacionados com o uso de anotações em papel são a pesquisa, partilha de conteúdo a uma escala global e a publicação de conteúdo pessoal com essa mesma escala.

A solução pretende permitir que vários juristas ou pessoas que trabalham na área consigam tirar mais valor da lei que já existe com conhecimento publicamente disponibilizado proveniente de pessoas que estudam a matéria há muitos anos.

1.3 Objetivos

Os objetivos do estágio foram definidos pelo cliente, RLS, e consistem em:

1. Criação de uma aplicação web de leitura das leis já acessíveis nas aplicações móveis do Anotado, incluindo as componentes de back-end e front-end;
2. Implementação da ferramenta para criação de anotações privadas por parte dos utilizadores ficando disponíveis no cliente móvel para consulta, incluindo as componentes de back-end e front-end;
3. Criação de uma loja online para compra de anotações disponíveis na aplicação, incluindo as componentes de back-end e front-end;
4. Implementação das alterações necessárias à plataforma para estar em concordância com a alteração de 2018 do Regulamento Geral de Proteção de Dados.

De uma forma mais detalhada o aluno tem que desenvolver a aplicação web e o servidor utilizando tecnologias atuais e implementar as funcionalidades definidas no Capítulo 4 - Especificações do Projeto, Secção 4.2 - Requisitos. Está ainda a cargo do aluno realizar a arquitetura do servidor e da aplicação web, desenhar a estrutura da base de dados, integrar com serviços externos de pagamento e realizar testes automáticos. O design e a garantia de qualidade da aplicação está fora dos objetivos do aluno e serão realizados por membros da empresa responsáveis por essas áreas.

1.4 Estrutura do documento

Nesta secção vamos identificar os capítulos do documento e o seu objetivo.

- Capítulo 1 - **Introdução**
 - É o capítulo inicial do documento, onde é explicado o contexto e a motivação que levou à necessidade deste produto e projeto. Também agrega tanto os objetivos do estágio como do projeto.
- Capítulo 2 - **Estado da Arte**

- Neste capítulo vamos analisar várias plataformas de partilha de anotações, pagas ou não. Vamos também analisar a atual aplicação móvel existente. Depois destas análises será feita uma agregação de pontos fortes e fracos de cada plataforma, com o objetivo de influenciar o desenho do produto final.
- **Capítulo 3 - Descrição do processo**
 - É neste capítulo que vamos falar sobre as abordagens e decisões tomadas antes de definir o conceito do produto e detalhar superficialmente todas as outras fases de planeamento e desenvolvimento do projeto.
- **Capítulo 4 - Especificações do projeto**
 - Neste capítulo vamos descrever todos os processos de especificação do projeto, desde o modelo de desenvolvimento adotado, requisitos do projeto, arquiteturas dos sistemas, tecnologias e os seus desafios e ainda os riscos encontrados.
- **Capítulo 5 - Desenvolvimento**
 - Neste capítulo vamos descrever o processo de desenvolvimento utilizado e a sua organização em *Sprints*.
- **Capítulo 6 - Verificação e validação**
 - Neste capítulo vamos listar e detalhar todos os métodos utilizados na verificação e validação do projeto.
- **Capítulo 7 - Resultados e métricas do estágio**
 - Neste capítulo iremos observar o resultado final da aplicação e analisar as métricas relativas ao desenvolvimento.
- **Capítulo 8 - Conclusão**
 - Este capítulo sumariza todo o trabalho feito no primeiro semestre, com foco nos detalhes mais importantes.

Capítulo 2

Estado da Arte

Neste capítulo vamos analisar várias plataformas de partilha de anotações.

O meio mais comum de distribuição de anotações que existe é em formato físico, que vai desde dos pequenos livros que são usados por alunos de ciclo e secundário aos códigos e leis anotadas.

A nível tecnológico, existem poucas ferramentas que se assemelhem ou partilhem funcionalidades com o que se pretende desenvolver. Isto deve-se ao facto de parte do público alvo serem profissionais de todas as idades e alguns deles não terem tanta facilidade em usar as tecnologias mais recentes, dando preferência ao formato em papel.

Também vamos analisar a aplicação móvel que existe atualmente, o **CPC Anotado**[2], apesar desta apenas possuir as funcionalidades de leitura é um ponto de partida para comparação com as existentes e com o que futuramente será a versão web. Esta análise é detalhada na **Secção 2.1**. Esta análise foi feita tendo em conta o estado da aplicação antes de começar o estágio.

Visto que é uma iteração de um produto já existente esta análise ao estado da arte teve como base alguns dos dados obtidos numa prévia análise ao estado de arte realizado pela empresa relativos a funcionalidades comuns a ambas as plataformas.

Foram apenas encontradas três plataformas web, Annotation Studio[3], DocumentCloud[4] e Genius[5], a funcionar e com conteúdos com semelhanças ao objetivo do projeto, mas até as funcionalidades oferecidas por estas, não são suficientes para dar resposta ao problema que foi proposto resolver. Foram encontradas outras plataformas parecidas mas não tinham conteúdos, WritingPod[6], ou não se encontravam a funcionar corretamente, Infinite Ulysses[7]. Estas plataformas web são analisadas em detalhe na **Secção 2.2**.

Para além destas plataformas existem inúmeras aplicações para leitura/criação de documentos digitais que permitem criar anotações no próprio documento, tais como o Adobe Acrobat Reader[8] e o XODO[9], existem ainda outras que permitem a fácil partilha desses documentos tais como o Google Docs[10], o Mendeley[11] ou o Kami[12].

Foi também encontrada uma extensão para o browser, Hypothesis[13] que permite criar anotações em *websites* e partilhar essas mesmas anotações.

2.1 Produto atual - aplicação móvel

O **CPC Anotado** foi a primeira aplicação móvel, desenvolvida pela Redlight Software[1] (RLS), da linha de produtos **Anotado**. Esta permite consultar o Código de Processo Civil Brasileiro (CPC) de 2015, os artigos correspondentes do CPC de 1973 e respectivas anotações criadas por profissionais da área. Estas anotações foram obtidas através de uma parceria com dois advogados brasileiros que convidaram vários profissionais do ramo da advocacia a anotar um ou vários artigos específicos da sua área.

O utilizador para poder consultar a totalidade dessas anotações tem que possuir perfil *premium* na aplicação, ou seja, tem que pagar para ter acesso a essas anotações. Caso o utilizador não tenha *premium* apenas tem direito a visualizar dez anotações grátis.

A aplicação móvel foi analisada tendo em conta a versão na qual se encontrava no início do estágio.

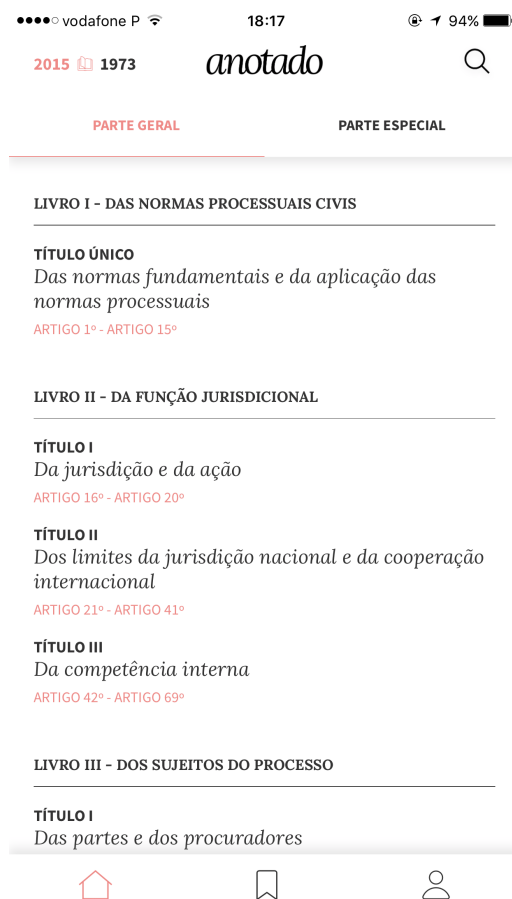


Figura 2.1: CPC Anotado- Listar conteúdo

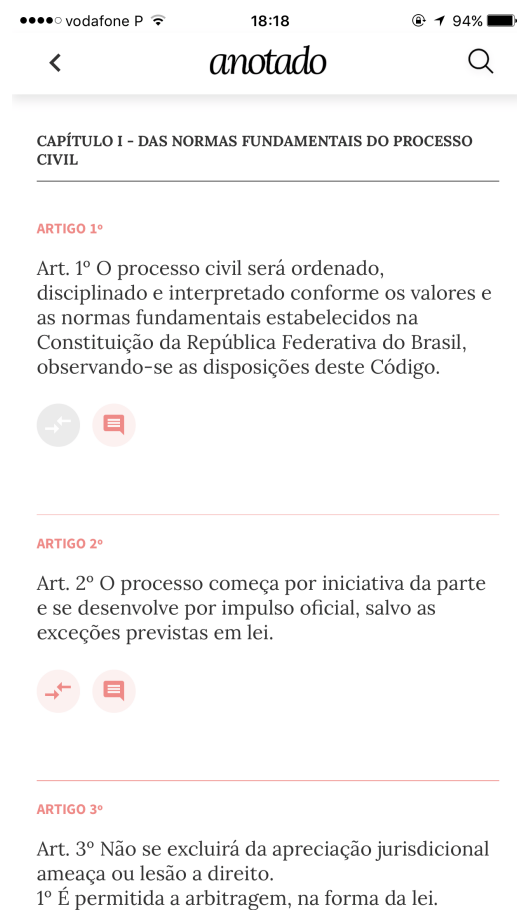


Figura 2.2: CPC Anotado - Listar artigos

O conteúdo é mostrado ao utilizador em forma de lista, no caso da Figura 2.1 é possível visualizar os títulos correspondentes aos Livros I e II da Parte Geral do CPC de 2015.

Ao clicar num dos títulos a aplicação abre uma página com a lista de artigos desse título. Cada artigo possui dois ícones, como apresentado na Figura 2.2, que indicam a existência de comparações ou anotações.

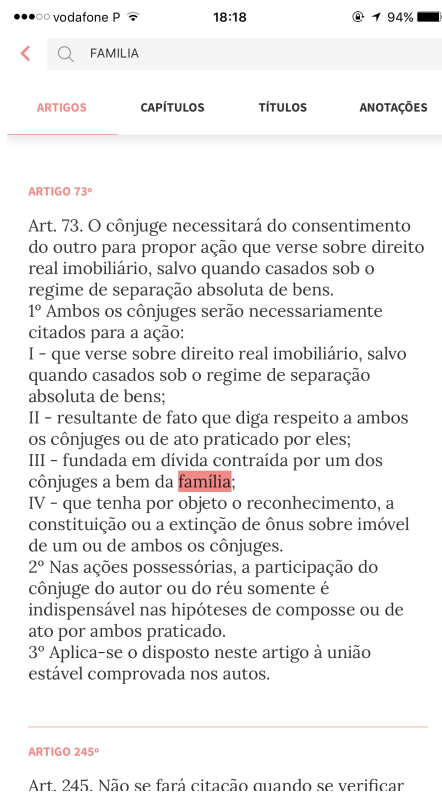


Figura 2.3: CPC Anotado - Pesquisar

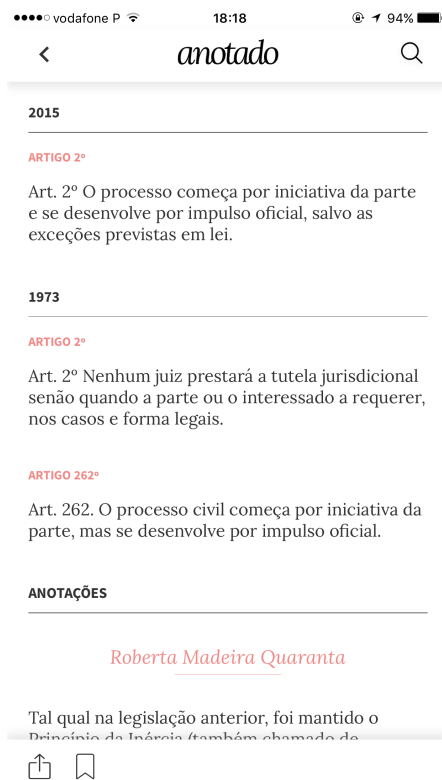


Figura 2.4: CPC Anotado - Visualizar Anotações com Premium

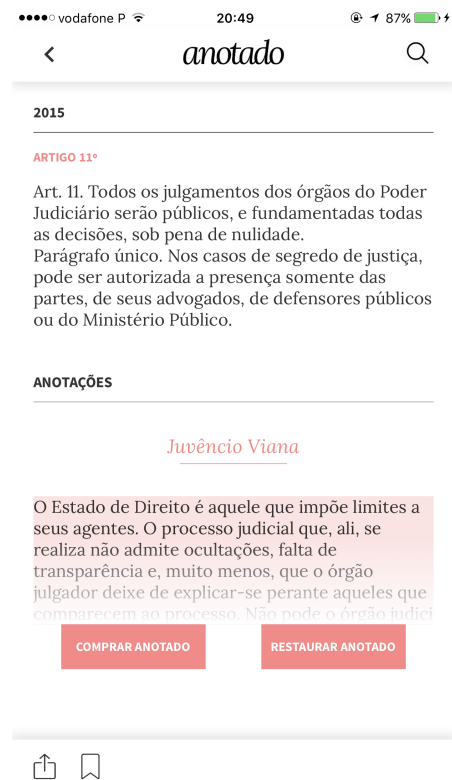


Figura 2.5: CPC Anotado - Visualizar Anotações sem Premium

A pesquisa apresenta os resultados em forma de lista e realça as ocorrências da palavra procurada, como é possível observar pela Figura 2.3 onde é mostrado o resultado de uma procura pelo termo “FAMILIA”. Na imagem podemos visualizar que a pesquisa não é sensível às maiúsculas e às minúsculas. É também possível alternar entre o tipo de conteúdos disponíveis, Artigos, Capítulos, Títulos e Anotações.

Na página de um artigo é possível visualizar as suas anotações e os artigos do CPC de 1973 que se relacionam com o mesmo, como é mostrado na Figura 2.4 é possível ver o Artigo 2º do CPC de 2015, os artigos correspondentes do CPC de 1973 e uma anotação. Caso o utilizador não possua o *premium*, ou seja, não pagou para ter acesso às anotações, não consegue visualizar as anotações por completo, aparecendo apenas uma parte da anotação e dois botões, um que permite comprar o acesso à anotação e outro que permite recuperar uma compra previamente realizada, como demonstrado na Figura 2.5.

2.2 Análise a plataformas web

Vamos seguidamente proceder à análise de plataformas web que se encontram a funcionar e com conteúdos disponíveis.

2.2.1 Annotation Studio

O **Annotation Studio**[3], desenvolvido pelo MIT[14], é uma plataforma mais focada para uso escolar, tanto que é possível organizar os documentos e as anotações por grupo, pessoais ou disciplina. É uma plataforma que necessita de registo para aceder a todas as funcionalidades, mas que é grátis e de livre acesso a qualquer pessoa.

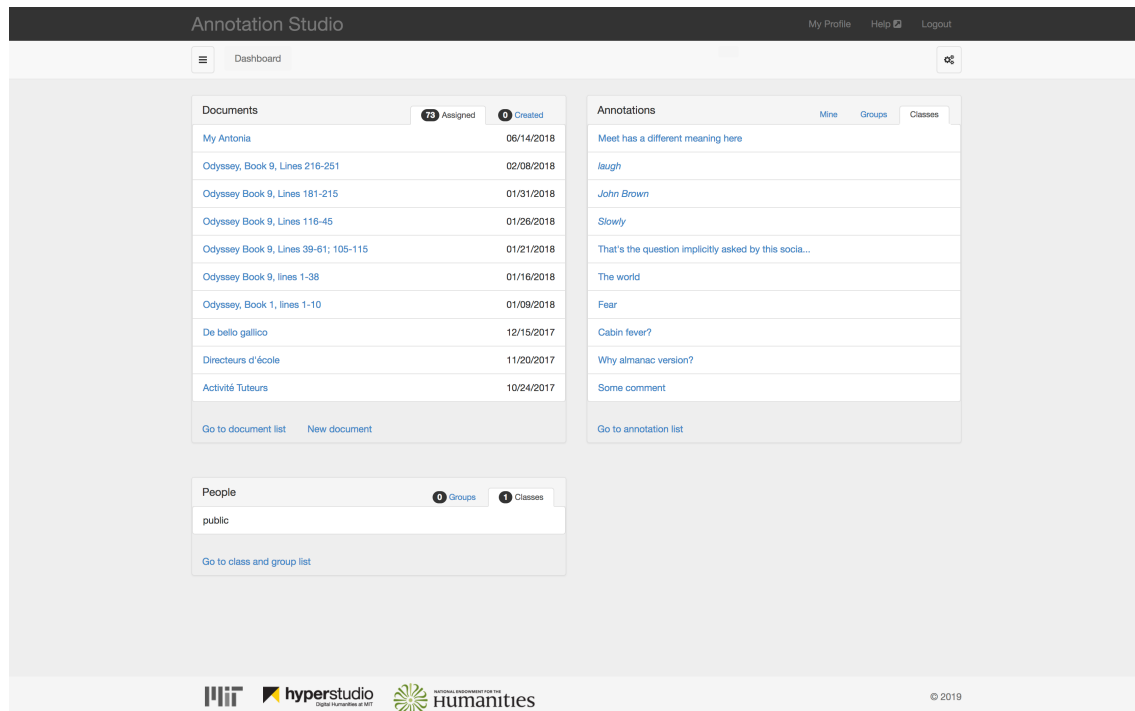
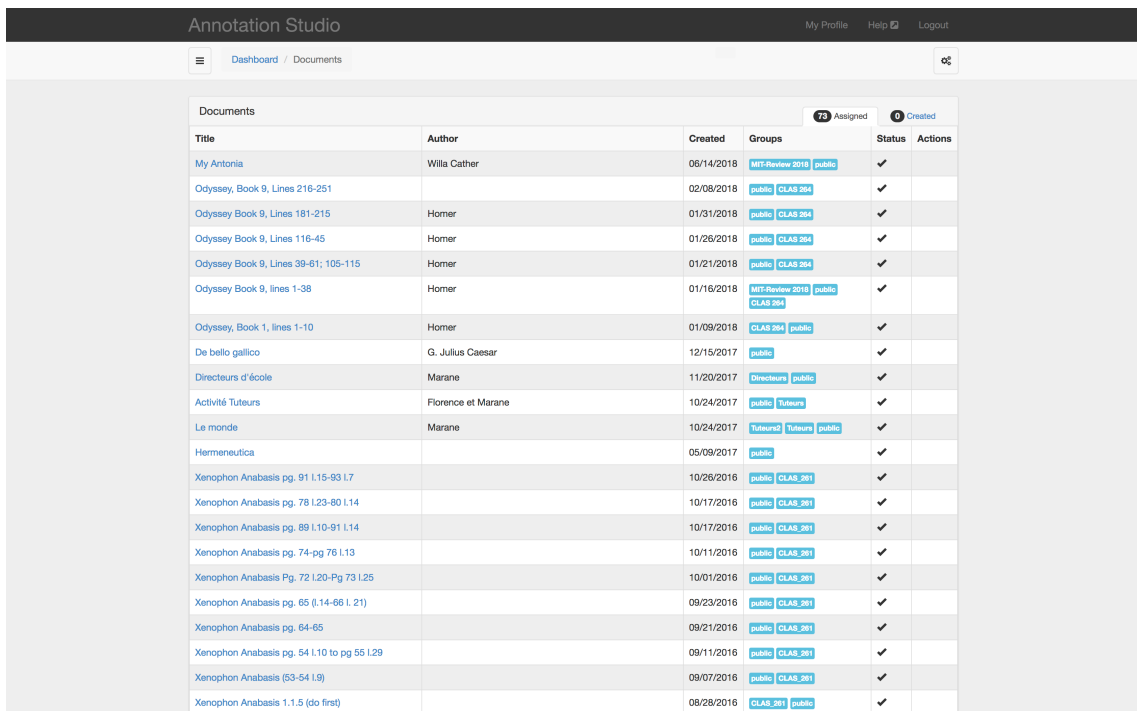


Figura 2.6: Annotation Studio - Dashboard

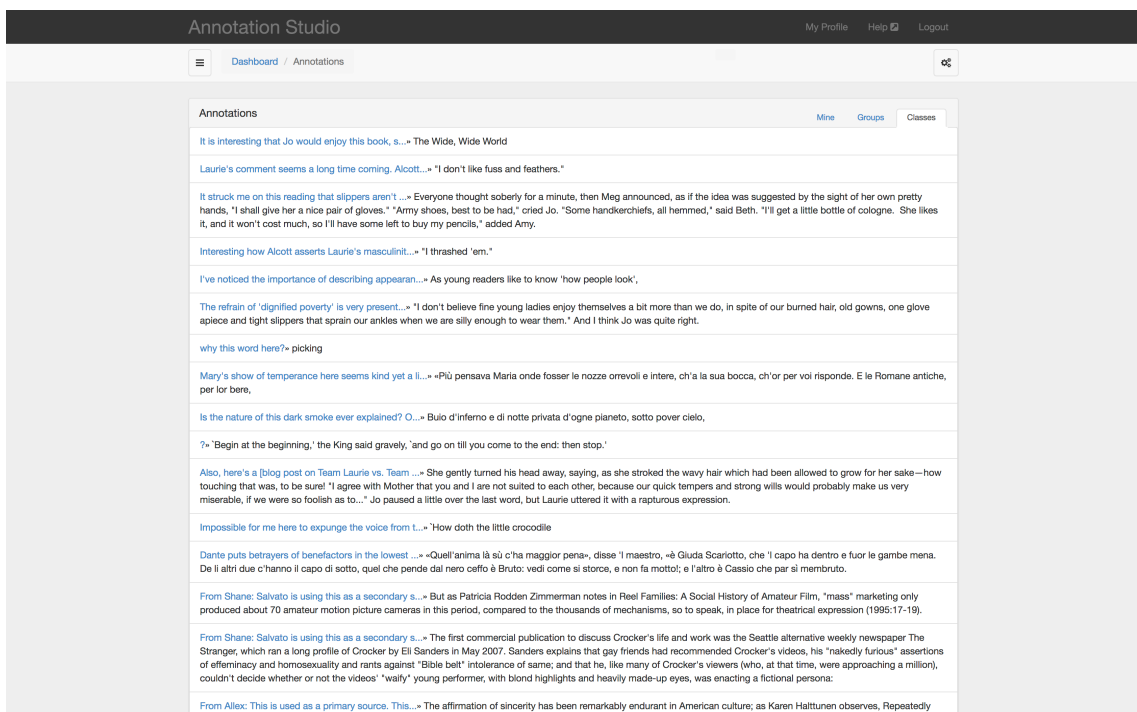
Na *dashboard*, Figura 2.6, é possível consultar uma mini lista de documentos e anotações, organizadas pela data de criação.



The screenshot shows the 'Documents' section of the Annotation Studio dashboard. It features a table with columns for Title, Author, Created, Groups, Status, and Actions. The table lists various documents, including 'My Antonia', 'Odyssey, Book 9, Lines 216-251', and 'De bello gallico'. Each row includes a 'Created' date, a 'Groups' column with links to specific groups, and a 'Status' column with a checkmark indicating completion.

Title	Author	Created	Groups	Status	Actions
My Antonia	Willa Cather	06/14/2018	MT-Review 2018 public	✓	
Odyssey, Book 9, Lines 216-251		02/08/2018	public CLAS 204	✓	
Odyssey Book 9, Lines 181-215	Homer	01/31/2018	public CLAS 204	✓	
Odyssey Book 9, Lines 116-45	Homer	01/26/2018	public CLAS 204	✓	
Odyssey Book 9, Lines 39-61; 105-115	Homer	01/21/2018	public CLAS 204	✓	
Odyssey Book 9, lines 1-38	Homer	01/16/2018	MT-Review 2018 public CLAS 204	✓	
Odyssey, Book 1, lines 1-10	Homer	01/09/2018	CLAS 204 public	✓	
De bello gallico	G. Julius Caesar	12/15/2017	public	✓	
Directeurs d'école	Marane	11/20/2017	Directeurs public	✓	
Activité Tuteurs	Florence et Marane	10/24/2017	public Tuteurs	✓	
Le monde	Marane	10/24/2017	Tuteurs2 Tuteurs public	✓	
Hermeneutica		05/09/2017	public	✓	
Xenophon Anabasis pg. 91 l.15-93 l.7		10/26/2016	public CLAS 201	✓	
Xenophon Anabasis pg. 78 l.23-80 l.14		10/17/2016	public CLAS 201	✓	
Xenophon Anabasis pg. 89 l.10-91 l.14		10/17/2016	public CLAS 201	✓	
Xenophon Anabasis pg. 74-pg 76 l.13		10/11/2016	public CLAS 201	✓	
Xenophon Anabasis Pg. 72 l.20-Pg 73 l.25		10/01/2016	public CLAS 201	✓	
Xenophon Anabasis pg. 65 (l.14-66 l. 21)		09/23/2016	public CLAS 201	✓	
Xenophon Anabasis pg. 64-65		09/21/2016	public CLAS 201	✓	
Xenophon Anabasis pg. 54 l.10 to pg 55 l.29		09/11/2016	public CLAS 201	✓	
Xenophon Anabasis (53-54 l.9)		09/07/2016	public CLAS 201	✓	
Xenophon Anabasis 1.1.5 (do first)		08/28/2016	CLAS 201 public	✓	

Figura 2.7: Annotation Studio - Listar conteúdos



The screenshot shows the 'Annotations' section of the Annotation Studio dashboard. It features a list of annotations with columns for Mine, Groups, and Classes. The annotations are text-based and include references to various literary works and authors, such as 'The Wide, Wide World', 'Laurie's comment seems a long time coming', and 'Dante puts betrayers of benefactors in the lowest...'. Each annotation is followed by a brief description or context.

