



Universidade de Coimbra
Faculdade de Ciências e Tecnologias
Departamento de Engenharia Informática
Mestrado em Engenharia Informática

Relatório Final de Estágio

Evolução da rede social Collectors Bridge

ORIENTANDO:

Miguel Barcelos

barcelos@student.dei.uc.pt

ORIENTADOR DA COLLECTORS BRIDGE:

Eng. Pedro Martins

pmartins@collectorsbridge.com

ORIENTADOR DO DEI | FCTUC:

Prof. Dr. Marco Vieira

mvieira@dei.uc.pt

Julho de 2015

Universidade de Coimbra
Faculdade de Ciências e Tecnologias
Departamento de Engenharia Informática
Mestrado em Engenharia Informática

Relatório Final de Estágio

Evolução da rede social Collectors Bridge

ORIENTANDO:

Miguel Barcelos

barcelos@student.dei.uc.pt

ORIENTADOR DA COLLECTORS BRIDGE:

Eng. Pedro Martins

pmartins@collectorsbridge.com

ORIENTADOR DO DEI | FCTUC:

Prof. Dr. Marco Vieira

mvieira@dei.uc.pt

JÚRI ARGUENTE:

Prof. Dr. Fernando Barros

barros@dei.uc.pt

JÚRI VOGAL:

Prof. Dr. Fernando Boavida

boavida@dei.uc.pt

Julho de 2015

Resumo

O colecionismo é um hábito que vem da antiguidade, derivado da atribuição de valor (afetivo, cultural ou simplesmente material) aos objetos, justificando a necessidade da sua preservação. É a prática que as pessoas têm de guardar, organizar, selecionar, trocar e expor diversos itens por categoria, em função de interesses pessoais. É neste contexto que surge a Collectors Bride, uma plataforma que pretende ser referência mundial no colecionismo, possibilitando aos colecionadores gerirem, apresentarem e transacionarem os seus artigos colecionáveis. Atualmente a plataforma encontra-se numa versão beta e está aberta ao público.

O objetivo deste projeto de estágio assenta na evolução da plataforma Collectors Bridge, cimentando o seu desenvolvimento. É utilizada a metodologia *Design Thinking* para a idealização dos módulos ou funcionalidades a desenvolver. O desenvolvimento está alinhado com o MVP (*Minimum Viable Product*) segundo estratégia LEAN e a metodologia utilizada é uma adaptação do SCRUM, funcionando com base em ciclos curtos de desenvolvimento. O trabalho resultou na adição de melhorias ao *BackOffice* e na criação de um módulo designado como “Marketplace”, que permite aos utilizadores transacionarem colecionáveis entre si, operação na qual a Collectors Bridge obterá uma pequena percentagem por cada compra/venda realizada.

Palavras-chave:

“Colecionismo”, “Collectors Bridge”, “Design Thinking”, “PayPal”, “Marketplace”, “Rede Social”. “Ruby on Rails”

Abstract

Collecting is a habit that comes from antiquity, derived from the value assignment (affective, cultural or simply material) to the objects, justifying the need for their preservation. It is the practice that people have to store, organize, select, exchange and display various items by category, due to personal interests. This is the context where Collectors Bride, a platform that aims to be a world reference in collecting, enables collectors to manage, show and transact their collectibles. Currently the platform is in a beta version and open to the public.

The goal of this internship project is based on the evolution of the Collectors Bridge platform, pulling together its development. It used the Design Thinking methodology to archive the idealization of the modules and functionality to be develop. The development is aligned with the MVP (Minimum Viable Product) according to the LEAN strategy and the methodology used is an adaptation of SCRUM, running based on short development cycles. The work resulted in the addition of improvements to the BackOffice and the creation of a module designated as the "Marketplace", which allows users to transact with each other collectibles, operation in which the Collectors Bridge will get a small percentage per purchase/sale made.

Keywords

“Collecting”, “Collectors Bridge”, “Design Thinking”, “PayPal”, “Marketplace”, “Social Networks”. “Ruby on Rails”

Agradecimentos

Collectors Bridge

Aos Engenheiros Pedro Martins e João Ramos pelo apoio e confiança que me depositaram.

Ao Miguel Correia por todo o companheirismo vivido ao longo do estágio.

A todos os elementos em que partilhei o local de trabalho pelo bom ambiente e forma acolhedora que me receberam.

Universidade de Coimbra

Ao Professor Doutor Marco Vieira por toda a disponibilidade, sugestões e apoio transmitidos ao longo deste projeto de estágio.