Annotations	Mine	Groups	Classes
It is interesting that Jo would enjoy this book, s...- The Wide, Wide World			
Laurie's comment seems a long time coming. Alcott...- "I don't like fuss and feathers."			
It struck me on this reading that slippers aren't...- Everyone thought soberly for a minute, then Meg announced, as if the idea was suggested by the sight of her own pretty hands, "I shall give her a nice pair of gloves." "Army shoes, best to be had," cried Jo. "Some handkerchiefs, all hemmed," said Beth. "I'll get a little bottle of cologne. She likes it, and it won't cost much, so I'll have some left to buy my pencils," added Amy.			
Interesting how Alcott asserts Laurie's masculinit...- "I thrashed 'em."			
I've noticed the importance of describing appearan...- As young readers like to know 'how people look',			
The refrain of 'dignified poverty' is very present...- "I don't believe fine young ladies enjoy themselves a bit more than we do, in spite of our burned hair, old gowns, one glove apiece and tight slippers that sprain our ankles when we are silly enough to wear them." And I think Jo was quite right.			
why this word here?- picking			
Mary's show of temperance here seems kind yet a li...- «Più pensava Maria onde fossen le nozze orrevoli e intere, ch'a la sua bocca, ch'or per voi risponde. E le Romane antiche, per lor bere,			
Is the nature of this dark smoke ever explained? O...- Bulò d'inferno e di notte privata d'ogne pianeta, sotto pover cielo,			
?'- 'Begin at the beginning,' the King said gravely, 'and go on till you come to the end: then stop.'			
Also, here's a [blog post on Team Laurie vs. Team ...- She gently turned his head away, saying, as she stroked the wavy hair which had been allowed to grow for her sake—how touching that was, to be sure! "I agree with Mother that you and I are not suited to each other, because our quick tempers and strong wills would probably make us very miserable, if we were so foolish as to..." Jo paused a little over the last word, but Laurie uttered it with a rapturous expression.			
Impossible for me here to expunge the voice from L...- "How doth the little crocodile			
Dante puts betrayers of benefactors in the lowest...- «Quell'anima là sù c'ha maggior pena», disse 'l maestro, «è Giuda Scariotto, che 'l capo ha dentro e fuor le gambe mena. De li altri due c'hanno il capo di sotto, quel che pende dal nero ceffo è Bruto: vedi come si storce, e non fa motto; e l'altro è Cassio che par sì membruto.			
From Shane: Salvato is using this as a secondary s...- But as Patricia Rodden Zimmerman notes in Reel Families: A Social History of Amateur Film, "mass" marketing only produced about 70 amateur motion picture cameras in this period, compared to the thousands of mechanisms, so to speak, in place for theatrical expression (1996:17-19).			
From Shane: Salvato is using this as a secondary s...- The first commercial publication to discuss Crocker's life and work was the Seattle alternative weekly newspaper The Stranger, which ran a long profile of Crocker by Eli Sanders in May 2007. Sanders explains that gay friends had recommended Crocker's videos, his "nakedly furious" assertions of effeminacy and homosexuality and rants against "Bible belt" intolerance of same; and that he, like many of Crocker's viewers (who, at that time, were approaching a million), couldn't decide whether or not the videos' "wally" young performer, with blond highlights and heavily made-up eyes, was enacting a fictional persona:			
From Alex: This is used as a primary source. This...- The affirmation of sincerity has been remarkably enduring in American culture; as Karen Halttunen observes, Repeatedly throughout American history, the sentimental impulse has returned to convince middle-class men and women of their social fate and to shape the incentives			

Figura 2.8: Annotation Studio - Listar anotações

Esta plataforma não possui uma funcionalidade de pesquisa, apenas de listagem de todos

os documentos, como podemos ver na Figura 2.7.

Para além da lista de todos os documentos existe também a possibilidade de listar todas as anotações, Figura 2.8, existentes na plataforma. Sendo que nesta lista temos acesso a parte do texto da anotação e ao texto que corresponde.

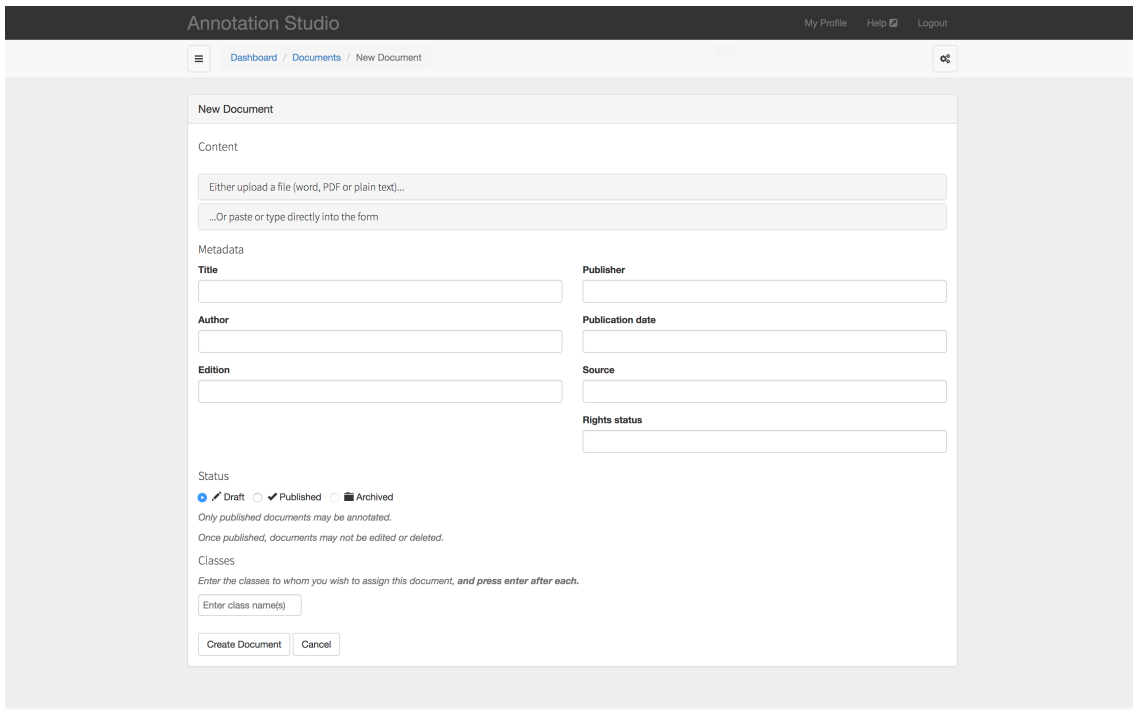


Figura 2.9: Annotation Studio - Adicionar conteúdo

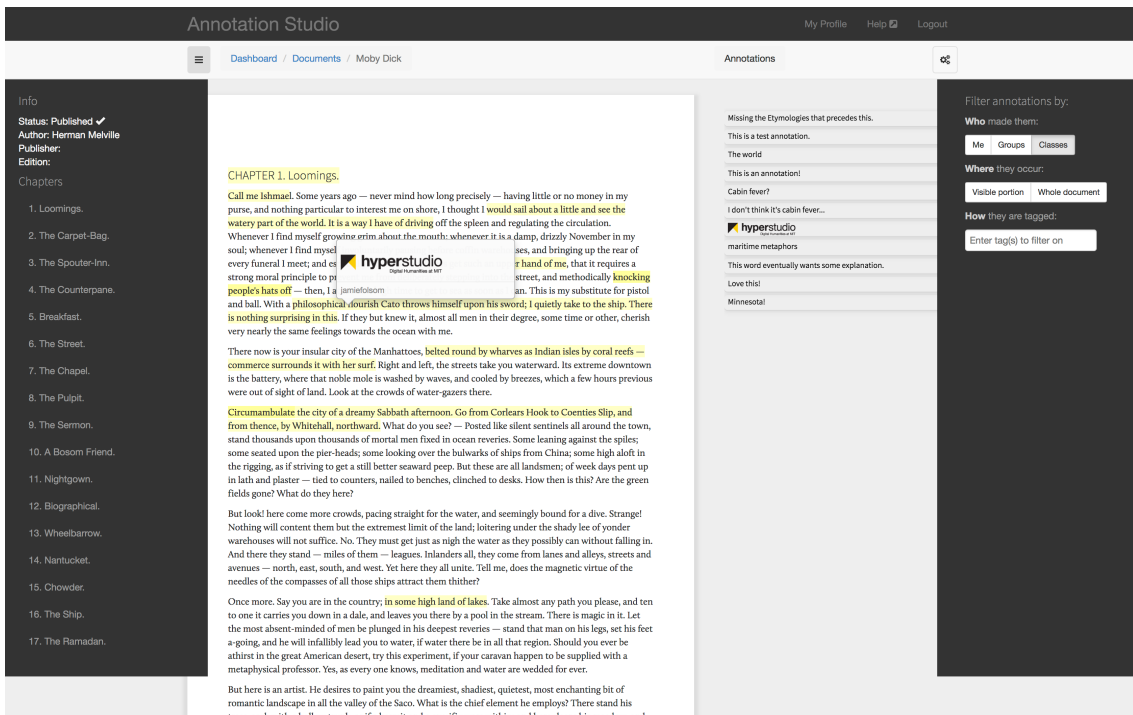


Figura 2.10: Annotation Studio - Visualizar Anotações no texto

O utilizador também pode inserir novos conteúdos na plataforma sejam eles em forma de ficheiros ou apenas texto, como mostrado na Figura 2.9.

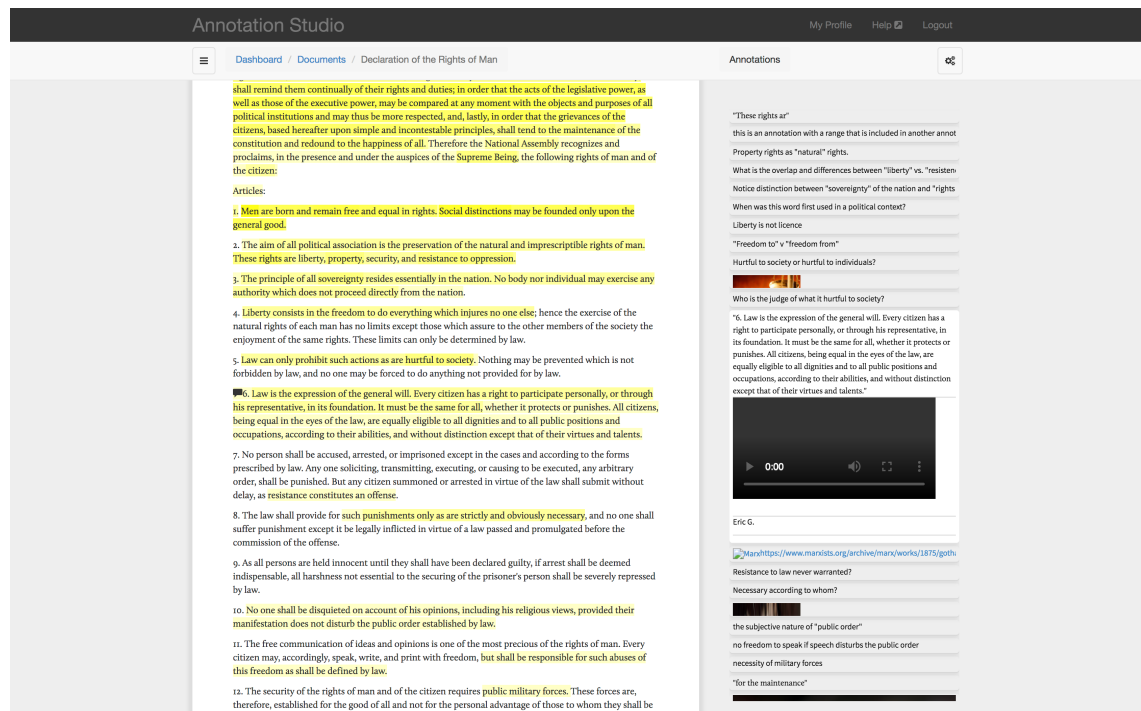


Figura 2.11: Annotation Studio - Visualizar Anotações na lateral

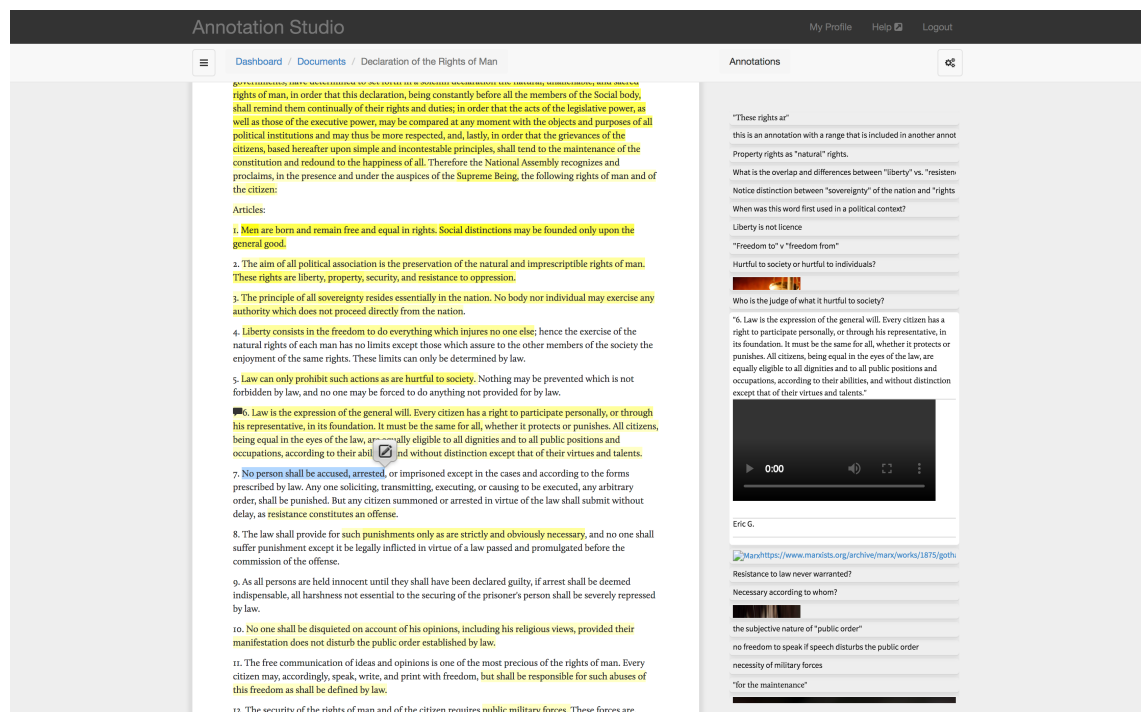


Figura 2.12: Annotation Studio - Criar Anotação

Na página de um documento é possível visualizar uma lista de anotações na lateral direita.

Existem duas formas de visualizar uma anotação por completo, uma opção é passar com o rato por cima do texto realçado a amarelo e anotação aparece por cima do documento, como é visível na Figura 2.10, ou então clicando na anotação que aparece na lista no lado direito, sendo que a anotação fica expandida, como na Figura 2.11.

Para criar uma anotação é necessário selecionar o texto que se pretende selecionar e depois clicar no botão com o lápis, Figura 2.12, aparecendo depois um editor, Figura 2.13, onde é possível escrever a anotação, sendo que o conteúdo da anotação pode conter imagens, vídeos ou hiperligações.

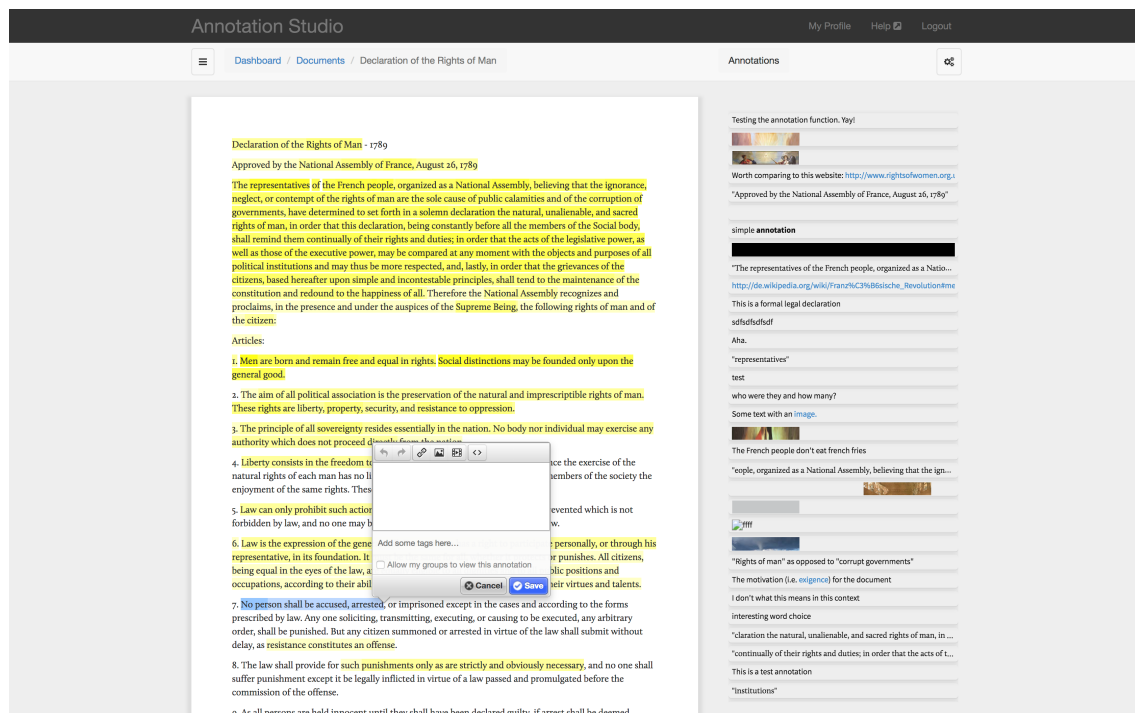


Figura 2.13: Annotation Studio - Escrever nova Anotação

2.2.2 DocumentCloud

No DocumentCloud[4], apesar de todos os conteúdos e anotações serem acessíveis por qualquer pessoa, é necessário conta para poder criar anotações e até à data de realização desta análise não foi possível obter uma conta, pois os registos apenas são possíveis para jornalistas: “*We’ve been open to journalists since 2010, but currently don’t accept non-journalism users.*”.

Como podemos observar através da Figura 2.14 a listagem e pesquisa de documentos na plataforma é muito modular e customizável. Existem quatro áreas distintas que facilitam a procura, duas na lateral esquerda e as outras duas no topo.

No menu lateral existe uma área para escolher os documentos que se pretende consultar, que facilita a procura por documentos anotados, e outra área com uma lista da fonte desses documentos.

No menu superior encontra-se a barra padrão de pesquisa, só que neste caso possibilita fazer uma pesquisa mais complexa, deixando utilizar várias opções para além da pesquisa só por texto. Por baixo dessa barra existem opções onde é possível aplicar filtros aos

resultados.

The screenshot shows the DocumentCloud interface with a search bar containing 'FILTER: annotated TEXT: portugal'. The results are displayed in a list format. On the left, there is a sidebar with 'ORGANIZATIONS' including Chicago-One News, University of Puerto Rico, WNYW-TV/Fox 5 News, #Copolitics, The (Colorado Springs) Gazette, The (Lynchburg) News & Advance, (Peoria) Journal Star, 100Reporters, 11Alive, 12 News, 14 East Magazine, 1878.info, and 3F. The main content area shows 263 results. The first few results are:

- In-Flight GSM**: 20 Pages - Contributed by Martin Umm... Dec 02, 2016. 3 Notes. 1 page mentioning "portugal".
- Cigarette Survey Summary 2010**: 3 Pages - Source: Conor Ryan - Contributed by Conor Ryan, The Story - Jul 16, 2012. Information released (via FOI) on survey carried out into illegal smoking habits. It was done for the Revenue Commissioners by IPSOS MRBI. 3 Notes. 1 page mentioning "portugal".
- 98 314 x2011003 2 Eng**: 9 Pages - Contributed by Kishlegi Abraham, David McKie's Research Methods - Feb 25, 2018. 2 Notes. 1 page mentioning "portugal".
- 2012 0004 B Bijlage Fiche Mozambique**: 5 Pages - Contributed by Ias van Beek, Platform Authentieke Journalistiek - May 10, 2018. 2 Notes. 1 page mentioning "portugal".

Figura 2.14: DocumentCloud - Listar conteúdos

The screenshot shows a document viewer in DocumentCloud. The document is a scanned page with a table of data. The table has columns for 'Part of description', 'Place of description', 'Equipment', 'Value', 'Conditions', and 'Organisation'. The text is in a historical or archival style. A yellow box highlights a section titled 'Shortage of Resources' with the following text:

"During the eight months of the 1940-41 Blitz, 16,000 tons of high explosives had destroyed factories, warehouses, docks and railway stations," writes Hampton in *The Austerity Olympics*. There was a huge rebuilding program after the war, but this meant that there was a shortage of materials such as timber, brick and cement to be used for Olympic venues. The Olympic Committee purchased timber springboards for the diving events from a supplier in Oregon. According to Hampton, Finland was asked to donate timber for the basketball floor.

On the right side of the viewer, there is a sidebar with a list of related documents or topics, including '1948 Olympics', 'Pitching In', 'Value of Equipment', 'Ports of Importation', 'Inspection', 'Shortage of Resources', 'Last-Minute Arrivals', 'After the Olympics', 'Colonel Bevan', 'Accommodation Plans', 'Richmond Park', 'Groupings', 'Research into Diets', 'Utbridge', 'American Food', 'Agreement with Middlesex Schools', 'Hired Chefs', 'Preparations at Wembley County', 'Australian Diet', 'Females in the Olympics', and 'Segregation of Sexes'. At the bottom of the sidebar, it says 'Original Document (PDF)', 'Print Notes', 'Contributed by Megan Gambino, Smithsonian.com', and 'DocumentCloud'.

Figura 2.15: DocumentCloud - Visualizar Anotações

Os resultados podem ser mostrados em formato lista ou em formato grelha, quando mostrados em lista aparece o documento que contém ocorrências das palavras pesquisadas

seguido de uma lista com as páginas e em que localização no texto aparecem os termos usados na procura.

As anotações são feitas através de caixas amarelas que delimitam uma área do documento. Para visualizar a anotação basta clicar nessa caixa que abre uma janela por cima do documento com a área delimitada pela caixa e o texto da anotação, como mostrado na Figura 2.15. Nesta pequena janela, para além de ler a anotação é ainda possível navegar entre as anotações existentes no documento, esta navegação é feita através de setas que permitem ir para a anotação anterior ou seguinte.

2.2.3 Genius

Apesar do **Genius**[5] ser uma plataforma só focada no mundo da música e os seus conteúdos serem letras de músicas, este proporciona uma experiência de utilização mais moderna e atual que as restantes plataformas.

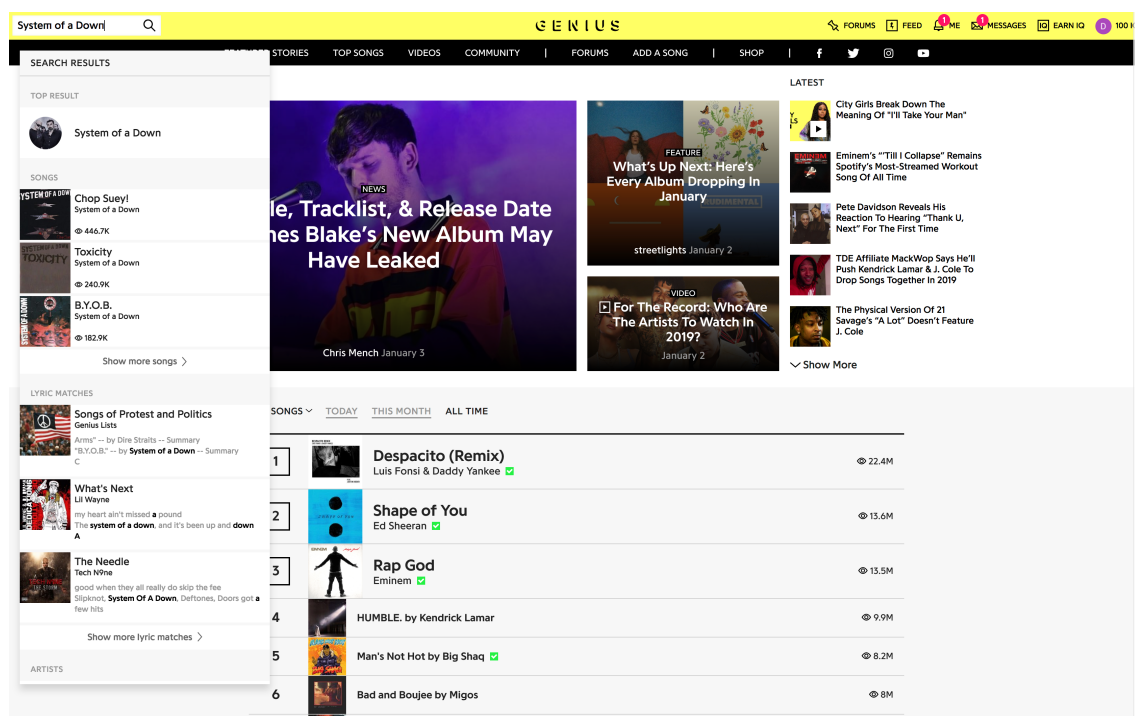


Figura 2.16: Genius - Listar/Pesquisar conteúdos

Para listar os conteúdos existe uma listagem na página inicial que pode ser configurada através do tipo de conteúdo que queremos ver. Inicialmente só mostra dez resultados, mas permite mostrar mais.

Quando se realiza uma pesquisa aparecem todos os tipos de conteúdos, músicas, letras que contêm a pesquisa, artistas e álbuns, disponíveis que de alguma forma se relacionam com a pesquisa, como mostrado na Figura 2.16 no canto superior esquerdo.

Edit Lyrics

[Verse 1: Daron Malakian]
 Dead men lying on the bottom of the grave
 Wondering when savior comes, is he going to be saved
 Maybe you're a sinner into your alternate life
 Maybe you're a joker, maybe you deserve to die

[Chorus: Serj Tankian & Daron Malakian]
 They were crying when their sons left
 God is wearing black
 He's gone so far to find no hope
 He's never coming back
 They were crying when their sons left
 All young men must go
 He's come so far to find the truth
 He's never going home

[Verse 2: Daron Malakian]
 Young men standing on the top of their own graves
 Wondering when Jesus comes, are they gonna be saved
 Cruelty to the winner, bishop tells the king his lies
 Maybe you're a mourner, maybe you deserve to die

[Chorus: Serj Tankian & Daron Malakian]
 They were crying when their sons left
 God is wearing black
 He's gone so far to find no hope
 He's never coming back
 They were crying when their sons left
 All young men must go

Soldier Side
 System Of A Down

Genius Annotation 1 contributor
 Even though leaders mourn the deaths of young soldiers they send into combat, maybe they should be the ones strapped with a rifle and fight their own wars.

Why DO they ALWAYS send the POOR?

Edit

Upvote +11

Suggest an improvement to earn IQ

Hagop019 · 4 months ago
 Its supposed to be rifle

0:00 -3:41 MUSIC

Figura 2.17: Genius - Visualizar Anotações

Para visualizar uma anotação basta carregar no texto que se encontra realçado a cinzento, que a anotação aparece do lado direito da letra, Figura 2.17. É ainda possível adicionar comentários na anotação ou no próprio conteúdo.

Search lyrics & more

GENIUS

FORUMS FEED ME MESSAGES EARN IQ 100 IQ

FEATURED STORIES TOP SONGS VIDEOS COMMUNITY FORUMS ADD A SONG SHOP

SYSTEM OF A DOWN **Stealing Society**
 System of a Down
 Produced by Daron Malakian & Rick Rubin
 Album Hypnotize

1 10.4K 12

Edit Lyrics

[Intro: Daron Malakian]
 Yeah!
 Alright!
 Yeah, alright!

Start the Genius Annotation (+5 IQ)

[Verse 1: Serj Tankian & Daron Malakian]
 Two skies watching it all, fading
 Two skies living it all, fading
 Two skies watching it all, fading
 Two skies fading, one's abating
 Two suns living it all, dying
 Two suns fighting, one's abiding
 Two suns watching them both, fighting
 Two suns seeing them both, dying
 Two suns watching them both, fighting
 Two suns fighting, one's abiding
 Two skies seeing them both, dying
 Two skies fading, one's abating

Stealing Society
 System Of A Down

0:00 -2:58 MUSIC

Figura 2.18: Genius - Criar Anotação

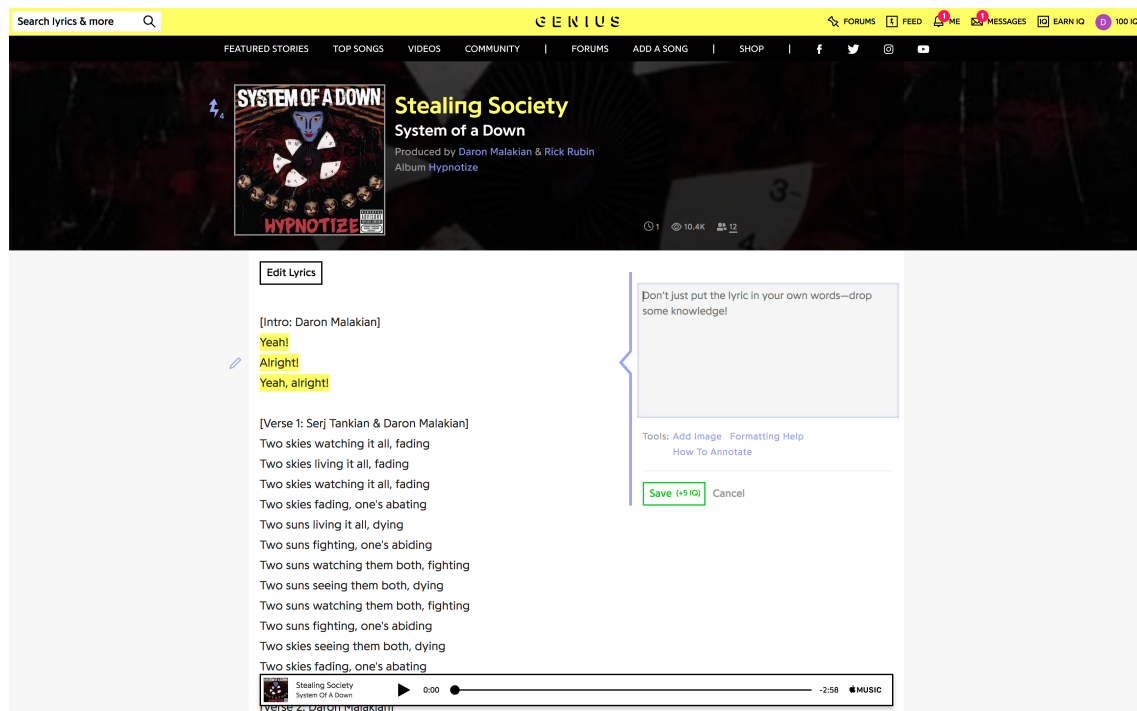













































Figura 2.19: Genius - Escrever nova Anotação

Para criar basta selecionar o texto que se pretende anotar e depois clicar em “Start the Genius Annotation”, Figura 2.18. Depois de clicar no botão aparece a caixa de texto, Figura 2.19, para criar a anotação, nesta caixa de texto é possível adicionar imagens e hiperligações.

É de notar que apesar de ser uma plataforma grátis, existe um forte uso de *Gamification* em que o utilizador ganha pontos por inserir conteúdo ou anotações na plataforma. Estes pontos permitem visualizar os utilizadores que mais contribuem para a plataforma no geral ou para um artista em específico.

2.3 Comparação das aplicações

Agora que já analisamos o produto atual e outras plataformas web, vamos fazer uma lista de funcionalidades e fazer uma comparação entre todas as plataformas analisadas. Vamos também adicionar o projeto que irá ser desenvolvido para perceber as diferenças e quais as vantagens ou desvantagens sobre os outros.

Plataforma Característica	Annotation Studio	Document Cloud	Genius	Anotado (Mobile)	Anotado (Web)
Login com Facebook ou Google					
Pesquisar Conteúdos ou Anotações					
Favoritos					
Adicionar Anotações					
Anotar com imagens, vídeo ou com formatações de texto					
Realçar área anotada					
Edição colaborativa					
Múltiplas anotações no mesmo texto					
Vender anotações					
Agrupamento de anotações					



- Característica inexistente



- Característica a funcionar parcialmente



- Característica a funcionar corretamente

Tabela 2.1: Tabela comparativa de características - Plataformas Web

Através da análise da Tabela 2.1 podemos concluir que a principal característica que destaca o nosso produto é o conjunto de funcionalidades relativo às anotações colaborativas e de venda das mesmas.

Uma outra funcionalidade que facilita a utilização da plataforma é o login através de outros serviços, neste caso Facebook ou Google, isto permite os utilizadores entrarem na plataforma mais facilmente e não os obriga a fazer o processo tradicional de registo e confirmação de email.

Apesar da funcionalidade de realçar a área anotada ser visualmente apelativa foi decidido não ser implementada no âmbito deste estágio por esta não acrescentar valor ao produto.

Os outros produtos não foram analisados em detalhe mas foi também realizada uma tabela com algumas das funcionalidades que são comuns a alguns dos produtos ou que são consideradas de importantes para uma aplicação de criação e partilha de anotações.

Plataforma / Característica	Acrobat Reader / XODO	Mendeley	Google Docs / Kami	Hypothesis	Anotado
Adicionar Anotações					
Anotar com imagens, vídeo ou com formatações de texto					
Realçar área anotada					
Edição colaborativa					
Múltiplas anotações no mesmo texto					
Vender anotações					
Agrupamento de anotações					

- Característica inexistente
- Característica a funcionar parcialmente
- Característica a funcionar corretamente

Tabela 2.2: Tabela comparativa de características - Aplicações de Leitura e Extensões

Como podemos ver pela Tabela 2.2 o principal foco destas aplicações é apenas a anotação de documentos ou páginas web externas, maioritariamente de forma individual.

Capítulo 3

Descrição do processo

Este capítulo tem como objetivo enumerar todos os passos aplicados no estágio e como estes se relacionam.

Este conjunto de passos tem como fim atingir um produto funcional e bem estruturado, utilizando boas práticas de desenvolvimento de software.

O primeiro passo realizado foi a **Abordagem ao conceito**, foi nesta fase que foram tomadas algumas das decisões que influenciaram os objetivos pretendidos para a aplicação e para o estágio, este passo é detalhado na **Secção 3.1**.

Depois de tomadas as decisões no primeiro passo passamos à definição das **Especificações do projeto**. Este passo tem como objetivo principal pegar nas conclusões e discussões do primeiro passo e definir o que é necessário fazer para o projeto se concretizar. É neste passo que definimos o modelo de desenvolvimento, os requisitos, as arquiteturas do sistema e as tecnologias a serem utilizadas. É o resultado desta fase que define como irá ser realizado todo o desenvolvimento do projeto. Esta etapa do processo é definida no **Capítulo 4**.

Depois de definidas as especificações do projeto passamos para a fase de **Desenvolvimento**. Esta fase é organizada seguindo o modelo de desenvolvimento definido na fase de especificações. É a fase que mais esforço requer pois é durante a mesma que é feita toda a implementação da aplicação. Esta fase do processo é detalhada no **Capítulo 5**.

Em paralelo com a fase de desenvolvimento é feita a **Verificação e validação** do projeto, à medida que se desenvolve uma funcionalidade realiza-se a verificação e validação da mesma. Esta fase existe para garantir o bom funcionamento do produto, ou seja, verificar que este funciona como é pretendido. Para ajudar na execução desta fase vamos recorrer principalmente a testes unitários e a testes de *Quality Assurance* (em português, Garantia de Qualidade). Todos os métodos e processos utilizados nesta fase são listados e detalhados no **Capítulo 6**.

3.1 Abordagem ao conceito

Este primeiro passo tem como objetivo definir o conceito do produto.

O foco principal deste passo é reunir as ideias dos elementos da equipa e procurar desenvolver estas de maneira a definir os objetivos principais do projeto e as suas funcionalidades.

O resultado deste passo vai ser o ponto de partida para definir o trabalho a desenvolver e

toda a informação obtida vai servir de base para algumas das fases seguintes.

Vamos então enumerar algumas das decisões tomadas antes de definir as especificações do projeto.

Dado que já existe uma aplicação móvel na linha de produtos Anotado[2], parte dos requisitos já se encontram bem definidos. Para além destes requisitos já existentes, tanto a RLS como os seus parceiros têm um papel importante no levantamento dos requisitos, sendo que estes podem surgir da necessidade de qualquer uma das partes.

A aplicação móvel apenas utiliza os serviços de pagamento disponibilizados pela *Play Store* e *App Store* e como estes não funcionam para aplicações web foi necessário encontrar alternativas. Uma das alternativas encontrada foi o PayPal[15], este é um serviço utilizado por grande parte dos gigantes de vendas online, como por exemplo o eBay. Foi ainda encontrada outra alternativa, o Stripe[16], que permite o pagamento direto através de cartões de crédito, este serviço é fácil de usar e ao contrário do PayPal não requer criação de uma conta.

Uma das principais funcionalidades pretendida é a criação de anotações de forma colaborativa. Esta funcionalidade levantou a questão de como seria realizada a gestão de versões de uma anotação quando editada por várias pessoas simultaneamente. Inicialmente pensou-se em permitir que várias pessoas pudessem editar uma anotação simultaneamente em tempo real, mas isso podia trazer alguns problemas de acesso concorrente, como por exemplo a validação final do texto criado. Pensou-se também em usar um sistema de “sugestões” em que um utilizador sugeria uma alteração e o criador da anotação poderia ou não aceitar, mantendo sempre o controlo total sobre o estado final da anotação. No fim, tendo em conta o valor que a funcionalidade daria ao produto, optou-se por permitir que qualquer utilizador edite a anotação, mas sendo sempre mantidas todas as versões anteriores na base de dados e associadas à versão atual.

Um outro problema discutido foi qual o melhor método de classificação de autores, anotações e bundles de anotações. Apesar desta funcionalidade ter sido discutida e se ter pensado em várias formas de a implementar, conclui-se que o pouco valor que acrescentaria ao produto, não justificava o esforço adicional.

3.2 Diagrama de Gantt

Para visualizar como estas fases se organizam ao longo dos dois semestres e qual o tempo alocado a cada uma das etapas vamos recorrer a um Diagrama de Gantt.

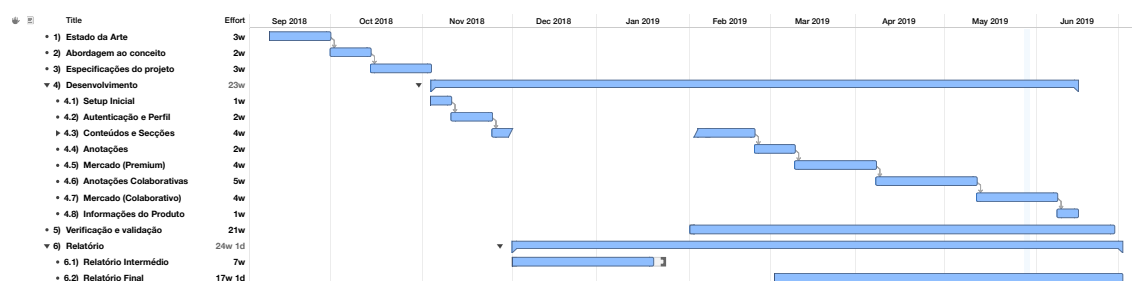


Figura 3.1: Diagrama de Gantt - Fases do projeto

Podemos observar que o desenvolvimento é a fase que requer mais esforço. É ainda possível

ver como é feita a divisão deste esforço e como é alocado a cada grupo de funcionalidades. A fase de desenvolvimento vai ser analisada com maior detalhe no Capítulo 5 - Secção 5.1.

Capítulo 4

Especificações do projeto

Este capítulo define um conjunto de diretrizes que devem ser seguidas para o desenvolvimento do projeto.

É neste capítulo que é apresentado o modelo de desenvolvimento de software escolhido, Secção 4.1, são enumeradas as funcionalidades a serem implementadas, Secção 4.2, a arquitetura do sistema, Secção 4.3 a as tecnologias a serem usadas, Secção 4.4.

Também são listados alguns dos desafios encontrados, Secção 4.5, e os possíveis riscos do projeto, Secção 4.6.

4.1 Modelo de desenvolvimento de software

O projeto será desenvolvido seguindo uma metodologia ágil, com algumas adaptações para funcionar com os outros processos internos da empresa, fortemente baseada em *Scrum*[17] e com algumas influências em *Kanban*[18]. Esta metodologia permite uma abordagem mais modular e aberta, ou seja, o projeto não necessita de ter os requisitos bem definidos e permite que ocorram alterações durante o desenvolvimento do mesmo.

As principais vantagens desta abordagem são as seguintes:

- Rápida implementação de funcionalidades, pois os *sprints* podem ser planeados em volta de uma funcionalidade específica.
- Aceita alterações a meio do desenvolvimento e facilita as mesmas.
- Apesar de ser uma abordagem dependente do cliente, neste caso não é relevante, pois o cliente é a própria empresa.

Como se trata de um projeto que o cliente pretende continuar a desenvolver/manter após o estágio, é relevante que este se adapte à estrutura e aos seus processos internos.

4.2 Requisitos

Nesta secção vamos enumerar todos os requisitos funcionais e não funcionais do sistema. O objetivo principal é listar e descrever todos os requisitos encontrados na abordagem ao

conceito anteriormente feita.

4.2.1 Requisitos funcionais

Os requisitos funcionais da aplicação podem ser divididos em seis grupos distintos:

- **Informações do Produto** - Todas as páginas de informação estática sobre o produto e a aplicação.
- **Autenticação e Perfil** - Todas as funcionalidades relacionadas com a criação de conta e autenticação na plataforma.
- **Conteúdos e Secções** - Funcionalidades relacionadas com a listagem e as interações possíveis com os conteúdos disponíveis.
- **Anotações** - Funcionalidades que dizem respeito à visualização e interação com anotações.
- **Anotações Colaborativas** - Funcionalidades de gestão dos grupos de anotações colaborativas, desde a criação dos grupos ao convite de novos utilizadores para ajudarem na criação/edição de anotações.
- **Mercado** - Funcionalidades de compra e venda de anotações. Este grupo pode ainda ser dividido em dois subgrupos: o **Mercado Premium**, anotações certificadas pela plataforma, e o **Mercado Colaborativo**, anotações criadas por qualquer utilizador da plataforma.

Estes requisitos podem ser consultados com algum detalhe e agrupados segundo os grupos anteriormente referidos no Anexo A.

4.2.2 Requisitos não-funcionais

Nesta secção vão ser listados os requisitos não-funcionais encontrados durante a fase de planeamento e que se adequam à natureza do produto. A lista de requisitos não-funcionais definida foi a seguinte:

- **ANTD-NF1: Segurança e Privacidade**
 - Apenas o utilizador deve poder consultar e alterar as suas informações pessoais.
 - **Medição:** Testes de penetração e testes gerais de segurança.
- **ANTD-NF2: Performance**
 - A aplicação web, numa ligação de internet com uma largura de banda de 20Mbps[19], não deve demorar mais que cinco segundos a carregar, sendo que nos primeiros dois segundos[20] já deve mostrar alguma informação.
 - O servidor não deve demorar mais que um segundo a responder a um pedido.
 - **Medição:** Testes de stress.

- **ANTD-NF3: Compatibilidade**

- A aplicação web deve ser compatível com os principais browsers de computador (mais de 5% de *market share*[21]).
- **Medição:** Testes de aceitação nos diferentes browsers.

- **ANTD-NF4: Usabilidade**

- Um utilizador desconhecido (estudante, advogado, etc) deve conseguir se registar e autenticar na plataforma em menos de um 1 minuto.
- Um utilizador novo autenticado com 5 minutos de utilização da aplicação deve ser capaz de executar as tarefas básicas da aplicação. As tarefas básicas são: aceder a conteúdos/secções/anotações, adicionar/consultar favoritos e criar as suas próprias anotações.
- **Medição:** Testes de usabilidade.

4.3 Arquiteturas do sistema

Nesta secção vamos definir as arquiteturas do sistema, servidor e aplicação web. Primeiro vamos definir a arquitetura do servidor analisando os seus utilizadores, serviços, conteúdos e funcionalidades. Seguidamente vamos analisar a estrutura da aplicação web e os componentes que a formam. Tudo isto é feito tendo em conta os requisitos previamente definidos.

4.3.1 Arquitetura do servidor

Para definir e analisar a arquitetura do servidor vai ser seguido o **modelo C4**[22]. Este modelo baseia-se em quatro diagramas para descrever o design e funcionamento do servidor. Cada diagrama tem um nível de profundidade diferente, começando pelo diagrama de contexto que permite visualizar os utilizadores e serviços/sistemas que o servidor vai possuir. O diagrama de *containers* já possui mais detalhe, neste diagrama já conseguimos definir as escolhas tecnológicas e os *containers* do sistema. Com o diagrama de componentes é possível perceber as principais funcionalidades do sistema. Por fim temos o diagrama de classes, que é opcional mas que permite ilustrar certos detalhes de implementação.

Primeiramente vamos definir e analisar o **diagrama de contexto**:

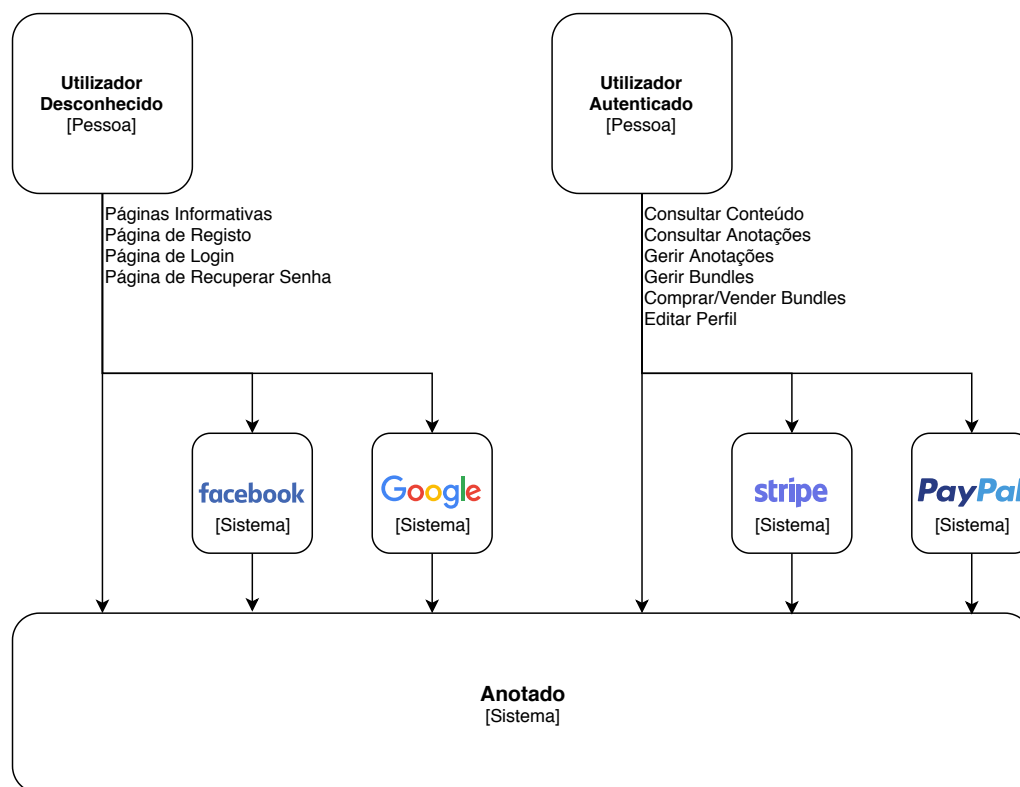


Figura 4.1: Diagrama de contexto

Através do diagrama de contexto, Figura 4.1, podemos observar que o nosso sistema tem dois tipos de utilizadores: os utilizadores desconhecidos que apenas têm acesso a funcionalidades de autenticação e informações gerais; e os utilizadores autenticados, estes têm

acesso a todas as funcionalidades do sistema.

Podemos ainda constatar que existem quatro serviços externos ao sistema, sendo dois deles utilizados para autenticação, Facebook[23] e Google[24], e os outros dois para fazer gestão de pagamentos, Stripe[16] e PayPal[15].

Após termos definido o diagrama de contexto vamos aprofundar mais o detalhe e definir o **diagrama de *containers***:

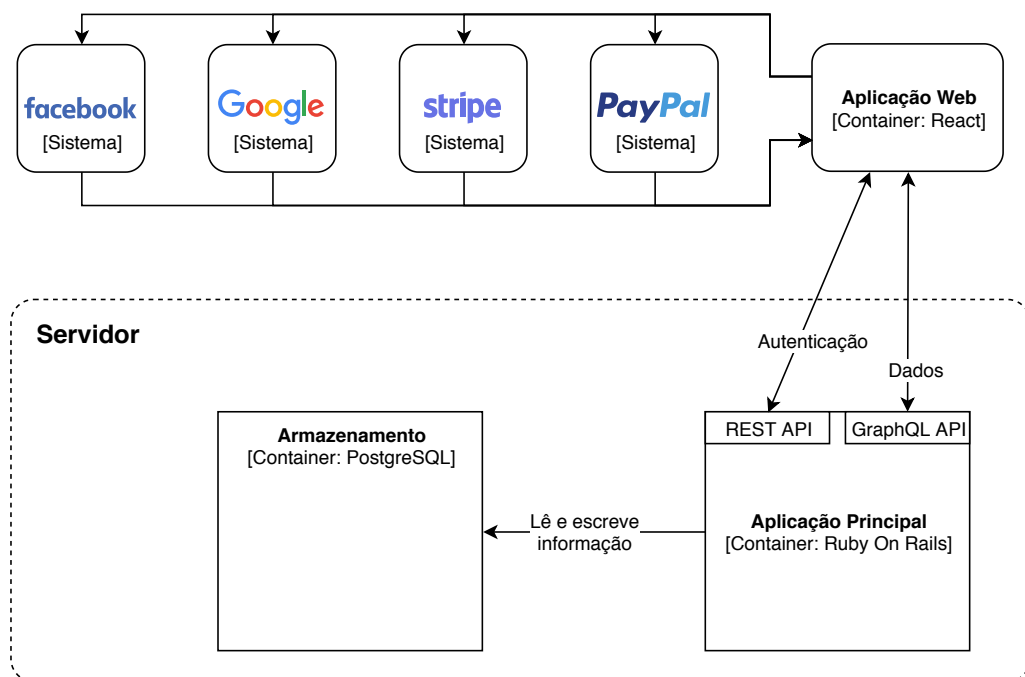


Figura 4.2: Diagrama de *containers*

Neste diagrama, Figura 4.2, podemos verificar que existem duas interfaces de comunicação com o servidor. Uma interface, a Representational State Transfer (REST) Application Programming Interface (API), apenas para pedidos relacionados com autenticação e a outra, GraphQL API, para todos os outros pedidos, passando por esta interface toda a informação relativa a conteúdos e anotações.

Para perceber melhor o funcionamento do servidor vamos definir e analisar o **diagrama de componentes**:

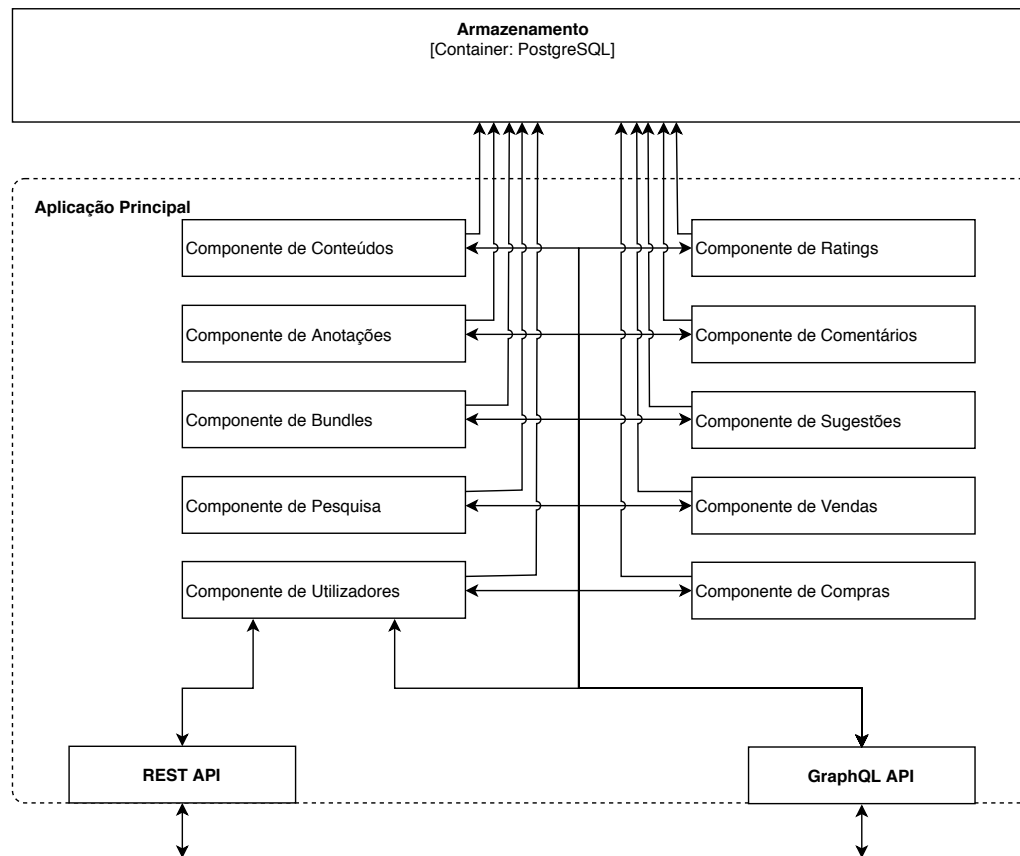


Figura 4.3: Diagrama de componentes

Através deste diagrama, Figura 4.3, conseguimos perceber melhor o funcionamento interno da aplicação principal do servidor. Esta aplicação é constituída por vários componentes que representam as principais funcionalidades do servidor.

É possível visualizar com melhor detalhe quais as funcionalidades de cada interface e que com componentes interage.

Podíamos ainda definir o diagrama de classes para detalhar alguma funcionalidade menos perceptível, mas neste caso vamos apenas detalhar as funcionalidades de cada componente:

- **Componente de Utilizadores:** Tem como função gerir as contas de utilizadores e as ações com estas relacionadas, desde do Login ao Editar Perfil.
- **Componente de Conteúdos:** Tem como função ler os conteúdos e respetivas secções da base de dados.
- **Componente de Anotações:** Tem como função ler, criar, editar e eliminar anotações da base de dados.
- **Componente de Bundles:** Tem como função ler, criar, editar e eliminar bundles da base de dados.
- **Componente de Pesquisa:** Tem como função procurar conteúdos, secções e anotações na base de dados.

- **Componente de Ratings:** Tem como função ler, criar, editar e eliminar os ratings de bundles da base de dados.
- **Componente de Comentários:** Tem como função ler, criar, editar e eliminar comentários a anotações da base de dados.
- **Componente de Sugestões:** Tem como função ler, criar, editar e eliminar sugestões a anotações da base de dados.
- **Componente de Vendas:** Tem como função ler, criar, editar e eliminar vendas da base de dados.
- **Componente de Compras:** Tem como função ler, criar, editar e eliminar compras da base de dados.

Para finalizar a arquitetura do servidor vamos definir e analisar o **diagrama de relacionamento de entidades**:

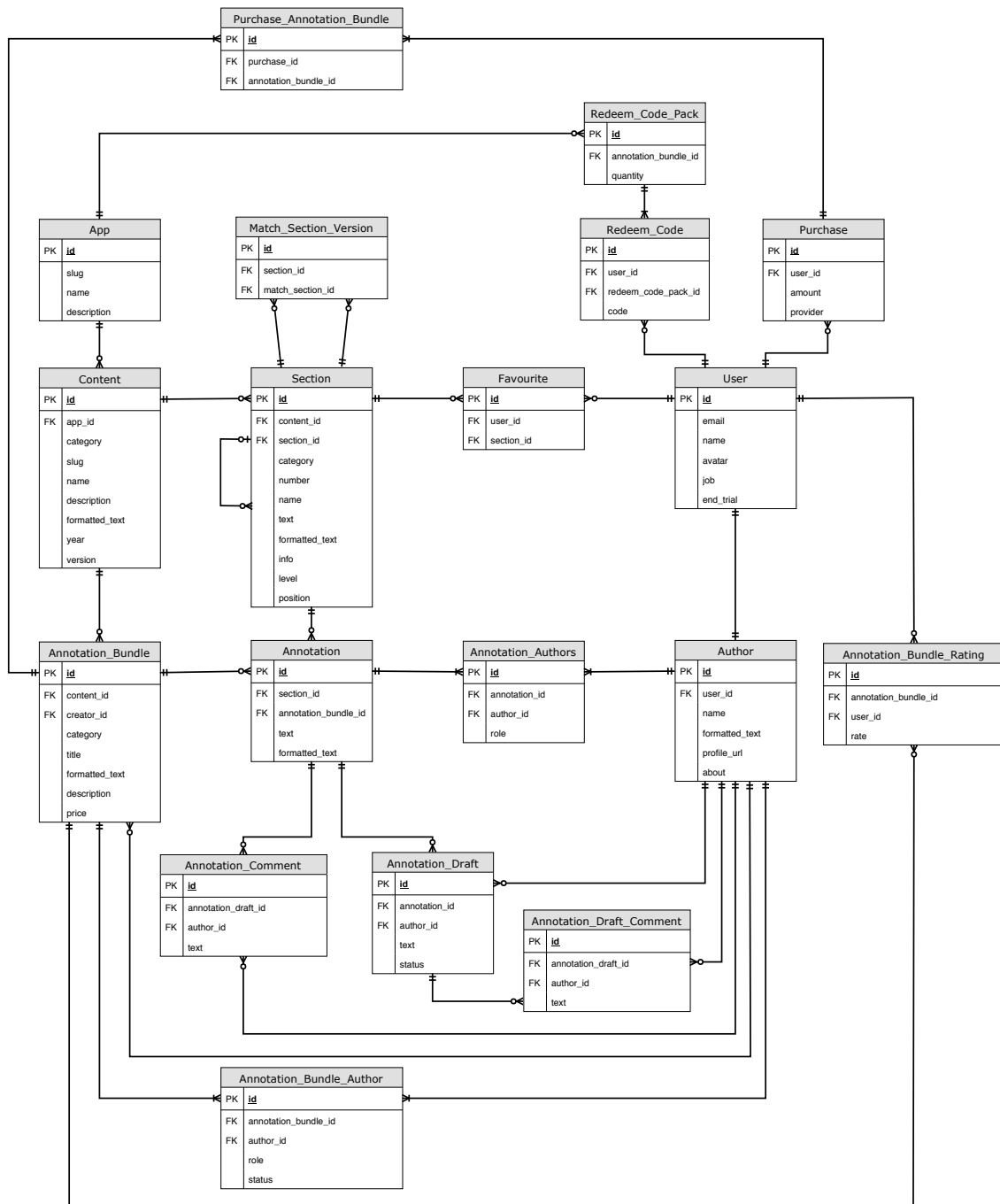


Figura 4.4: Diagrama de relacionamento de entidades

Com este diagrama, Figura 4.4, conseguimos perceber quais as entidades que fazem parte da estrutura da base de dados e como estas se relacionam.

- **User**:
 - Esta entidade representa um utilizador e é composta pelos seguintes campos: email, nome, foto, profissão e data do fim do período de experimentação.

- Está diretamente relacionada com:
 - * Favourite: um utilizador pode ter zero ou mais favoritos
 - * Purchase: um utilizador pode ter zero ou mais compras
 - * Redeem Code: um utilizador pode ter zero ou mais códigos de resgate
 - * Author: um utilizador tem um autor
 - * Annotation Bundle Rating: um utilizador pode classificar zero ou mais bundles de anotações
- **App**:
 - Esta entidade representa uma aplicação e é composta pelos seguintes campos: abreviatura, nome e descrição. Uma aplicação representa um conjunto de conteúdos, com esta entidade é possível agrupar os conteúdos.
 - Está diretamente relacionada com:
 - * Content: uma aplicação pode ter zero ou mais conteúdos
- **Content**:
 - Esta entidade representa um conteúdo e é composta pelos seguintes campos: categoria, abreviatura, nome, descrição, ano e versão.
 - Está diretamente relacionada com:
 - * App: um conteúdo pertence sempre a uma aplicação
 - * Annotation Bundle: um conteúdo pode ter zero ou mais bundles de anotações
- **Section**:
 - Esta entidade representa uma secção e é composta pelos seguintes campos: categoria, número, nome, texto, informação, nível e posição.
 - Está diretamente relacionada com:
 - * Content: uma secção pertence sempre a um conteúdo
 - * Section: uma secção pode ou não pertencer a outra secção
 - * Match Section: uma secção pode ter zero ou mais secções paralelas através da entidade Match Section Version
 - * Annotation: uma secção pode ter zero ou mais anotações
- **Author**:
 - Esta entidade representa um autor e é composta pelos seguintes campos: nome, perfil externo e descrição.
 - Está diretamente relacionada com:
 - * User: um autor pertence sempre a um utilizador
 - * Annotation: um autor pode ter zero ou mais anotações através da entidade Annotation Author
 - * Annotation Comment: um autor pode ter zero ou mais comentários numa anotação
 - * Annotation Draft: um autor pode ter zero ou mais sugestões numa anotação
 - * Annotation Draft Comment: um autor pode ter zero ou mais comentários numa sugestão
 - * Annotation Bundle:

- um autor pode criar zero ou mais bundles de anotações
- um autor pode ser autor de zero ou mais bundles de anotações através da entidade Annotation Bundle Author

● **Annotation Bundle:**

- Esta entidade representa um bundle de anotações e é composta pelos seguintes campos: categoria, título, descrição e preço.
- Está diretamente relacionada com:
 - * Content: um bundle de anotações pertence sempre a um conteúdo
 - * Annotation: um bundle de anotações pode ter zero ou mais anotações
 - * Author:
 - um bundle de anotações pertence a um autor
 - um bundle de anotações pode ter zero ou mais autores através da entidade Annotation Bundle Author
 - * Annotation Bundle Rating: um bundle de anotações pode ter zero ou mais classificações
 - * Purchase Annotation Bundle: um bundle de anotações pode ter zero ou mais compras associadas

● **Annotation:**

- Esta entidade representa uma anotação e é composta pelo seguinte campo: texto.
- Está diretamente relacionada com:
 - * Section: uma anotação pertence sempre a uma secção
 - * Annotation Bundle: uma anotação pertence sempre a um bundle de anotações
 - * Author: uma anotação pode ter um ou mais autores através da entidade Annotation Author, sendo que cada autor tem uma função
 - * Annotation Comment: uma anotação pode ter zero ou mais comentários
 - * Annotation Draft: uma anotação pode ter zero ou mais sugestões

● **Annotation Comment:**

- Esta entidade representa um comentário a uma anotação e é composta pelo seguinte campo: texto.
- Está diretamente relacionada com:
 - * Annotation: um comentário pertence sempre a uma anotação
 - * Author: um comentário pertence sempre a um autor

● **Annotation Draft:**

- Esta entidade representa uma sugestão e é composta pelos seguintes campos: texto e estado.
- Está diretamente relacionada com:
 - * Annotation: uma sugestão pertence sempre a uma anotação
 - * Author: uma sugestão pertence sempre a um autor
 - * Annotation Draft Comment: uma sugestão pode ter zero ou mais comentários

- **Annotation Draft Comment:**

- Esta entidade representa um comentário a uma sugestão e é composta pelo seguinte campo: texto.
- Está diretamente relacionada com:
 - * Annotation Draft: um comentário pertence sempre a uma sugestão
 - * Author: um comentário pertence sempre a um autor

- **Purchase:**

- Esta entidade representa uma compra e é composta pelos seguintes campos: valor e serviço.
- Está diretamente relacionada com:
 - * User: uma compra pertence sempre a um utilizador
 - * Annotation Bundle: uma compra pode ter um ou mais bundles de anotações associados

- **Redeem Code Pack:**

- Esta entidade representa um conjunto de códigos de resgate e é composta pelo seguinte campo: quantidade.
- Está diretamente relacionada com:
 - * App: um conjunto de códigos de resgate pertence sempre a uma aplicação
 - * Redeem Code: um conjunto de códigos de resgate pode ter um ou mais códigos de resgate associados

- **Redeem Code:**

- Esta entidade representa um código de resgate e é composta pelo seguinte campo: código.
- Está diretamente relacionada com:
 - * User: um código de resgate pode ou não pertencer a um utilizador
 - * Redeem Code Pack: um código de resgate pertence sempre a um conjunto de códigos de resgate

As entidades que são possíveis de pesquisar possuem ainda o campo *formatted_text* que apenas possui o texto sem nenhuma acentuação, isto faz com que as ações de pesquisa sejam mais eficientes e obtenham melhores resultados.

4.3.2 Arquitetura da aplicação web

Agora que já definimos e analisámos a arquitetura do servidor e as suas funcionalidades, podemos agora analisar a arquitetura da aplicação web. Para tal vamos criar um diagrama com os componentes que vão constituir esta aplicação.

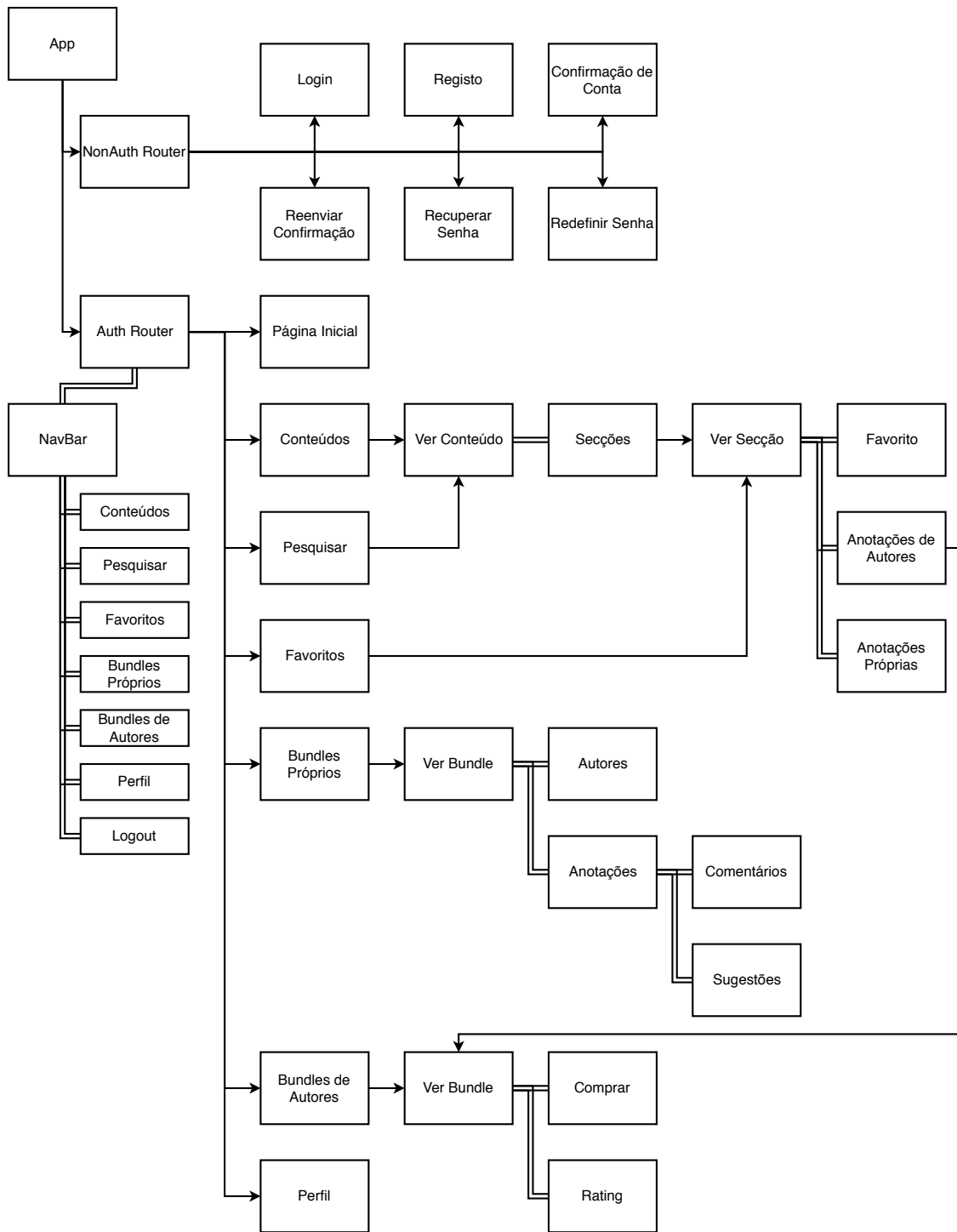


Figura 4.5: Diagrama da aplicação web

Este diagrama, Figura 4.5, representa a aplicação web e como esta se encontra organizada

Para perceber melhor o diagrama existem alguns pontos a ter em conta:

- Cada caixa representa um componente, que por sua vez podem representar uma página da aplicação web ou um elemento que é carregado por essa página.
- Existem dois tipos de ligações:

- As ligações condicionais, representadas pela seta simples. No caso de existir mais que uma ligação possível apenas uma delas vai ser apresentada. Por exemplo, no caso do **Auth Router** apenas uma das sete opções vai ser carregada dependendo do URL.
- As ligações simultâneas, representadas pela dupla ligação. Ao contrário da ligação anterior esta significa que todos os componentes são carregados na página. Por exemplo, quando o **Auth Router** é carregado o menu de navegação, **Navbar**, também é carregado e conseqüentemente todas as suas opções.
- O componente **App** é a base da aplicação. É responsável por inicializar a aplicação com todas as configurações definidas.
- Para gerir o *routing* da aplicação são usados dois componentes:
 - O **NonAuth Router** é responsável por gerir que páginas um utilizador não autenticado pode aceder. Caso um utilizador autenticado tente aceder alguma destas páginas é redirecionado para a página inicial.
 - O **Auth Router** à semelhança do anterior faz a gestão das páginas a que um utilizador autenticado pode aceder. Caso um utilizador não autenticado tente aceder alguma das páginas que requerem autenticação é redirecionado para a página de login.

Agora que já percebemos como o diagrama está definido, vamos detalhar as funções dos componentes principais:

- **App**: Componente base da aplicação. Responsável pela configurações iniciais de todas as ferramentas usadas.
- **NonAuth Router**: Responsável pelo *routing* das páginas que não requerem autenticação.
- **Login**: Página com um formulário que permite a um utilizador não autenticado se autenticar.
- **Registo**: Página que permite a um utilizador não autenticado criar uma nova conta.
- **Confirmação de Conta**: Página para a qual o utilizador é redirecionado através do link com um token (*/confirm/t0k3n*) presente no email de confirmação de conta, sendo depois confirmada a conta.
- **Reenviar Confirmação**: Página que permite o utilizador pedir um novo email de confirmação.
- **Recuperar Senha**: Página que permite o utilizador pedir um email para recuperar a senha.
- **Redefinir Senha**: Página para a qual o utilizador é redirecionado através do link com um token (*/reset/t0k3n*) presente no email de recuperação de senha, sendo depois possível redefinir a senha através de um formulário.
- **Auth Router**: Responsável pelo *routing* das páginas que requerem autenticação.
- **NavBar**: Este componente é carregado em todas as páginas que requerem autenticação e é através dele que o utilizador navega pela aplicação.

- **Conteúdos:** Componente responsável por navegar pelos conteúdos existentes no sistema e consequentemente aceder e consultar os mesmos.
- **Pesquisar:** Componente responsável pela pesquisa de conteúdos, secções ou anotações do sistema e listagem dos resultados.
- **Favoritos:** Página que lista todas as secções que o utilizador adicionou aos favoritos.
- **Bundles Próprios:** Página que permite fazer gestão dos bundles próprios ou a que pertence.
- **Bundles de Autores:** Página que lista todos os bundles que são possíveis de comprar e que permite depois aceder aos mesmos.
- **Perfil:** Página que mostra as informações do utilizador e permite editar essa mesma informação.

4.4 Tecnologias e ferramentas

Nesta secção são enumeradas as escolhas tecnológicas tomadas no planeamento do projeto. A escolha das principais tecnologias é explicada na secção ??.

4.4.1 Servidor

Na tabela 4.1 são listadas todas as tecnologias usadas para o desenvolvimento do servidor.

Nome	Função Principal	Descrição
Ruby [25]	Linguagem de programação do servidor	Ruby é uma linguagem de programação focada na simplicidade e na produtividade.
Ruby on Rails [26]	<i>Framework</i> base do servidor	Ruby on Rails é uma <i>framework</i> de Ruby que simplifica o desenvolvimento de aplicações web e APIs.
PostgreSQL [27]	Gestão e armazenamento de dados	PostgreSQL é um sistema de gestão de base de dados relacionais.
Devise [28]	Gestão e autenticação de utilizadores	Devise é uma <i>gem</i> que adiciona todas as funcionalidades necessárias para gerir e autenticar utilizadores numa aplicação de Ruby on Rails.
Active Admin [29]	Gestão de dados pelo administrador	Active Admin é uma <i>gem</i> que adiciona um sistema de administração a uma aplicação de Ruby on Rails. Esta <i>gem</i> através de uma interface gráfica permite ao administrador visualizar, criar, editar ou apagar dados da aplicação.
GraphQL [30]	Criar APIs	GraphQL é uma <i>query language</i> para APIs. O servidor disponibiliza um esquema com todas as entidades e relações que permite ao cliente pedir apenas a informação necessária.
GraphQL Ruby [31]	Gestão de pedidos GraphQL	GraphQL Ruby é uma <i>gem</i> que permite gerir os pedidos GraphQL feitos ao servidor.

Tabela 4.1: Lista e descrição das de tecnologias do servidor

4.4.2 Aplicação web

Na tabela 4.2 são listadas todas as tecnologias usadas para o desenvolvimento da aplicação web.

Nome	Função Principal	Descrição
JavaScript[32]	Linguagem de programação da aplicação web	JavaScript é uma linguagem de programação, sendo mais conhecida como a linguagem de <i>scripting</i> para páginas web.
React[33]	<i>Library</i> base da aplicação web	React é uma <i>library</i> de JavaScript que facilita a criação de interfaces interativas.
Apollo Client[34]	Gestão de pedidos GraphQL	Apollo Client é uma <i>library</i> que facilita a realização e gestão de pedidos GraphQL ao servidor. Permite ainda trabalhar os dados, armazená-los localmente, entre outras funcionalidades.
React Router[35]	Gestão de <i>routing</i> da aplicação web	React Router é uma <i>library</i> que permite gerir e definir todo o <i>routing</i> da aplicação web. Facilmente se define o que deve aparecer num determinado URL.

Tabela 4.2: Lista e descrição das tecnologias da aplicação web

4.4.3 Ferramentas

Dado a complexidade das fases de planeamento e desenvolvimento do projeto foram utilizadas várias ferramentas para facilitar certos processos e ajudar na organização e gestão dos mesmos. Parte destas ferramentas já eram de conhecimento do aluno, apenas algumas foram sugeridas pela empresa, que as usa no dia-a-dia.

Na tabela 4.3 são listadas e descritas todas as ferramentas usadas durante o estágio.

Nome	Função Principal	Descrição
Git [36]	Controlo de versões	Git é uma ferramenta que permite controlar as modificações feitas ao código do projeto. Essas modificações ficam guardadas e associadas a uma versão sendo possível voltar atrás na versão ou rever as diferenças entre duas versões.
Jira [37]	Organização de tarefas	O Jira permite a equipa organizar as tarefas por colunas tendo em conta o estado de cada uma. Permite ainda de forma fácil gerir e debater sobre cada tarefa.
Slack [38]	Comunicação	O Slack é a principal ferramenta de comunicação entre a equipa.
ESLint [39]	Regras e padrões de código	O ESLint é uma ferramenta que analisa código de JavaScript seguindo um conjunto de regras e padrões. Esta ferramenta é usada pelo programador para corrigir problemas de código.
Rubocop [40]	Regras e padrões de código	À semelhança do ESLint o Rubocop é uma ferramenta de análise de código só que específica para Ruby.
Jest [41]	Realização de testes	O Jest é uma ferramenta que permite criar e realizar testes ao código criado em JavaScript. Esta ferramenta permite ainda gerar uma relatório da cobertura de código testado.
RSpec [42]	Realização de testes	Tal como o Jest, o RSpec é uma ferramenta de testes, mas desenhada para Ruby on Rails. Mas ao contrário do Jest não tem a funcionalidade de gerar relatórios.

Tabela 4.3: Lista e descrição das de ferramentas

4.5 Desafios tecnológicos

Nesta secção vamos detalhar alguns dos principais desafios tecnológicos encontrados durante a fase de planeamento do projeto e algumas das razões que levaram a escolher as tecnologias anteriormente listadas. Serão também listadas as principais razões que levaram a escolher as tecnologias principais escolhidas para o servidor e para a aplicação web.

Dado que o produto já possui um servidor implementado, que gere os utilizadores e as compras da aplicação móvel, é de interesse do cliente manter um servidor comum às duas aplicações, web e móvel. Este servidor encontra-se implementado em **Ruby On Rails (RoR)**.

Apesar do aluno já ter alguma experiência nesta tecnologia, a curva da aprendizagem da mesma é relativamente baixa quando comparada com outras tecnologias semelhantes, como por exemplo o *Django* ou *Flask* para *Python* e *Spark* ou *Play* para *Java*.

Para manter as funcionalidades de ambas as aplicações num só servidor e continuar a suportar as já existentes decidiu-se manter as tecnologias já usadas para o servidor.

Contrariamente ao servidor, não existe nada implementado para a aplicação web, logo a escolha da tecnologia é mais aberta. Podíamos recorrer ao próprio **RoR** ou então usar uma ferramenta externa ao servidor, neste caso foi tido em conta o **React** que é uma *framework* de *JavaScript*.

Apesar do uso do **RoR** para servir as páginas web ser mais fácil e rápido de implementar, não possui a flexibilidade de usar uma ferramenta externa, como o *JavaScript*.

Dentro do espectro de *frameworks* existentes para *JavaScript* toda a equipa, incluindo o aluno, tinha experiência em **React**. O React permite criar páginas muito dinâmicas com interações mais complexas do que usando o **RoR**. Para além desta possibilidade de criar interfaces interativas, uma das principais vantagens é o facto de se basear em componentes, estes componentes são pedaços de código que podem ser reutilizados em vários ecrãs da aplicação, isto permite e facilita a reutilização de código. Dado estas vantagens foi fácil escolher o **React** como principal tecnologia da aplicação web.

Depois de escolhidas as tecnologias foi sugerido usar **GraphQL** para troca de dados entre o servidor e a aplicação web face ao clássico **REST**.

Ambas as tecnologias são de fácil implementação, tanto no servidor como na aplicação web, e possuem várias vantagens e desvantagens.[43]

A grande vantagem encontrada do **GraphQL** face ao **REST** é que o espaço de dados possíveis de aceder no **REST** está limitado a uma lista predefinida de *endpoints* enquanto no **GraphQL** apenas existe um *endpoint* e o espaço de dados é definido através de um esquema de relações, como numa base de dados. Ou seja, isto possibilita que os pedidos sejam modulares e flexíveis, o cliente apenas pede os campos que precisa daquele recurso, o que permite gerir o tráfego de informação entre servidor e cliente. Estas relações permitem ainda fazer pedidos complexos com agrupamento de dados através das relações existentes, mantendo a modularidade e flexibilidade.

Para demonstrar estas vantagens podemos analisar o seguinte exemplo, em que se pretende listar um utilizador e as secções que tem na sua lista de favoritos:

REST

• Pedidos

```
GET /user?id=1
GET /bookmarks?user_id=1
GET /section?id=1
GET /section?id=2
```

• Respostas

```
{
  user: {
    id: 1,
    name: 'John',
    email: 'john@email.com',
    age: 41,
    about: 'Hello',
    picture: 'selfie.png',
  }
}
```

```
{
  bookmarks: [
    {
      id: 1,
      section_id: 1,
    },
    {
      id: 2,
      section_id: 2,
    }
  ],
}
```

```
{
  section: {
    id: 1,
    name: 'Section One',
    text: 'Lorem ipsum.',
    description: 'First',
    number: 1,
  }
}
```

```
{
  section: {
    id: 2,
    name: 'Section Two',
    text: 'Lorem ipsumt.',
    description: 'Second',
    number: 2,
  }
}
```

GraphQL

• Pedido

```
query getUserById(id: 1){
  user {
    id
    name
    picture
    bookmarks {
      id
      section {
        id
        name
      }
    }
  }
}
```

• Resposta

```
{
  user: {
    id: 1,
    name: 'John',
    picture: 'selfie.png',
    bookmarks: [
      {
        id: 1,
        section: {
          id: 1,
          name: 'Section One',
        }
      },
      {
        id: 2,
        section: {
          id: 2,
          name: 'Section Two',
        }
      }
    ],
  }
}
```

Figura 4.6: REST vs GraphQL

Como podemos observar pelo exemplo, Figura 4.6, com o **GraphQL** conseguimos diminuir o número de pedidos efetuados e reduzir a quantidade de informação que o cliente recebe.

Depois desta análise comparativa das duas técnicas, o uso do **GraphQL** pareceu o mais indicado dado apresentar várias vantagens face ao **REST**.

Outro desafio encontrado e talvez o mais crítico para o desenvolvimento do projeto foi a integração de serviços de pagamento. Nem o aluno nem a equipa tinham experiência de desenvolvimento com estes sistemas.

4.6 Riscos e estratégias de mitigação

Dada a natureza do projeto numa fase inicial foi importante tentar antever possíveis problemas que pudessem ocorrer durante o desenvolvimento do mesmo. Para evitar tais problemas foram analisados os possíveis riscos e como os podemos resolver antecipadamente.

Tendo isto em conta, na fase inicial do projeto foram analisados e listados os riscos possíveis de acontecerem durante o desenvolvimento, caracterizando estes tendo em conta o seu impacto e probabilidade, seguindo os seguintes parâmetros:

- **Probabilidade:**
 - **Baixa:** menor que 40 %
 - **Média:** entre 40% a 70%
 - **Alta:** mais que 70 %
- **Impacto:**
 - **Marginal:** Dificulta o desenvolvimento do projeto, mas não afeta o produto final.
 - **Crítico:** Também dificulta o desenvolvimento do projeto e afeta o produto final, podendo não ficar como pretendido.
 - **Catastrófico:** Pode impedir a finalização do projeto.

A análise feita levou a que fossem levantados os seguintes riscos:

Risco #1 - Incumprimento do processo

- **Identificador:** R-1
- **Nome:** Incumprimento do processo
- **Fonte:** O proceso utilizado pelo cliente (*Agile*)
- **Descrição:** O aluno tem pouca experiência a trabalhar num ambiente em que os processos são aplicados rigorosamente. As falhas no processo podem levar a que certas tarefas não sejam completas.
- **Probabilidade:** Baixa
- **Impacto:** Crítico
- **Estratégia de Mitigação:** O aluno inicialmente irá receber um mentor para o ambientar sobre os processos utilizados na empresa e no projeto. Durante o desenvolvimento este mentor esporadicamente irá confirmar que o os processos internos estão a ser seguidos e utilizados de forma correta.

Risco #2 - Dificuldade de implementação com GraphQL

- **Identificador:** R-2
- **Nome:** Dificuldade de implementação com GraphQL
- **Fonte:** Utilização de GraphQL no servidor e na aplicação web
- **Descrição:** O aluno nunca teve contacto com a tecnologia em questão. Dado esta falta de experiência podem surgir atrasos ou até mesmo ser impossível a implementação de certas funcionalidades.
- **Probabilidade:** Média
- **Impacto:** Crítico
- **Estratégia de Mitigação:** Antes de começar a implementação o aluno deve dedicar tempo a ler e perceber a documentação de GraphQL disponível tanto para o servidor como para a aplicação web. E durante a implementação caso surjam dúvidas a equipa da RLS está sempre disponível a tirar dúvidas.

Risco #3 - Problemas a integrar os serviços de pagamento

- **Identificador:** R-3
- **Nome:** Problemas a integrar os serviços de pagamento
- **Fonte:** É necessário integrar a aplicação com serviços de pagamento
- **Descrição:** Tanto o aluno como a equipa da RLS nunca implementaram os serviços de pagamento com as tecnologias utilizadas a nível de servidor e aplicação web. A não ou má implementação destes serviços põe em causa a principal funcionalidade do sistema, que é o mercado.
- **Probabilidade:** Alta
- **Impacto:** Catastrófico
- **Estratégia de Mitigação:** O aluno deve analisar com cuidado os métodos de pagamento escolhidos e como estes devem ser implementados para as tecnologias em escolha, para tal deve ler a documentação existente de ambos os serviços escolhidos.

Para perceber melhor a dimensão e o impacto dos riscos vamos os dispor numa tabela:

Impacto \ Probabilidade	Marginal	Crítico	Catastrófico
Baixa		R-1	
Média		R-2	
Alta			R-3

Tabela 4.4: Tabela inicial de riscos e suas características

Como podemos observar pela tabela 4.4 todos os riscos apresentam um impacto igual ou superior a **crítico** para resolver isto e tentar reduzir a probabilidade de ocorrência ou até mesmo eliminar o risco foram aplicadas as estratégias de mitigação definidas para cada risco.

1. **Incumprimento do processo:** A estratégia de mitigação descrita anteriormente foi aplicada e como tal o aluno já percebe como funcionam os processos da realização do projeto naquela empresa. Como resultado o risco foi eliminado.
2. **Dificuldade de implementação com GraphQL:** O aluno aplicou a estratégia de mitigação e já percebe o funcionamento base do GraphQL e como realizar a sua implementação, tanto na parte do servidor como na da aplicação web. Este risco passou de **Médio/Crítico** para **Baixo/Marginal**.
3. **Problemas a integrar os serviços de pagamento:** Este risco ainda não foi totalmente controlado, mas o aluno já começou a aplicar a estratégia de mitigação e já tem algum conhecimento que o permite integrar com os serviços de pagamento pretendidos. Este risco passou de **Alto/Catastrófico** para **Médio/Crítico**.

Depois de aplicadas as estratégias de mitigação dispomos novamente os riscos na tabela:

Impacto \ Probabilidade	Marginal	Crítico	Catastrófico
Baixa	R-2		
Média		R-3	
Alta			

Tabela 4.5: Tabela final de riscos e suas características

Como podemos ver pela tabela 4.5 apesar das estratégias de mitigação terem funcionado, continuou a existir um risco crítico com uma probabilidade média e outro com impacto marginal e probabilidade baixa. Apesar de dois dos riscos se terem mantido, estes não foram prejudiciais para a finalização do projeto.

Capítulo 5

Desenvolvimento

Neste capítulo vamos detalhar o processo de desenvolvimento usado e as ferramentas utilizadas na gestão e organização do mesmo.

Na Secção 5.1 iremos perceber como foi organizado o desenvolvimento do projeto e na Secção 5.2 é detalhado todo o processo de desenvolvimento pelo qual uma tarefa de desenvolvimento passa desde da sua definição até estar finalizada.

5.1 Sprints

Inicialmente o desenvolvimento foi dividido em *Sprints*.

Um *Sprint* é constituído por um conjunto de tarefas a serem efetuadas num determinado intervalo de tempo. O projeto foi dividido em *Sprints* tendo em conta grupos de funcionalidades relacionadas. A duração de um *Sprint* foi inicialmente definida como sendo uma semana. Todos os dias era realizada uma reunião, com toda a equipa, em que era reportado o estado do desenvolvimento e sempre que necessário esclarecimento de pequenas dúvidas. Para além destas reuniões diárias foram ainda realizadas outras reuniões com o objetivo de mostrar o estado da aplicação à empresa e discutir eventuais alterações no projeto.

O desenvolvimento dos grupos de funcionalidades foi organizado num total de 23 *Sprints* da seguinte maneira:

1. Setup Inicial

- Descrição: Configuração inicial do servidor e da aplicação web. Criação da base de dados e respetivos modelos. Instalação das ferramentas de testes.
- Duração: 1 *Sprint*
- Data de início: 05/11/2018
- Data de fim: 09/11/2018

2. Autenticação e Perfil

- Descrição: Implementação das funcionalidades de autenticação e visualização de perfil.
- Duração: 2 *Sprints*
- Data de início: 12/11/2018

- Data de fim: 23/11/2018

3. Conteúdos e Secções

- Descrição: Implementação das funcionalidades de visualização de conteúdos e de interações com os mesmos.
- Duração: 4 *Sprints*
- Data de início: 26/11/2018 (Ocorreu uma interrupção devido à meta intermédia)
- Data de fim: 22/02/2019

4. Anotações

- Descrição: Implementação das funcionalidades de visualização e edição de anotações.
- Duração: 2 *Sprints*
- Data de início: 25/02/2019
- Data de fim: 08/03/2019

5. Mercado (Premium)

- Descrição: Implementação das funcionalidades das compras *premium*. Integração com os serviços externos PayPal[15] e Stripe[16].
- Duração: 4 *Sprints*
- Data de início: 11/03/2019
- Data de fim: 05/04/2019

6. Anotações Colaborativas

- Descrição: Implementação das funcionalidades de criação e edição dos *bundles* colaborativos. Implementação do sistema de convites de autores através do email. Implementação da criação, edição e gestão de versões de anotações colaborativas.
- Duração: 5 *Sprints*
- Data de início: 08/04/2019
- Data de fim: 10/05/2019

7. Mercado (Colaborativo)

- Descrição: Implementação das funcionalidades de venda e compra de *bundles* colaborativos. Integração com os serviços externos PayPal[15] e Stripe[16].
- Duração: 4 *Sprints*
- Data de início: 13/05/2019
- Data de fim: 07/06/2019

8. Informações do Produto

- Descrição: Implementação das páginas com informações relativas ao produto.
- Duração: 1 *Sprint*
- Data de início: 10/06/2019
- Data de fim: 14/06/2019

No Diagrama de Gantt da Figura 5.1 é possível visualizar a organização das *Sprints* ao longo do tempo e quando ocorreram as reuniões de *showcase*.

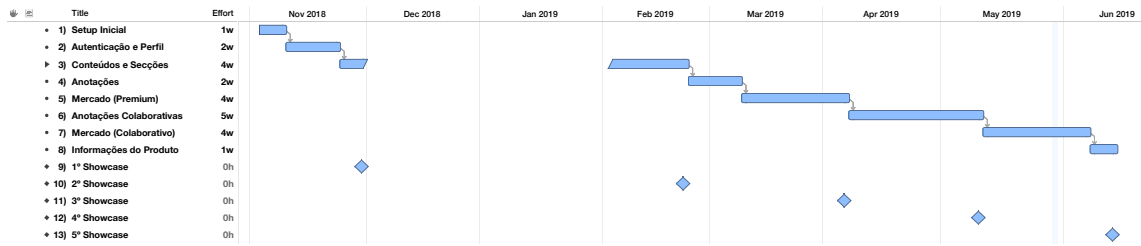


Figura 5.1: Diagrama de Gantt dos *Sprints*

5.2 Metodologia

Para ajudar na gestão e visualização da metodologia escolhida recorreremos ao JIRA[37]. Esta ferramenta permite alocar as tarefas a colunas que representam o seu estado, de acordo com a metodologia *Kanban*[18].

Na Figura 5.2 podemos observar a página de uma tarefa, nessa página é possível ver várias informações relativas a essa mesma tarefa, nome, descrição, comentários, estado, pessoa responsável e entre outros campos.

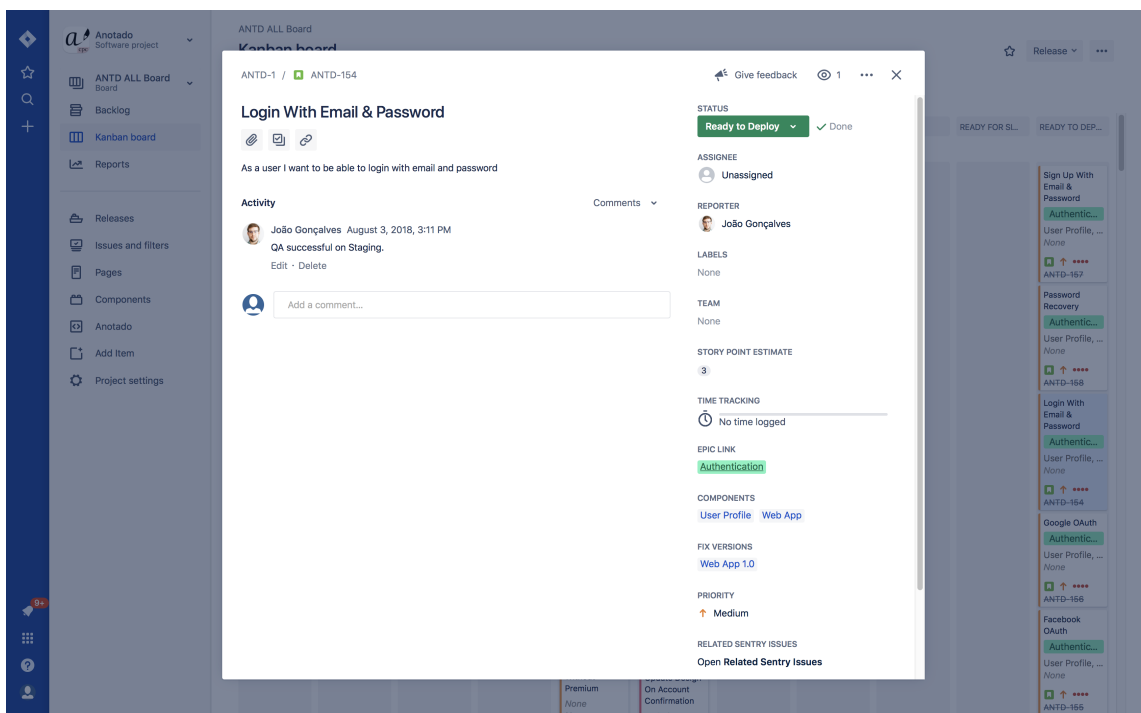


Figura 5.2: Exemplo de visualização de uma tarefa no JIRA

Como dito anteriormente, as tarefas são organizadas em colunas, como é visível na Figura 5.3. Cada coluna corresponde a um estado da tarefa, para além das colunas visíveis, existe também uma lista, “Backlog”, com todas as tarefas por desenvolver dos próximos *sprints*.

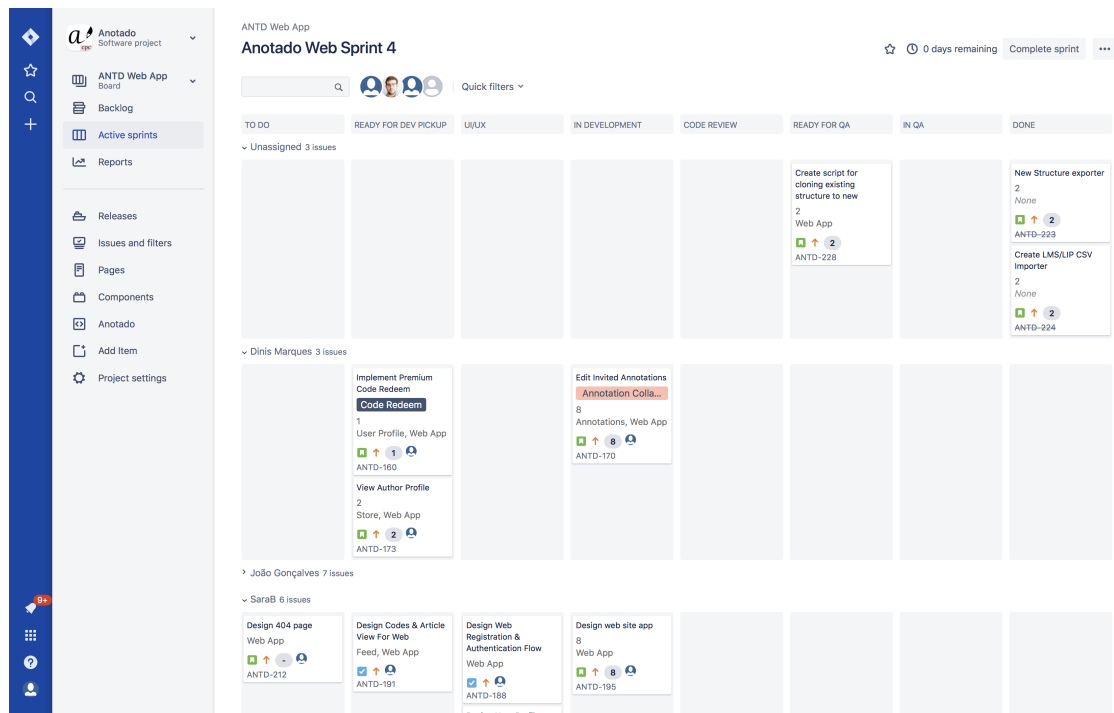


Figura 5.3: JIRA - Exemplo da ferramenta de gestão de tarefas

Para perceber o significado de cada coluna e como as tarefas são associadas a cada um dos estados vamos enumerar cada coluna, explicar o seu significado e como as tarefas são associadas a essa coluna.

0. **“Backlog”**: Lista de tarefas já definidas e por desenvolver que não fazem parte do *sprint* atual.
1. **“To Do”**: Caso alguma tarefa precise de ser repensada ou debatida é colocada nesta coluna.
2. **“Ready For Dev Pickup”**: Nesta coluna encontram-se todas as tarefas que devem ser desenvolvidas no *sprint* atual.
3. **“UI/UX”**: Esta coluna é utilizada pelos *designers* quando estão a desenhar os ecrãs de uma tarefa.
4. **“In Development”**: Uma tarefa encontra-se nesta coluna enquanto está a ser desenvolvida. No final deste desenvolvimento é criado um *merge request* e a tarefa passa para “Code Review”.
5. **“Code Review”**: Neste estado o elemento responsável pela tarefa delega, idealmente, dois membros da equipa para fazerem análise do código submetido no *merge request*. Caso o código seja aprovado pelos elementos a tarefa passa para a coluna “Ready For QA”, caso contrário, esta volta para a coluna “Ready For Dev Pickup”.
6. **“Ready For QA”**: As tarefas nesta coluna estão prontas a ser testadas.
7. **“In QA”**: Quando se realizam testes de Controlo de Qualidade a uma tarefa, esta é colocada nesta coluna. Caso a tarefa passe os testes é colocada na coluna “Done”, caso contrário, é colocada na coluna “Ready For Dev Pickup”.

8. **“Done”**: Estado final de uma tarefa. A tarefa é considerada completa quando atinge este estado. Se chegou a este passo é sinal que está pronta a ser utilizada no servidor de produção, contudo, podem sempre haver defeitos ou falhas que passem nos testes, sendo estes só encontrados futuramente.

Através da Figura 5.4 podemos perceber melhor como funciona este processo e qual o ciclo de vida de uma tarefa.

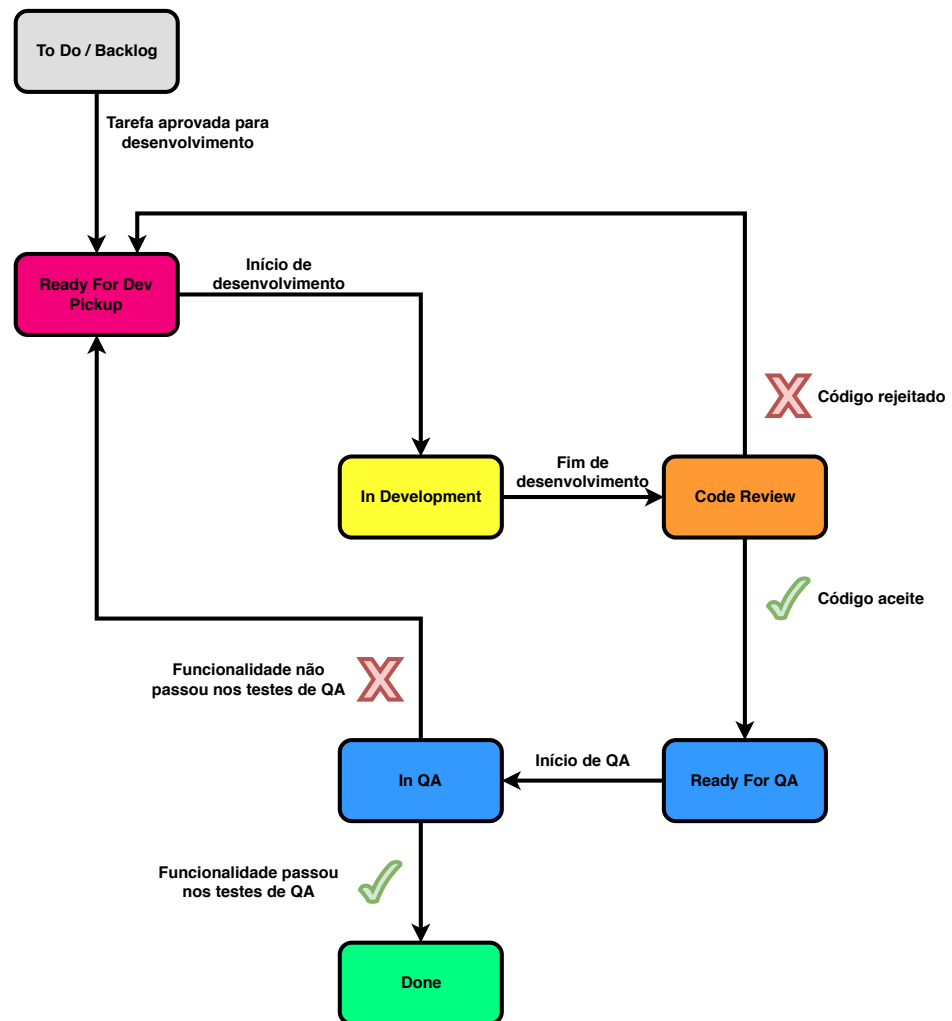


Figura 5.4: Diagrama do processo de desenvolvimento

Capítulo 6

Verificação e validação

Neste capítulo vamos listar todos os métodos utilizados no processo de verificação e validação do projeto.

Para controlar a qualidade do projeto foram utilizados vários processos e ferramentas, algumas automáticas, que facilitam a detecção de erros ou comportamentos inesperados.

A verificação e validação de uma tarefa começam pela **Indentação de Código**, Secção 6.1, passando depois pela criação de **Testes**, Secção 6.2. Depois de implementada e testada é realizado o *Code Review*, Secção 6.3, que caso seja feito com sucesso, ou seja, a tarefa não apresente problemas nesta etapa fica pronta para ser alvo de *Quality Assurance (QA)*, Secção 6.4. A partir do momento que a tarefa passa no *QA* e é colocada em produção temos o *Sentry*, Secção 6.5, que permite detetar e guardar os erros que ocorrem na aplicação.

Apesar das diferenças dos processos utilizados, todos têm um objetivo comum, aumentar a qualidade do produto final. Todos estes passos são detalhados mais à frente neste capítulo.

temos de dizer quando estão presentes cada um tipo:

6.1 Indentação de Código

A indentação de código é realizada durante a fase de desenvolvimento do projeto.

Dado o ambiente empresarial em que o projeto se insere e como o cliente quer dar continuidade ao mesmo é essencial o código seguir boas práticas de indentação e estruturação para ser de fácil compreensão por todos os programadores.

Para garantir a qualidade do código foram utilizadas duas ferramentas que detetam más práticas e erros de formatação de código.

No servidor foi utilizada como ferramenta de análise de código Ruby o Rubocop[40], cuja utilização está exemplificada na Figura 6.1.

Na aplicação web foi utilizado o ESLint[39], uma ferramenta de análise a código JavaScript. Na Figura 6.2 é possível visualizar um exemplo do *feedback* gerado.

A utilização destas ferramentas melhora a qualidade do código que facilita a sua leitura e consequentemente acelera o processo de *Code Review*, este processo é explicado na Secção 6.3.

```

2. DCM@din-wdei373: ~/Projects/anotado-brand/server/server (zsh)
→ server git:(master) X rubocop
Warning: unrecognized cop RuboCop::Cop::Rails::SkipsModelValidations found in .rubocop.yml
Inspecting 223 files
.....C.....
.....

Offenses:

app/models/v2/section.rb:3:37: C: Style/PercentLiteralDelimiters: %s-literals should be delimited by ( and ).
  pg_search_scope :search, against: %[text formatted_text info name]
                                     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
app/models/v2/section.rb:5:40: C: Layout/ExtraSpacing: Unnecessary spacing detected.
  default_scope { order(position: asc) }
                        ^
app/models/v2/section.rb:13:23: C: Style/HashSyntax: Use the new Ruby 1.9 hash syntax.
  has_many :sections, :class_name => 'V2::Section',
                               ^^^^^^^^^^^^^^^^^
223 files inspected, 3 offenses detected
→ server git:(master) X █

```

Figura 6.1: Exemplo do resultado do comando *rubocop*

```

2. DCM@din-wdei373: ~/Projects/anotado-brand/server/web (zsh)
→ web git:(master) X yarn lint
yarn run v1.7.0
$ node node_modules/eslint/bin/eslint.js src

/Users/DCM/Projects/anotado-brand/server/web/src/core/setup.js
 58:5  warning  Unexpected console statement  no-console
 64:5  warning  Unexpected console statement  no-console

/Users/DCM/Projects/anotado-brand/server/web/src/core/utils/helpers.js
 46:38  error  Replace `..` with `;`          prettier/prettier
 47:23  error  Replace `""` with `''`        prettier/prettier
 52:27  error  Expected '===' and instead saw '=='  eqeqeq
 53:5   error  Delete `..`                    prettier/prettier
 54:27  error  Expected '===' and instead saw '=='  eqeqeq
 55:1   error  Insert `..`                    prettier/prettier

✖ 8 problems (6 errors, 2 warnings)
 4 errors and 0 warnings potentially fixable with the `--fix` option.

error Command failed with exit code 1.
info Visit https://yarnpkg.com/en/docs/cli/run for documentation about this command.
→ web git:(master) X █

```

Figura 6.2: Exemplo do resultado do comando *yarn lint*

6.2 Testes

Os testes são realizados durante a fase de desenvolvimento e o objetivo destes é detetar problemas nas funcionalidades durante e após o desenvolvimento das mesmas.

Existem vários tipos de testes que podem ser utilizados para testar um projeto desta natureza. Neste projeto foi dada prioridade aos testes unitários, uma vez que o tempo era limitado e também foi considerado que não havia uma funcionalidade tão complexa que tivesse de ser obrigatoriamente protegida com outro tipo de testes.

Dado que o projeto é dividido em servidor e aplicação web foram novamente utilizadas duas ferramentas diferentes para realizar testes unitários. No caso do servidor foi utilizado o *RSpec*[42], na Figura 6.3 é possível observar o exemplo da execução de dois ficheiros de teste que possuem nove testes no total. Na aplicação web foi utilizado o *Jest*[41], na Figura 6.4 é também possível visualizar o exemplo da execução de múltiplos ficheiros de testes. Ambas as ferramentas mostram o tempo que os testes demoraram a executar.

```

2. DCM@din-wdei373: ~/Projects/anutado-brand/server/server (zsh)
→ server git:(master) bundle exec rspec spec/models/v2/annotation_bundle_spec.rb spec/actions/v2/annotation_bundle_mutator_spec.rb -f d

V2::AnnotationBundle
  Validations
    creates a valid AnnotationBundle
  Associations
    should belong to content
    should belong to author
    should have many annotations
    should have many annotation_bundle_authors
    should have many annotation_bundle_ratings

V2::AnnotationBundleMutator
  annotation bundle mutator create
    success
      should respond success with valid bundle
  annotation bundle mutator edit
    success
      should respond success with valid bundle
    invalid bundle
      should respond with error

Finished in 0.77336 seconds (files took 9.62 seconds to load)
9 examples, 0 failures

→ server git:(master) X

```

Figura 6.3: Exemplo do resultado do comando *rspec*

```

2. yarn test (node)
PASS src/views/pages/App/lms/Law/index.test.js
PASS src/views/pages/App/lms/Article/index.test.js
PASS src/views/pages/App/lms/PointCard/index.test.js
PASS src/views/pages/App/lms/Paragraph/index.test.js
PASS src/views/pages/App/lms/Point/index.test.js
PASS src/views/pages/App/lms/ArticleCard/index.test.js
PASS src/views/pages/App/lms/ParagraphCard/index.test.js

Test Suites: 7 passed, 7 total
Tests: 7 passed, 7 total
Snapshots: 0 total
Time: 10.256s
Ran all test suites matching /App\/lms\/i.

Watch Usage: Press w to show more.

```

Figura 6.4: Exemplo do resultado do comando *yarn test*

Sempre que era implementada uma funcionalidade eram criados testes para validar a mesma. A criação de testes foi um processo que por vezes se revelou demorado, tendo nalguns casos tomado um esforço superior ao da implementação da própria funcionalidade. Isto levou à priorização de testes sobre funcionalidades lógicas em detrimento de funcionalidades relacionadas apenas com visualização de dados.

Na Figura 6.5 temos os resultados da cobertura dos testes aos modelos do servidor. Estes testes consistem em testar as validações, associações e os métodos de cada modelo do servidor.

Na Figura 6.6 temos os resultados da cobertura dos testes às *actions* do servidor, estas

actions são os ficheiros que possuem grande parte da lógica do servidor. É de realçar que os ficheiros *purchase_mutator.rb* e *annotation_bundle_sales_mutator.rb* apresentam uma cobertura reduzida devido a estes interagirem com as APIs de pagamento, como estas APIs são externas ao servidor e os resultados obtidos podem variar é difícil de criar testes para estas.

Models V2 (95.24% covered at 4.6 hits/line)

14 files in total. 147 relevant lines. 140 lines covered and 7 lines missed

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/models/v2/purchase_annotation_bundle.rb	62.5 %	27	16	10	6	0.6
app/models/v2/section.rb	96.0 %	68	25	24	1	7.2
app/models/v2.rb	100.0 %	5	3	3	0	5.0
app/models/v2/annotation.rb	100.0 %	30	15	15	0	2.9
app/models/v2/annotation_author.rb	100.0 %	4	3	3	0	1.0
app/models/v2/annotation_bundle.rb	100.0 %	56	23	23	0	2.2
app/models/v2/annotation_bundle_author.rb	100.0 %	15	8	8	0	1.0
app/models/v2/annotation_bundle_author_invite.rb	100.0 %	11	5	5	0	1.0
app/models/v2/annotation_bundle_rating.rb	100.0 %	6	3	3	0	1.0
app/models/v2/app.rb	100.0 %	38	14	14	0	1.0
app/models/v2/author.rb	100.0 %	27	14	14	0	1.1
app/models/v2/content.rb	100.0 %	19	9	9	0	30.1
app/models/v2/favourite.rb	100.0 %	9	5	5	0	10.2
app/models/v2/match_section_version.rb	100.0 %	9	4	4	0	1.8

Figura 6.5: Coverage dos testes aos modelos

Actions V2 (75.59% covered at 1.54 hits/line)

9 files in total. 426 relevant lines. 322 lines covered and 104 lines missed

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/actions/mutations/v2/purchase_mutator.rb	27.12 %	104	59	16	43	0.4
app/actions/mutations/v2/annotation_bundle_sales_mutator.rb	38.1 %	75	42	16	26	0.5
app/actions/mutations/v2/trial_mutator.rb	81.82 %	19	11	9	2	0.9
app/actions/mutations/v2/favourite_mutator.rb	82.76 %	51	29	24	5	2.3
app/actions/mutations/v2/annotation_bundle_invites_mutator.rb	87.5 %	140	80	70	10	1.4
app/actions/mutations/v2/annotation_bundle_annotations_mutator.rb	89.47 %	99	57	51	6	1.5
app/actions/mutations/v2/private_annotation_mutator.rb	90.48 %	112	63	57	6	3.5
app/actions/mutations/v2/redeem_code_mutator.rb	91.67 %	39	24	22	2	2.0
app/actions/mutations/v2/annotation_bundle_mutator.rb	93.44 %	111	61	57	4	1.1

Figura 6.6: Coverage dos testes às actions

Na Figura 6.7 temos os resultados da cobertura dos testes da aplicação web. Na aplicação web as páginas que utilizam APIs externas possuem uma cobertura inferior, é o caso do *SignUp* e *Login*, onde é possível utilizar o Facebook ou Google para autenticação, e do *PremiumMarket* e *CollaborativeMarket*, devido à integração dos serviços de pagamento.

All files
75.08% Statements 449/598 **65.98%** Branches 225/341 **72.59%** Functions 188/259 **75.65%** Lines 438/579

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File	Statements	Branches	Functions	Lines
App/cpc/Article	100%	3/3	50%	1/2
App/cpc/ArticleCard	83.33%	5/6	75%	3/4
App/cpc/Book	51.16%	22/43	36.67%	11/30
App/cpc/BookCard	100%	4/4	100%	2/2
App/cpc/Chapter	92.31%	12/13	75%	3/4
App/cpc/Code	100%	6/6	100%	4/4
App/cpc/Header	80%	4/5	100%	3/3
App/cpc/Part	100%	14/14	100%	4/4
App/cpc/Title	100%	17/17	90%	9/10
App/cpc/TitleCard	80%	4/5	50%	1/2
App/fam/Article	80%	8/10	40%	2/5
App/fam/ArticleCard	75%	6/8	62.5%	5/8
App/fam/Law	90%	9/10	77.78%	7/9
App/fam/LawCard	80%	4/5	100%	0/0
App/fam/Paragraph	100%	3/3	50%	2/4
App/fam/ParagraphCard	83.33%	5/6	50%	1/2
App/lms/Article	76.92%	10/13	62.5%	5/8
App/lms/ArticleCard	63.64%	7/11	75%	6/8
App/lms/Law	90%	9/10	77.78%	7/9
App/lms/Paragraph	90%	9/10	57.14%	4/7
App/lms/ParagraphCard	75%	6/8	75%	3/4
App/lms/Point	100%	3/3	50%	3/6
App/lms/PointCard	83.33%	5/6	100%	0/0
Apps	100%	17/17	100%	7/7
Bookmarks	90.91%	10/11	66.67%	4/6
CollaborativeMarket/Bundle	0%	0/9	0%	0/6
Login	47.83%	11/23	25%	1/4
MyBundle/Annotation	65.38%	17/26	78.95%	15/19
MyBundle/AssociateAnnotations	78.57%	22/28	66.67%	10/15
MyBundle/Bundle	78.57%	11/14	71.43%	10/14
MyBundle/NewAnnotation	69.23%	18/26	72.73%	8/11
MyBundle/Sell	81.82%	9/11	100%	2/2
MyBundles	88.89%	8/9	85.71%	6/7
NewBundle	78.57%	22/28	61.54%	8/13
NewBundle/Annotation	100%	7/7	100%	0/0
PremiumMarket/App	0%	0/11	0%	0/6
PremiumMarket/TrialActivation	73.33%	11/15	75%	9/12
Profile	70%	14/20	36.36%	4/11
RecoverPassword	100%	13/13	100%	4/4
RedeemCode	91.67%	11/12	100%	2/2
ResendConfirmation	100%	14/14	100%	4/4
ResetPassword	100%	14/14	83.33%	5/6
Search	63.27%	31/49	70.59%	36/51
SignUp	63.64%	14/22	66.67%	4/6

Figura 6.7: Coverage dos testes às páginas da aplicação

6.3 Code Review

Depois do desenvolvimento estar concluído, passamos para o *Code Review (CR)*, ou em português, Revisão de Código. De relembrar que caso seja necessário efetuar correções a tarefa pode voltar para a fase de desenvolvimento como explicado no Capítulo 5 - Seção 5.2.

O *CR* é um processo de leitura e análise de código com o intuito de encontrar problemas que não foram detetados pelo programador ou pelas ferramentas de análise automática. Este

processo era realizado depois da abertura de um *Merge Request* no Git e era efetuado por pelo menos dois elementos da empresa, sendo que o *Merge Request* só era aceite e adicionado à *branch* principal depois de ser aprovado por duas pessoas diferentes, reduzindo assim a probabilidade de existir erros no código.

Na Figura 6.8 podemos visualizar um exemplo de uma discussão resultante do processo de *CR*.

Discussion 2 Commits 13 Pipelines 3 Changes 28 Show all activity 1/1 discussion resolved

André Perdigão @andrepcg started a discussion on the diff 2 months ago Resolved by Dinis Marques 2 months ago

server/app/actions/mutations/v2/annotation_bundle_invites_mutator.rb

```

36 +                                     role: role,
37 +                                     status: status)
38 +
39 +   if invite.persisted?
40 +     data = true
41 +     success = true
42 +   else
43 +     data = invite.errors.full_messages.join(' ')
44 +     success = false
45 +   end
46 +
47 +   return data, success
48 + end
49 +
50 + def self.invite_email(name, email, bundle, role)
51 +   invite = V2::AnnotationBundleAuthorInvite.create(annotation_bundle: bundle,

```

André Perdigão @andrepcg · 2 months ago Owner

Existe alguma validação de email? Talvez seja nice ter e cancelar o flow caso seja invalido. Pode ate ser no model (nao confirmei se o model tem essa validacao)

Dinis Marques @DCoelhoM · 2 months ago Developer

Algo assim: `validates :email, format: Devise.email_regexp ?`

Reply... Unresolve discussion

Dinis Marques @DCoelhoM added 1 commit 2 months ago

- 85c4a9d8 - antd-168: add email validation

Compare with previous version

Dinis Marques @DCoelhoM resolved all discussions 2 months ago

Figura 6.8: Exemplo de uma discussão resultante de *CR*

6.4 Quality Assurance

O *QA*, ou em português, Garantia de Qualidade, é um processo focado em testar e validar as funcionalidades, desde as tratamento de erros aos comportamentos gerados por certas interações.

Este processo é realizado depois do *CR* e foi realizado por elementos da RLS designados para o efeito. Quando era detetado algum problema com a funcionalidade a pessoa responsável adicionava um comentário no cartão da tarefa a explicar o problema encontrado, na Figura 6.9 podemos ver um exemplo.

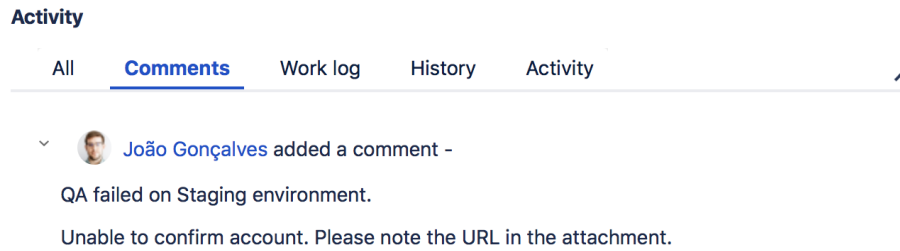


Figura 6.9: Exemplo de uma mensagem caso a tarefa falhe no *QA*

6.5 Sentry

Para a deteção de erros que ocorrem durante o *runtime*, em português tempo de execução, no servidor ou na aplicação web, foi utilizado o *Sentry*[44]. Esta ferramenta permite detetar a origem e causa desse mesmo erro, facilitando a sua resolução.

Esta ferramenta permite registar erros que ocorrem durante a utilização da aplicação por parte do utilizador, não sendo necessário este entrar em contacto com a equipa de suporte para reportar um problema. Estes erros não são detetáveis pelo QA.

Na Figura 6.10 é possível observar uma lista exemplo de erros detetados pelo *Sentry* e na Figura 6.11 temos algumas das informações que o *Sentry* guarda sobre um erro.

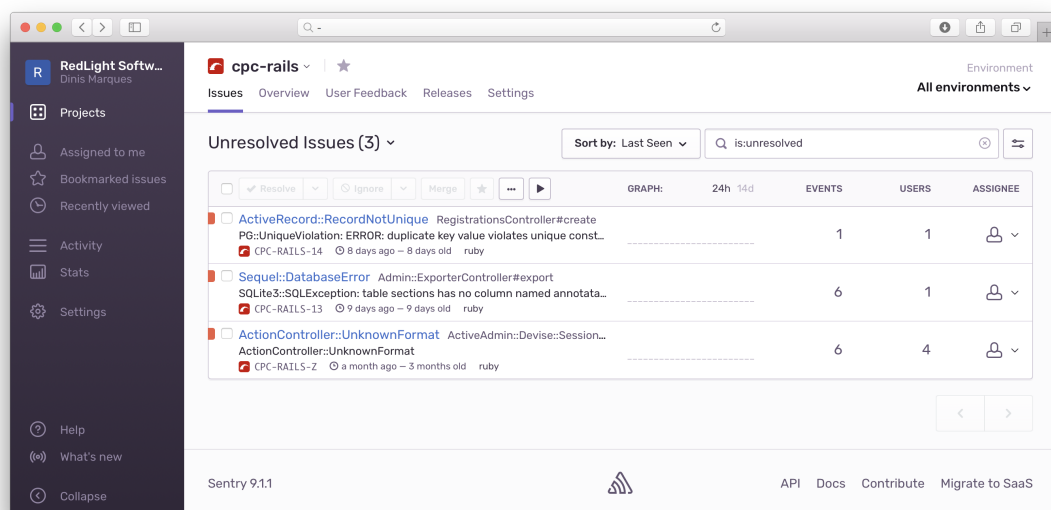


Figura 6.10: Exemplo da página de erros do *Sentry*

ActionController::UnknownFormat ActiveAdmin::Devise::SessionsController#new ISSUE #

● ActionController::UnknownFormat CPC-RAILS-Z

ruby

Resolve Ignore Star Share

Details Comments 0 User Feedback 0 Tags Events Merged

Event c329db183a194e8fa62e89d8e8108135 K Older Newer >

Jun 14, 2019 2:57:48 AM UTC | [JSON \(35.7 KB\)](#)

⚠ There was 1 error encountered while processing this event Show

46.105.99.163 **Firefox**
Version: 64.0 **Windows 10**
Version: Unknown

TAGS

browser Firefox 64.0 browser.name Firefox environment production level error logger ruby os.name Windows 10

release 37104b957109be65f6e8535549fed3f8e681ffe5 server_name anotado

transaction ActiveAdmin::Devise::SessionsController#new url https://anotado.weareredlight.com/admin/login.php

user ip:46.105.99.163

MESSAGE

ActionController::UnknownFormat: ActionController::UnknownFormat

EXCEPTION (most recent call first) Full Raw

ActionController::UnknownFormat

ActionController::UnknownFormat

Figura 6.11: Exemplo da visualização de um erro no *Sentry*

Capítulo 7

Resultados e métricas do estágio

Neste capítulo vamos detalhar o estado final do projeto, Secção 7.1, e analisar as métricas obtidas para confirmar que os objetivos foram cumpridos, Secção 7.2.

7.1 Estado final da aplicação

Nesta secção vamos analisar o estado final da aplicação desenvolvida durante o estágio e detalhar algumas das funcionalidades implementadas.

Todas as páginas de autenticação são visualmente semelhantes, apenas mudam os campos a preencher, por isso vamos apenas analisar a página de login, Figura 7.1, que permite ao utilizador entrar na aplicação e aceder a todas as funcionalidades da aplicação. O utilizador pode fazer login através do email e senha ou através dos serviços externos Facebook e Google.

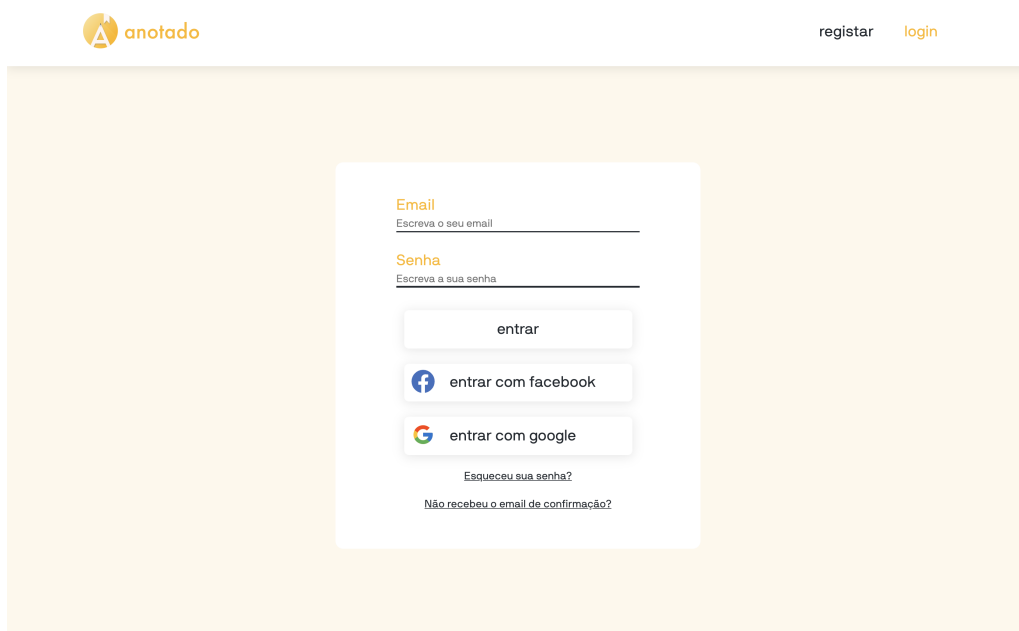


Figura 7.1: Página de login

Quando o utilizador efetua o login é redirecionado para a página dos conteúdos disponíveis, Figura 7.2. Tem também disponível no lado esquerdo um menu de navegação onde pode aceder às várias funcionalidades da aplicação inclusive utilizar diretamente a funcionalidade de pesquisa.

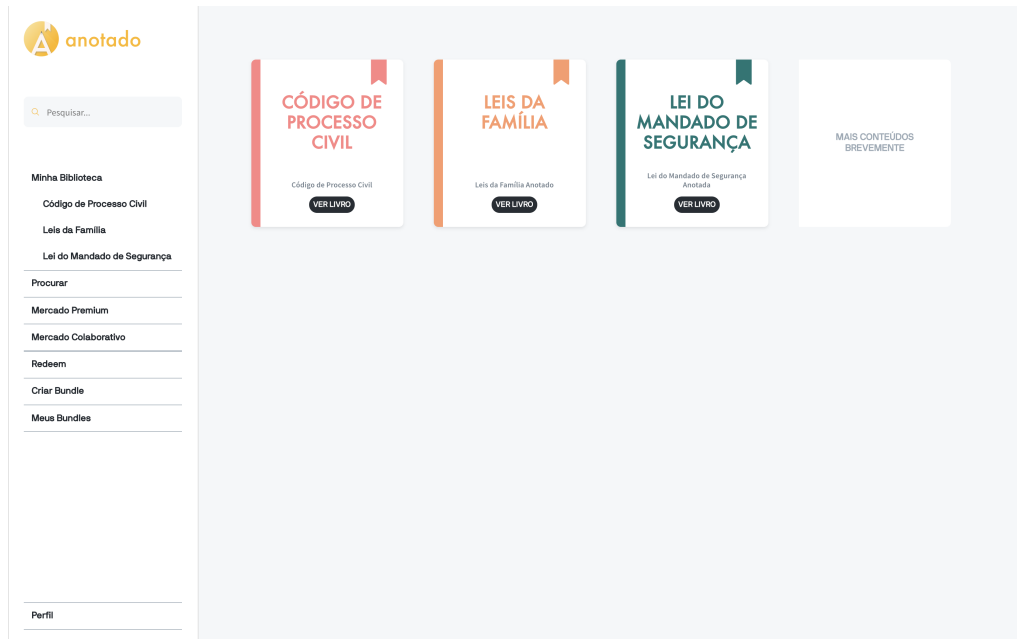


Figura 7.2: Página de listagem de conteúdos

Ao entrar num dos conteúdos, Figura 7.3, são apresentadas as várias secções que este possui. Esta listagem pode ser diferente de conteúdo para conteúdo. O utilizador nesta página tem algumas ações relativas às secções disponíveis, como ir diretamente para a funcionalidade de criar uma nova anotação ou adicionar/remover a secção dos favoritos.

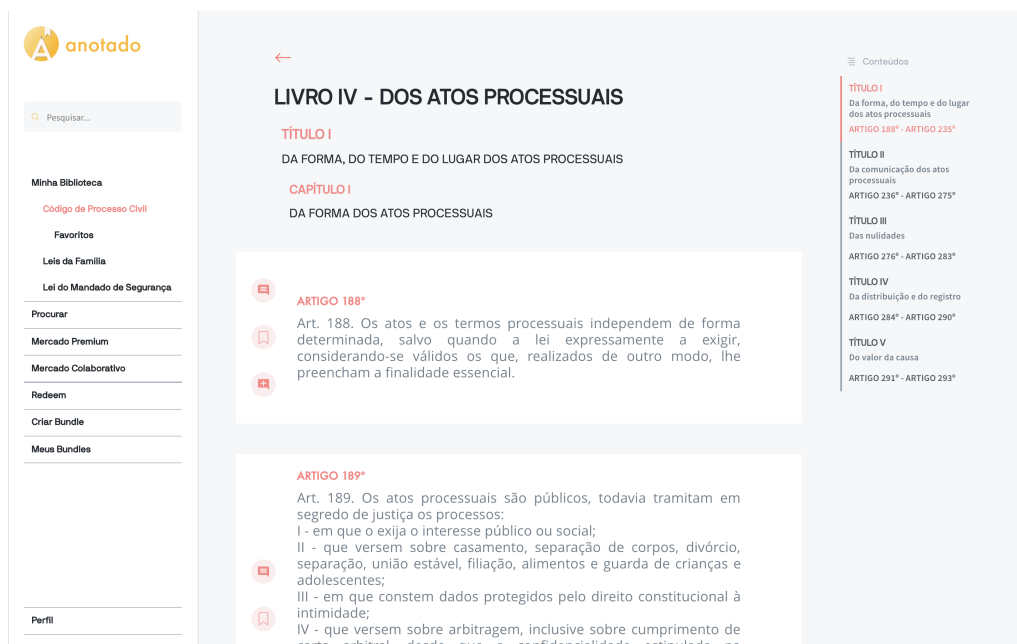


Figura 7.3: Página de listagem de secções

Já dentro da secção o utilizador pode listar todas as anotações associadas, Figura 7.4 ou criar uma nova, Figura 7.5.

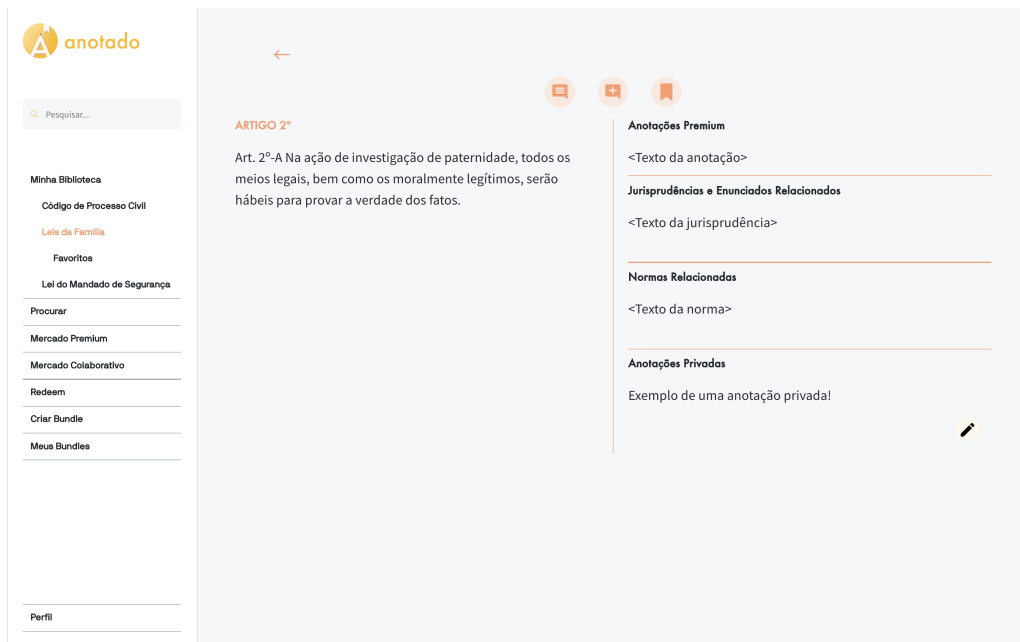


Figura 7.4: Página de uma secção - listagem de anotações

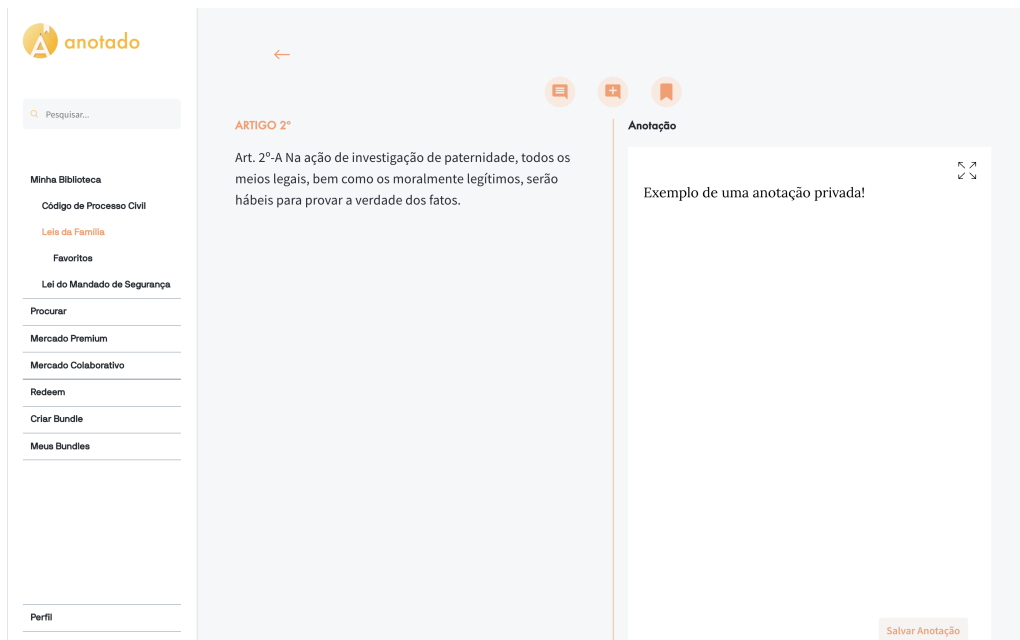


Figura 7.5: Página de uma secção - criar nova anotação privada

O utilizador dentro de cada conteúdo pode aceder aos seus favoritos, Figura 7.6, para um rápido acesso às secções que colocou favorito.



Figura 7.6: Página dos favoritos

O utilizador pode pesquisar, Figura 7.7, no conteúdo que pretender e em qualquer tipo de secção ou anotação dentro desse conteúdo.



Figura 7.7: Página de pesquisa

Através do menu lateral o utilizador pode aceder à loja premium e visualizar todos os conteúdos premium que são disponibilizados pela aplicação, Figura 7.8.

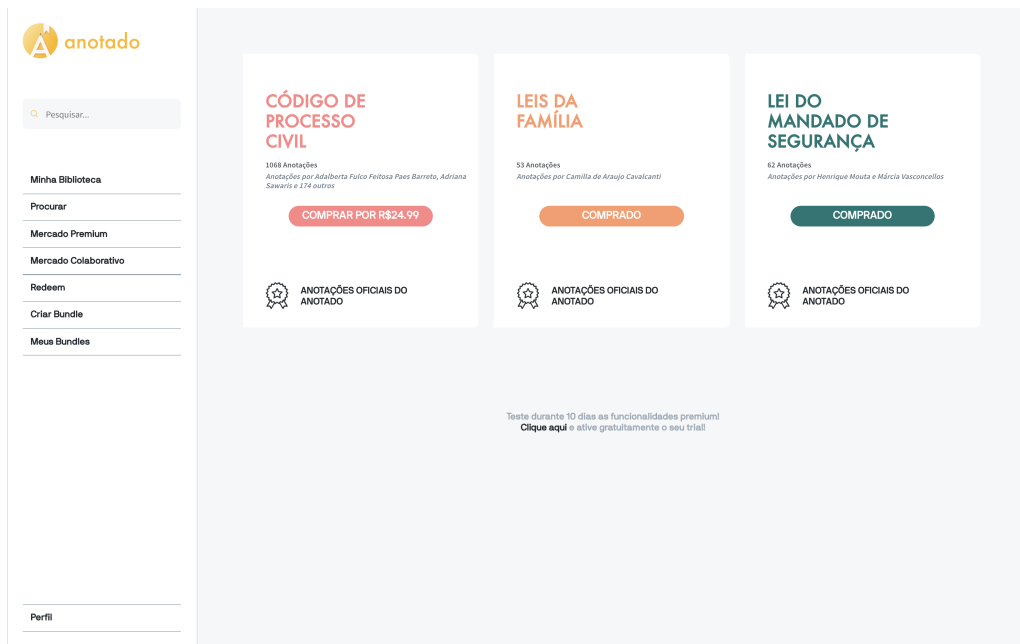


Figura 7.8: Página da loja premium

Ao clicar num dos produtos o utilizador pode visualizar algumas das anotações e caso queira comprar o premium pode efetuar o pagamento usando o *PayPal* ou o cartão de crédito através do *Stripe*, Figura 7.9.

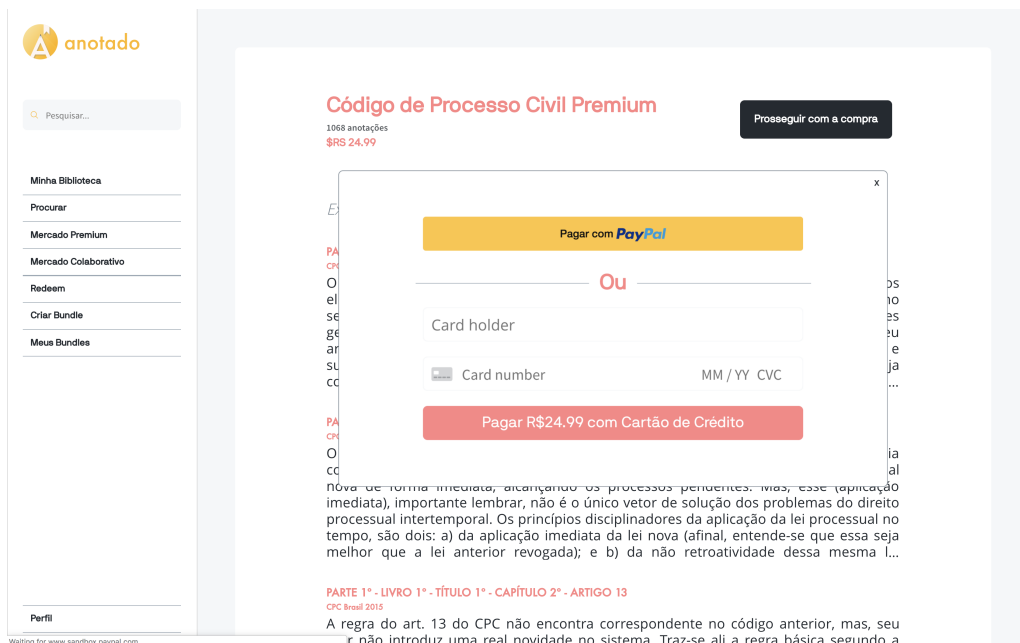


Figura 7.9: Página de uma compra

O utilizador pode ainda criar um bundle colaborativo, Figura 7.10, onde pode associar as suas anotações privadas.

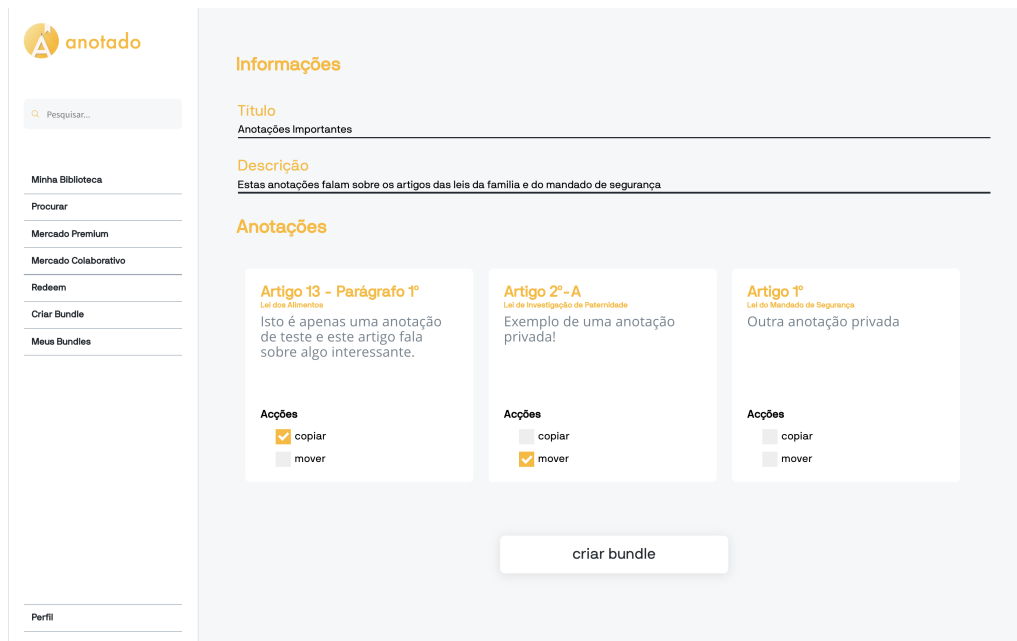


Figura 7.10: Página de criação de bundle colaborativo

Na página do bundle, Figura 7.11, o utilizador consegue convidar autores através do email para criarem e editarem anotações, pode também adicionar novas anotações ou bundle. Nesta página o criador do bundle pode ainda o colocar para venda.

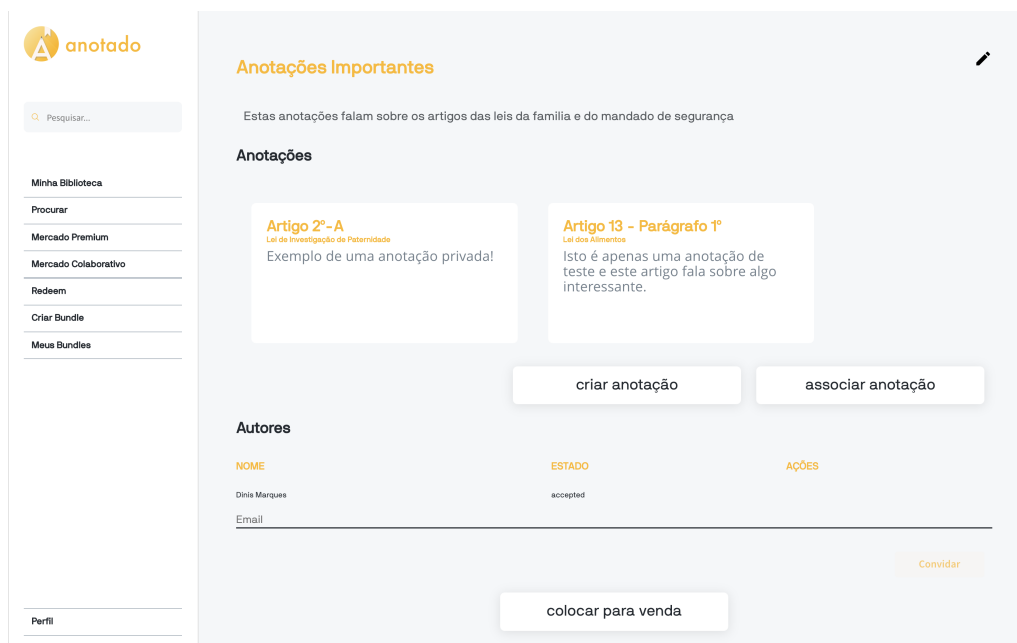


Figura 7.11: Página de um bundle colaborativo

Para além da aplicação web existe também um painel de administração, Figura 7.12, que foi criado recorrendo ao *ActiveAdmin*[29]. Este painel permite ao administrador gerir os conteúdos, secções, compras e códigos de oferta disponíveis na aplicação.

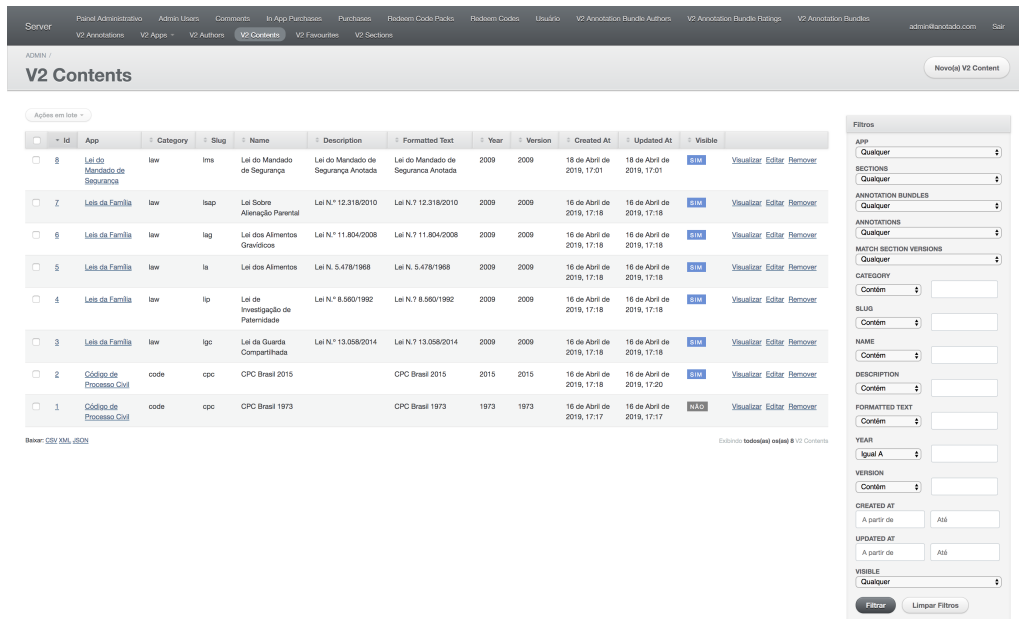


Figura 7.12: Página de um bundle colaborativo

7.2 Objetivos e métricas

Nesta secção vamos analisar as métricas relacionadas com o projeto e verificar se os objetivos planeados foram cumpridos.

7.2.1 Requisitos Funcionais

Primeiramente vamos analisar as métricas relativas aos requisitos funcionais, Capítulo 4 - Secção 4.2.1, com o intuito de perceber quais foram cumpridos e implementados.

Através da Tabela 7.1 podemos observar a percentagem dos requisitos implementados tendo em conta a sua prioridade.

Prioridade	Percentagem Implementada
Must Have	100% (28/28)
Should Have	75% (6/8)
Could Have	14% (1/7)
Won't Have	0% (0/1)

Tabela 7.1: Métricas dos requisitos funcionais

Dos requisitos com prioridade **Should Have** apenas os **Termos e Condições** e a **Política de Privacidade** não foram implementados. Já nos requisitos com prioridade **Could Have** apenas foi implementado o requisito **Ativar tempo de experimentação**.

7.2.2 Requisitos Não Funcionais

Para confirmar que os requisitos não funcionais, Capítulo 4 - Secção 4.2.2, fossem cumpridos foram realizados vários tipos de testes.

Como o requisito **Usabilidade** não era totalmente da responsabilidade do aluno este foi cumprido seguindo os padrões já criados pela equipa de design, sendo que estão também previstos testes de usabilidade num futuro próximo.

Para cumprir o requisito de **Segurança e Privacidade** todos os pedidos realizados ao servidor necessitam de ser autenticados através de um *token* JWT[45]. Este *token* é digitalmente assinado pelo servidor permitindo que toda a informação que este transporta possa ser verificada e confiada. Cada *token* de autenticação é único e está apenas associado a um utilizador.

Para garantir a **Compatibilidade** entre browsers foi utilizado o *Can I Use*[46] para verificar se certas funcionalidades da linguagem de *JavaScript* funcionavam nos browsers pretendidos. Na Figura 7.13 é apresentado um exemplo do *Can I Use* onde podemos visualizar as versões dos browsers que suportam a funcionalidade *scrollIntoView* do *JavaScript*.

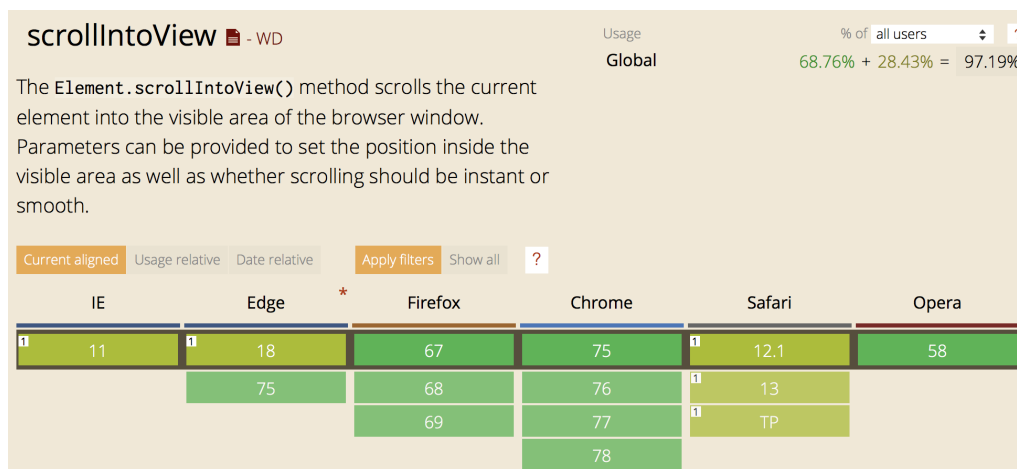


Figura 7.13: Exemplo do *Can I Use*

Para testar o requisito não funcional relativo à **Performance** foi utilizado o *ab*[47] que é uma ferramenta para fazer *benchmarking* a servidores HTTP, na Figura 7.14 é mostrado o comando e as configurações possíveis para utilizar essa ferramenta. Esta ferramenta permite definir o número de clientes que queremos que se conectem ao servidor simultaneamente e o número de pedidos que cada um efetua. Na Figura 7.15 é possível visualizar o resultado de um teste utilizando a ferramenta.

```

$ ab -H "authorization: Bearer <AuthToken>" #Token de autenticao
-T application/json
-p <PayloadFile> #Ficheiro com o corpo do pedido
-s 120
-k
-c <NumberOfClients> #Numero de clientes simultaneos
-n <TotalRequests> #Total de pedidos executados
https://<API_URL>/graphql/v2 #URL do pedido

```

Figura 7.14: Comando *ab* e as suas configurações

```

This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking api-staging. anotado. rls- intra. com (be patient)
Completed 100 requests
Completed 200 requests
Finished 250 requests

Server Software:      nginx/1.10.3
Server Hostname:     api-staging. anotado. rls- intra. com
Server Port:         443
SSL/TLS Protocol:    TLSv1.2, ECDHE-RSA-CHACHA20-POLY1305, 2048, 256
Server Temp Key:     X25519 253 bits
TLS Server Name:     api-staging. anotado. rls- intra. com

Document Path:       /graphql/v2
Document Length:     0 bytes

Concurrency Level:   50
Time taken for tests: 2.130 seconds
Complete requests:  250
Failed requests:    0
Non-2xx responses:  250
Keep-Alive requests: 0
Total transferred:  112000 bytes
Total body sent:    142500
HTML transferred:   0 bytes
Requests per second: 117.35 [# /sec] (mean)
Time per request:   426.090 [ms] (mean)
Time per request:   8.522 [ms] (mean, across all concurrent requests)
Transfer rate:      51.34 [Kbytes/sec] received
                   65.32 kb/s sent
                   116.66 kb/s total

Connection Times (ms)
      min   mean[+/-sd] median   max
Connect:  164   213   39.8   211   638
Processing:  74   141   34.3   138   229
Waiting:    65   139   33.1   137   224
Total:     251   354   54.7   347   761

Percentage of the requests served within a certain time (ms)
 50%    347
 66%    367
 75%    377
 80%    384
 90%    421
 95%    451
 98%    459
 99%    459
100%   761 (longest request)

```

Figura 7.15: Exemplo do resultado do comando *ab*

Esta ferramenta foi executada para cada funcionalidade com um número de clientes igual a 500 e um total de pedidos igual a 2500, isto equivale a cada cliente fazer 5 pedidos.

Estes testes foram feitos através da máquina pessoal do aluno ao servidor de *staging* que se encontra hospedado numa máquina do serviço Linode[48] que apenas possui 2GB RAM e 1 CPU Core, por isso devemos considerar que os resultados obtidos podem melhorar com o aumento das características da máquina.

Teste	Percentagem de pedidos respondidos num determinado tempo (ms)									Tempo (ms)			
	50%	66%	75%	80%	90%	95%	98%	99%	100%	Min	Média	Desvio Padrão	Máx
(1)Listar todas as Aplicações	160	361	1582	2152	3257	3667	4897	5222	5665	20	924	1352.6	5665
(2)Listar Conteúdos de uma Aplicação(CPC)	129	203	385	728	2427	4411	5868	6446	6905	49	726	1392.6	6905
(3)Listar os Favoritos de uma Aplicação	116	141	158	204	2401	4324	6355	7369	8075	44	698	1504.0	8075
(4)Listar Conteúdos de uma Aplicação(LMS)	139	168	307	1650	6662	54593	61420	61568	61764	32	4955	14734.8	61764
(5)Listar uma Secção	119	160	209	1678	13311	60544	61238	61535	66009	49	5463	15846.2	66009
(6)Criar nova Anotação(1024 caracteres, 1KB)	227	285	997	2486	4354	6477	7347	7659	8722	44	1202	1989.9	8722
(7)Criar nova Anotação(16384 caracteres, 16KB)	715	1913	3194	4193	8045	10314	10986	11465	15920	46	2372	3194.6	15920
(8)Listar os Bundles Colaborativos	112	120	139	187	2440	4077	5026	5485	6189	47	652	1298.9	6489
(9)Listar as compras Premium	165	262	1471	2080	3355	3745	5065	5318	5565	46	929	1395.0	5565
(10)Listar uma compra Premium	140	197	379	1288	3503	12129	19228	21941	24033	50	1580	4179.6	24033
(11)Pesquisa	278	330	353	2701	3382	15678	23711	26363	28485	91	2072	5202.9	28485

Tabela 7.2: Resultados dos testes de performance ao servidor

Através da Tabela 7.2 verificamos que alguns dos testes demoraram em média mais de 1 segundo, mas tendo em conta as características da máquina e analisando com mais detalhe toda a tabela podemos concluir que os resultados estão perto do pretendido.

Analisando a tabela teste a teste conseguimos perceber que os testes relativos à criação de uma nova anotação são o que demoram mais tempo e que são dependentes do tamanho da anotação. O segundo teste que demorou mais tempo foi o da pesquisa, esta funcionalidade irá demorar mais quanto maior for o conteúdo no qual se efetua a pesquisa. De realçar que nos testes **3** e **4** ocorreu *timeout* em cerca de 5% dos pedidos, ou seja, o servidor demorou mais de 5 segundos a responder ao pedido.

Para testar a performance da aplicação web foi usada a funcionalidade das *DevTools* do *Chrome* que analisa a *Performance*[49] de uma página web, na Figura 7.16 é possível ver o exemplo de um resultado da ferramenta. Estes testes foram realizados assumindo que o servidor está a funcionar normalmente e o utilizador está com uma largura de banda de 20Mbps.

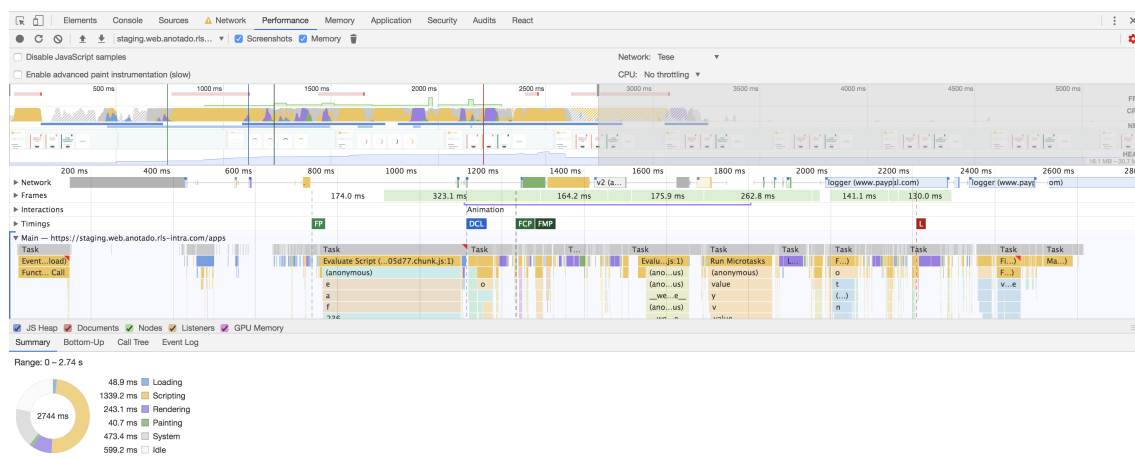


Figura 7.16: Exemplo do Chrome Dev Tools - Performance

Página	Time (ms)				
	Loading	Scripting	Rendering	Painting	Total
Listar todas as Aplicações	70.8	1158.5	61.2	23.7	1314.2
Listar Conteúdos de uma Aplicação(CPC)	107.6	2052.3	97.4	14.9	2272.2
Listar Conteúdos de uma Aplicação(LMS)	28.3	2125.5	126.5	0.4	2280.7
Listar Favoritos	44.8	1368.9	106.7	16.4	1536.8
Listar compras Premium	50.2	844.7	51.3	163.3	1109.5
Listar uma compra Premium	91.5	2635.6	404.1	62.8	3194.0
Pesquisa	55.2	1587.6	74.1	36.6	1753.5

Tabela 7.3: Resultados dos testes de performance à aplicação web

Como podemos ver pela Tabela 7.3 todas as páginas testadas carregam na totalidade em menos de 5 segundos e se tivermos em conta que depois do tempo de *loading* já existe algum conteúdo ou *feedback* a ser mostrado na página podemos afirmar que antes de o utilizador passar 2 segundos na página já consegue visualizar algum tipo de informação.

Capítulo 8

Conclusão

A primeira fase do estágio, que correspondeu ao primeiro semestre, foi principalmente dedicada ao planeamento, escolha do modelo de desenvolvimento, levantamento dos requisitos, desenho das arquiteturas, análise e escolha de tecnologias, desafios tecnológicos, riscos inerentes ao projeto e aprendizagem das tecnologias usadas.

O estado da arte permitiu principalmente perceber quais as possíveis alternativas de apresentar os conteúdos e as suas respetivas anotações. Também ajudou a entender posição do produto que se está a desenvolver em relação ao mercado e a possíveis concorrentes.

O modelo de desenvolvimento escolhido permitiu que os requisitos pudessem sofrer alterações e adaptar-se às necessidades do cliente. Também permitiu que ao fim de cada *sprint* existisse um produto funcional pronto a ser apresentado ao cliente o que permitiu obter *feedback* e repensar certas funcionalidades.

A análise de riscos ao ser feita antecipadamente e a aplicação das respetivas estratégias de mitigação evitou o aparecimento de contratempos numa fase mais avançada do projeto.

A segunda e última fase, realizada durante o segundo semestre, foi focada no desenvolvimento, verificação e validação do projeto. Esta fase foi a mais longa de todas.

O desenvolvimento foi a fase que maior esforço necessitou e foi nesta fase que foram encontrados os maiores desafios relacionados com as tecnologias que até à realização do estágio a experiência era nula ou quase nula, foi também nesta fase que foi adquirido o maior conhecimento tanto sobre as tecnologias como sobre a utilização de processos de desenvolvimento e as suas vantagens.

A verificação e validação permitiu garantir a qualidade do trabalho desenvolvido. Esta fase demonstrou a importância de verificar e validar um projeto, seja através de simples testes unitários ou métodos mais completos de verificação e validação.

Para concluir foi um projeto interessante em todas as suas etapas, sendo que todas proporcionaram a aprendizagem de novos conhecimentos em diversas áreas, ferramentas ou tecnologias. Proporcionou também o primeiro contacto com o mundo empresarial e de como é trabalhar num ambiente como o da RLS em que todos os elementos, desde o primeiro dia do estágio, se mostraram disponíveis a ajudar no que fosse preciso

8.1 Futuro do Projeto

Apesar do estágio ter terminado e o projeto já se encontrar funcional existem detalhes que podem ser limados e novas funcionalidades que podem ser adicionadas.

Por isso foi feita uma análise em conjunto com o cliente sobre o que poderá ser o futuro do projeto:

1. **Curto prazo:** A nível de implementação o mais importante a implementar primeiro seria um serviço semiautomático que pague ao criador do bundle um percentagem do valor das vendas, em que apenas seja necessário uma pessoa responsável confirmar o pagamento. A nível de User Interface (UI)/User Experience (UX) algumas das interfaces precisam de ser melhoradas e utilizar técnicas de *call to action* que incentivem a compra de produtos dentro da plataforma.
2. **Médio prazo:** A médio prazo a plataforma deve permitir que os conteúdos sejam inseridos para além dos administradores pelos utilizadores. A parte monetária também beneficiaria com a implementação de um sistema que faça a divisão automáticas do dinheiro das vendas por todos os autores, retirando a necessidade de uma pessoa confirmar os pagamentos. A aplicação móvel pode ser utilizada em sem uma conexão à internet, esta funcionalidade também seria útil na aplicação web, podendo ser implementada a médio ou longo prazo dependendo do esforço necessário.
3. **Longo prazo:** A longo prazo o projeto irá beneficiar de um sistema de aprovação de anotações automático que detete tentativas de spam ou plágio. O sistema de edição também pode melhorar utilizando um sistema tipo *Git* para gestão de versões das anotações. Com o crescimento da plataforma a funcionalidade de pesquisa, que segundo os testes de *load balance* é das mais pesadas, deve ser repensada e talvez separada da aplicação principal, isto pode ser resolvido usando o Elasticsearch[50] que é um motor de busca otimizado.

Referências

- [1] Redlight Software. Redlight Software. <https://weareredlight.com>. [Online; accessed 08-January-2019].
- [2] Anotado. Anotado. <https://direitoanotado.com.br>. [Online; accessed 03-January-2019].
- [3] Annotation Studio. Annotation Studio. <https://www.annotationstudio.org>. [Online; accessed 03-January-2019].
- [4] DocumentCloud. DocumentCloud. <https://www.documentcloud.org/>. [Online; accessed 03-January-2019].
- [5] Genius. Song Lyrics & Knowledge. <https://genius.com>. [Online; accessed 03-January-2019].
- [6] WritingPod. Online Writing Community. <http://www.writingpod.com/>. [Online; accessed 08-January-2019].
- [7] Infinite Ulysses. Infinite Ulysses. <http://infiniteulysses.com>. [Online; accessed 08-January-2019].
- [8] Adobe. Acrobat Reader. <https://acrobat.adobe.com/us/en/acrobat/pdf-reader.html>. [Online; accessed 10-April-2019].
- [9] XODO. GET IT DONE WITH PDF. <https://www.xodo.com>. [Online; accessed 10-April-2019].
- [10] Google. Docs. <https://www.google.com/docs/about/>. [Online; accessed 10-April-2019].
- [11] Mendeley. PDF Reading & Annotation. <https://www.mendeley.com/guides/android/03-pdf-reading-annotation>. [Online; accessed 10-April-2019].
- [12] Kami. Your Digital Classroom Hero. <https://www.kamiapp.com>. [Online; accessed 10-April-2019].
- [13] Hypothes.is. Annotate the web, with anyone, anywhere. <https://web.hypothes.is>. [Online; accessed 10-April-2019].
- [14] Massachusetts Institute of Technology. Massachusetts Institute of Technology. <https://web.mit.edu/>. [Online; accessed 08-January-2019].
- [15] PayPal. PayPal Developer Documentation. <https://developer.paypal.com/>. [Online; accessed 26-December-2018].

- [16] Stripe. Online payment processing for internet businesses. <https://stripe.com/>. [Online; accessed 26-December-2018].
- [17] Atlassian. Scrum - what it is, how it works, and why it's awesome. <https://www.atlassian.com/agile/scrum>. [Online; accessed 02-January-2019].
- [18] Atlassian. Kanban - A brief introduction. <https://www.atlassian.com/agile/kanban>. [Online; accessed 02-January-2019].
- [19] Speedtest. Monthly comparisons of internet speeds from around the world. <https://www.speedtest.net/global-index>. [Online; accessed 03-January-2019].
- [20] Akamai. 2 Seconds is Threshold for eCommerce Performance. <https://www.akamai.com/us/en/about/news/press/2009-press/akamai-reveals-2-seconds-as-the-new-threshold-of-acceptability-for-ecommerce-web-page-response-times.jsp>. [Online; accessed 03-January-2019].
- [21] yourbrowser.is. Browser market share. <https://yourbrowser.is/browser-market-share/>. [Online; accessed 03-January-2019].
- [22] Coding the Architecture. C4 model poster. http://www.codingthearchitecture.com/2014/08/24/c4_model_poster.html. [Online; accessed 26-December-2018].
- [23] Facebook. Facebook for Developers. <https://developers.facebook.com/>. [Online; accessed 26-December-2018].
- [24] Google. Google Developers. <https://developers.google.com/>. [Online; accessed 26-December-2018].
- [25] Ruby Programming Language. A programmers' best friend. <https://www.ruby-lang.org/en/>. [Online; accessed 17-December-2018].
- [26] Ruby on Rails. A web-application framework that includes everything needed to create database-backed web applications according to the Model-View-Controller (MVC) pattern. <https://rubyonrails.org/>. [Online; accessed 17-December-2018].
- [27] PostgreSQL. The world's most advanced open source database. <https://www.postgresql.org/>. [Online; accessed 17-December-2018].
- [28] Devise. Flexible authentication solution for Rails with Warden. <https://github.com/plataformatec/devise>. [Online; accessed 17-December-2018].
- [29] Active Admin. The administration framework for Ruby on Rails. <https://activeadmin.info/>. [Online; accessed 17-December-2018].
- [30] GraphQL. A query language for your API. <https://graphql.org/>. [Online; accessed 17-December-2018].
- [31] GraphQL Ruby. Add GraphQL to your Ruby app. <http://graphql-ruby.org/>. [Online; accessed 17-December-2018].
- [32] JavaScript. JavaScript (JS) is a lightweight interpreted or JIT-compiled programming language with first-class functions. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Online; accessed 17-December-2018].
- [33] React. A JavaScript library for building user interfaces. <https://reactjs.org/>. [Online; accessed 17-December-2018].

- [34] Apollo GraphQL. Learn about the Apollo platform: Client, Engine, GraphQL servers, GraphQL support, and more. <https://www.apollographql.com/>. [Online; accessed 17-December-2018].
- [35] React Router. Declarative Routing for React.js. <https://reacttraining.com/react-router/>. [Online; accessed 17-December-2018].
- [36] Git. Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. <https://git-scm.com/>. [Online; accessed 17-December-2018].
- [37] Jira. Issue & Project Tracking Software. <https://www.atlassian.com/software/jira>. [Online; accessed 17-December-2018].
- [38] Slack. Where Work Happens. <https://slack.com/>. [Online; accessed 17-December-2018].
- [39] ESLint. Pluggable JavaScript linter. <https://eslint.org/>. [Online; accessed 17-December-2018].
- [40] Rubocop. The Ruby Linter that Serves and Protects. <https://docs.rubocop.org/en/latest/>. [Online; accessed 17-December-2018].
- [41] Jest. Delightful JavaScript Testing. <https://jestjs.io/>. [Online; accessed 17-December-2018].
- [42] RSpec. Testing framework for Rails. <https://github.com/rspec/rspec-rails>. [Online; accessed 17-December-2018].
- [43] Apollo GraphQL. GraphQL vs. REST. <https://blog.apollographql.com/graphql-vs-rest-5d425123e34b>. [Online; accessed 28-December-2018].
- [44] Sentry. Error Tracking Software - JavaScript, Python, PHP, Ruby, more. <https://sentry.io/welcome/>. [Online; accessed 10-April-2019].
- [45] JWT. JSON Web Tokens. <https://jwt.io>. [Online; accessed 5-June-2019].
- [46] Can i use. Support tables for HTML5, CSS3, etc. <https://caniuse.com/>. [Online; accessed 10-April-2019].
- [47] man ab. Apache HTTP server benchmarking tool. <http://manpages.org/ab>. [Online; accessed 10-April-2019].
- [48] Linode. SSD Cloud Hosting & Linux Servers. <https://www.linode.com/>. [Online; accessed 10-April-2019].
- [49] Google Developers. Get Started With Analyzing Runtime Performance. <https://developers.google.com/web/tools/chrome-devtools/evaluate-performance/>. [Online; accessed 10-April-2019].
- [50] Elasticsearch. RESTful, Distributed Search & Analytics. <https://www.elastic.co/pt/products/elasticsearch>. [Online; accessed 10-April-2019].
- [51] Draw.io. Flowchart Maker & Online Diagram Software. <https://www.draw.io>. [Online; accessed 17-December-2018].
- [52] Agile Business Consortium. MoSCoW Prioritisation. <https://www.agilebusiness.org/content/moscow-prioritisation>. [Online; accessed 10-April-2019].

Anexos

Anexo A

Requisitos Funcionais

Dado a natureza do modelo de desenvolvimento escolhido, não existe uma obrigação de definir os requisitos de forma explícita e concreta, dado que estes podem ser alterados e redefinidos durante qualquer altura do desenvolvimento do projeto.

Contudo, como é um projeto de um produto já existente, alguns dos requisitos já se encontram bem definidos e detalhados para a fase de desenvolvimento descrita neste relatório no Capítulo 5 - Desenvolvimento.

Este detalhe, apesar de tornar os requisitos mais favoráveis a sofrerem alterações, torna o processo de implementação mais fácil.

A descrição dos requisitos segue a seguinte estrutura:

- **ID:** Identificador único do requisito
- **Nome:** Nome do requisito
- **Descrição:** Descrição do objetivo do requisito
- **Prioridade:** Priorização e importância da funcionalidade. É avaliada através do método MoSCoW[52].

Grupo I - Informações do Produto

ANTD-1: Landing Page

- **ID:** ANTD-1
- **Nome:** Landing Page
- **Descrição:** Página inicial com informações sobre produto e ligações para outras páginas úteis.
- **Prioridade:** *Could have*

ANTD-2: Termos e Condições

- **ID:** ANTD-2

- **Nome:** Termos e Condições
- **Descrição:** Páginas com os termos e condições
- **Prioridade:** *Should have*

ANTD-3: Política de Privacidade

- **ID:** ANTD-3
- **Nome:** Política de Privacidade
- **Descrição:** Páginas com a política de privacidade
- **Prioridade:** *Should have*

Grupo II - Autenticação e Perfil

ANTD-4: Login

- **ID:** ANTD-4
- **Nome:** Login
- **Descrição:** O utilizador efetua o login para aceder às funcionalidades principais do sistema
- **Prioridade:** *Must have*

ANTD-5: Registo

- **ID:** ANTD-5
- **Nome:** Registo
- **Descrição:** O utilizador cria uma nova conta
- **Prioridade:** *Must have*

ANTD-6: Confirmação de conta

- **ID:** ANTD-6
- **Nome:** Confirmação de conta
- **Descrição:** O utilizador confirma a conta depois de efetuar registo
- **Prioridade:** *Must have*

ANTD-7: Reenviar email de confirmação

- **ID:** ANTD-7
- **Nome:** Reenviar email de confirmação
- **Descrição:** O utilizador não recebeu o email de confirmação e precisa de pedir um novo
- **Prioridade:** *Must have*

ANTD-8: Login com Facebook

- **ID:** ANTD-8
- **Nome:** Login com Facebook
- **Descrição:** O utilizador faz login através do Facebook
- **Prioridade:** *Should have*

ANTD-9: Login com Google

- **ID:** ANTD-9
- **Nome:** Login com Google
- **Descrição:** O utilizador faz login através do Google
- **Prioridade:** *Should have*

ANTD-10: Recuperar senha

- **ID:** ANTD-10
- **Nome:** Recuperar senha
- **Descrição:** O utilizador pretende recuperar a senha
- **Prioridade:** *Must have*

ANTD-11: Visualizar perfil

- **ID:** ANTD-11
- **Nome:** Visualizar perfil
- **Descrição:** O utilizador visualiza os seus dados(email, nome, foto e descrição) de perfil
- **Prioridade:** *Should have*

ANTD-12: Editar perfil

- **ID:** ANTD-12
- **Nome:** Editar perfil
- **Descrição:** O utilizador pretende atualizar os seus dados da conta
- **Prioridade:** *Should have*

ANTD-13: Alterar tamanho de letra

- **ID:** ANTD-13
- **Nome:** Alterar tamanho de letra
- **Descrição:** O utilizador pretende aumentar o tamanho de letra em toda a aplicação
- **Prioridade:** *Should have*

ANTD-14: Logout

- **ID:** ANTD-14
- **Nome:** Logout
- **Descrição:** O utilizador desconecta-se da plataforma
- **Prioridade:** *Must have*

Grupo III - Conteúdo e Secções

ANTD-15: Listar conteúdos

- **ID:** ANTD-15
- **Nome:** Listar conteúdos
- **Descrição:** O utilizador lista todos os conteúdos disponíveis
- **Prioridade:** *Must have*

ANTD-16: Visualizar conteúdo

- **ID:** ANTD-16
- **Nome:** Visualizar conteúdo
- **Descrição:** O utilizador visualiza todas as secções de um conteúdo
- **Prioridade:** *Must have*

ANTD-17: Visualizar secção

- **ID:** ANTD-17
- **Nome:** Visualizar secção
- **Descrição:** O utilizador visualiza a informação de uma secção
- **Prioridade:** *Must have*

ANTD-18: Procurar por conteúdos/secções/anotações

- **ID:** ANTD-18
- **Nome:** Procurar por conteúdos/secções/anotações
- **Descrição:** O utilizador procura por conteúdo usando palavras chave
- **Prioridade:** *Must have*

ANTD-19: Adicionar secção aos favoritos

- **ID:** ANTD-19
- **Nome:** Adicionar secção aos favoritos
- **Descrição:** O utilizador guarda uma secção para aceder mais tarde
- **Prioridade:** *Must have*

ANTD-20: Remover secção dos favoritos

- **ID:** ANTD-20
- **Nome:** Remover secção dos favoritos
- **Descrição:** O utilizador já não tem interesse nessa secção e remove-a dos favoritos
- **Prioridade:** *Must have*

ANTD-21: Listar secções favoritas

- **ID:** ANTD-21
- **Nome:** Listar secções favoritas
- **Descrição:** O utilizador acede aos seus favoritos
- **Prioridade:** *Must have*

Grupo IV - Anotações

ANTD-22: Visualizar anotações de autores

- **ID:** ANTD-22
- **Nome:** Visualizar anotações de autores
- **Descrição:** O utilizador pretende ler as anotações de uma secção
- **Prioridade:** *Must have*

ANTD-23: Criar anotação própria

- **ID:** ANTD-23
- **Nome:** Criar anotação própria
- **Descrição:** O utilizador pretende anotar uma secção
- **Prioridade:** *Must have*

ANTD-24: Visualizar anotações próprias

- **ID:** ANTD-24
- **Nome:** Visualizar anotações próprias
- **Descrição:** O utilizador pretende ler as anotações que criou de uma secção
- **Prioridade:** *Must have*

ANTD-25: Editar anotação própria

- **ID:** ANTD-25
- **Nome:** Editar anotação própria
- **Descrição:** O utilizador pretende editar/atualizar uma anotação previamente criada
- **Prioridade:** *Must have*

ANTD-26: Remover anotação própria

- **ID:** ANTD-26
- **Nome:** Remover anotação própria
- **Descrição:** O utilizador remove uma anotação que criou numa secção
- **Prioridade:** *Must have*

ANTD-27: Listar bundles de anotações de autores

- **ID:** ANTD-27
- **Nome:** Listar bundles de anotações de autores
- **Descrição:** O utilizador pretende listar os bundles de anotações de autores disponíveis
- **Prioridade:** *Must have*

ANTD-28: Visualizar bundle de anotações de autores

- **ID:** ANTD-28
- **Nome:** Visualizar bundle de anotações de autores
- **Descrição:** O utilizador pretende ver as informações (descrição, preço, número de anotações) sobre um bundle de anotações de autor
- **Prioridade:** *Must have*

ANTD-29: Classificar bundle de anotações de autores

- **ID:** ANTD-29
- **Nome:** Classificar bundle de anotações de autores
- **Descrição:** O utilizador pretende expressar a sua opinião sobre um bundle, por isso classifica-o de 0 a 5
- **Prioridade:** *Could have*

ANTD-30: Listar bundles de anotações próprias

- **ID:** ANTD-30
- **Nome:** Listar bundles de anotações próprias
- **Descrição:** O utilizador pretende aceder a listar os bundles a que pertence
- **Prioridade:** *Must have*

ANTD-31: Criar novo bundle de anotações próprias

- **ID:** ANTD-31
- **Nome:** Criar novo bundle de anotações próprias
- **Descrição:** O utilizador pretende criar um novo bundle de anotações para um conteúdo
- **Prioridade:** *Must have*

ANTD-32: Visualizar bundle de anotações próprias

- **ID:** ANTD-32
- **Nome:** Visualizar bundle de anotações próprias
- **Descrição:** O utilizador pretende visualizar um bundle do qual faz parte
- **Prioridade:** *Must have*

Grupo V - Anotações colaborativas

ANTD-33: Convidar um utilizador através do email para um bundle de anotações próprias

- **ID:** ANTD-33
- **Nome:** Convidar um utilizador através do email para um bundle de anotações próprias
- **Descrição:** O utilizador pretende que outros utilizadores o ajudem a criar anotações
- **Prioridade:** *Must have*

ANTD-34: Aceitar um convite para um bundle de anotações próprias

- **ID:** ANTD-34
- **Nome:** Aceitar um convite para um bundle de anotações próprias
- **Descrição:** O utilizador recebeu um convite para se juntar a um bundle de anotações próprias e deseja aceitar o mesmo
- **Prioridade:** *Must have*

ANTD-35: Remover utilizador de um bundle de anotações próprias

- **ID:** ANTD-35
- **Nome:** Remover utilizador de um bundle de anotações próprias
- **Prioridade:** *Must have*

ANTD-36: Comentar uma anotação

- **ID:** ANTD-36
- **Nome:** Comentar uma anotação
- **Descrição:** Um utilizador expressa a sua opinião sobre uma anotação
- **Prioridade:** *Could have*

ANTD-37: Criar uma sugestão de alteração de anotação

- **ID:** ANTD-37
- **Nome:** Criar uma sugestão de alteração de anotação
- **Descrição:** O utilizador sugere uma anotação alternativa à atual
- **Prioridade:** *Could have*

ANTD-38: Comentar uma sugestão de alteração de anotação

- **ID:** ANTD-38
- **Nome:** Comentar uma sugestão de alteração de anotação
- **Descrição:** O utilizador expressa a sua opinião sobre uma sugestão
- **Prioridade:** *Won't have*

ANTD-39: Aceitar uma sugestão de alteração de anotação

- **ID:** ANTD-39
- **Nome:** Aceitar uma sugestão de alteração de anotação
- **Descrição:** O utilizador concorda com a sugestão e aceita-a
- **Prioridade:** *Could have*

ANTD-40: Rejeitar uma sugestão de alteração de anotação

- **ID:** ANTD-40
- **Nome:** Rejeitar uma sugestão de alteração de anotação
- **Descrição:** O utilizador não acha a sugestão correta e rejeita-a
- **Prioridade:** *Could have*

Grupo VI - Mercado

ANTD-41: Comprar bundle de anotações

- **ID:** ANTD-41
- **Nome:** Comprar bundle de anotações
- **Descrição:** O utilizador compra um bundle para obter acesso às anotações
- **Prioridade:** *Must have*

ANTD-42: Vender bundle de anotações

- **ID:** ANTD-42
- **Nome:** Vender bundle de anotações
- **Descrição:** O utilizador criou um bundle e agora pretende vendê-lo
- **Prioridade:** *Must have*

ANTD-43: Ativar tempo de experimentação

- **ID:** ANTD-43
- **Nome:** Ativar tempo de experimentação
- **Descrição:** O utilizador tem interesse em experimentar as funcionalidades *premium* da aplicação
- **Prioridade:** *Could have*

ANTD-44: Ativar código de oferta

- **ID:** ANTD-44
- **Nome:** Ativar código de oferta
- **Descrição:** O utilizador recebeu um código de oferta e tenta ativá-lo
- **Prioridade:** *Should have*