A todos os professores do Mestrado que contribuíram para o meu conhecimento académico e desenvolvimento pessoal.

Família e amigos

A toda a minha família e amigos que me apoiaram sempre ao longo desta caminhada.

A todos o meu obrigado!

Índice

RESUMO.....	V
ABSTRACT	VII
AGRADECIMENTOS	IX
ÍNDICE DE FIGURAS	XIII
ÍNDICE DE TABELAS.....	XVI
LISTA DE ACRÓNIMOS	XVII
1. INTRODUÇÃO.....	1
1.1. ENTIDADE DE ACOLHIMENTO E A SUA PLATAFORMA.....	1
1.2. OBJETIVOS DO ESTÁGIO	3
1.3. ESTRUTURA DO DOCUMENTO	3
2. ESTADO DA ARTE	7
2.1. O COLECIONISMO.....	7
2.2. PLATAFORMAS DE COLECIONISMO CONCORRENTES	8
2.3. COMPARAÇÃO E DISCUSSÃO DAS PLATAFORMAS	12
2.4. ENFOQUE E POSICIONAMENTO DAS PLATAFORMAS	15
3. ABORDAGEM.....	17
3.1. METODOLOGIAS DE TRABALHO.....	17
3.2. TESTES	22
3.3. TECNOLOGIAS	23
3.4. AMBIENTE DE DESENVOLVIMENTO E TESTES.....	27
3.5. PLANEAMENTO	28
3.6. RISCOS	31
4. IDEALIZAÇÃO E ESPECIFICAÇÃO	33
4.1. <i>DESIGN THINKING</i>	33
4.2. MÓDULOS SELECIONADOS PARA A IMPLEMENTAÇÃO	36
4.3. REQUISITOS NÃO FUNCIONAIS	36
4.4. ARQUITETURA GERAL DA PLATAFORMA.....	37
5. MELHORIAS DO BACKOFFICE	39
5.1. MOTIVAÇÃO E PROBLEMA.....	39
5.2. OBJETIVOS E REQUISITOS	40
5.3. DESENHO DO SISTEMA.....	40
5.4. IMPLEMENTAÇÃO	41
5.5. TESTES	43
6. MÓDULO MARKETPLACE.....	45

6.1.	MOTIVAÇÃO E PROBLEMA	45
6.2.	OBJETIVOS E REQUISITOS	45
6.3.	DESENHO DO SISTEMA.....	47
6.4.	IMPLEMENTAÇÃO	53
6.5.	TESTES	67
6.6.	LANÇAMENTO EM PRODUÇÃO	69
7.	CONCLUSÃO	71
7.1.	CONCLUSÃO E REFLEXÃO	71
7.2.	TRABALHO FUTURO	72
	REFERÊNCIAS BIBLIOGRÁFICAS.....	73
	ANEXO A - USER STORIES IMPLEMENTADAS E GRÁFICOS BURNDOWN CHART	77
	ANEXO B - ANÁLISE DE RISCOS	91
1.	DESCRIÇÃO DOS RISCOS	91
2.	AVALIAÇÃO DOS RISCOS	93
	ANEXO C - DESCRIÇÃO DOS MODELOS	97
1.	<i>BACKOFFICE</i>	97
2.	<i>MARKETPLACE</i>	97
	ANEXO D - TESTES AUTOMATIZADOS	103
1.	TESTES UNITÁRIOS	103
2.	TESTES DE REGRESSÃO.....	106
	ANEXO E - ANÁLISE DOS VALORES DAS COMISSÕES DOS CONCORRENTES	109
	ANEXO F - ANÁLISE DE SOLUÇÕES DO PAYPAL	111
	ANEXO G - MANUAL DE CONFIGURAÇÃO DO AMBIENTE DE TRABALHO	115
	ANEXO H - PROTÓTIPOS NÃO FUNCIONAIS	119
1.	INTRODUÇÃO	120
2.	MÓDULO DE PUBLICIDADE V2.....	121
3.	MÓDULO MARKETPLACE V1	124
4.	MÓDULO <i>MYSTERY</i> V1	130
3.	MÓDULO <i>FREEMIUM</i> V1	134
4.	MÓDULO <i>MESSAGE</i> V1.....	136
5.	MÓDULO DE COMENTÁRIOS V2.....	138
6.	MÓDULO EXTERNAL CONTENT V1	139
7.	MÓDULO DE VÍDEO V1	140
	ANEXO I - TESTES FUNCIONAIS MANUAIS	143

Índice de Figuras

FIGURA 1-1 - <i>SCREENSHOT</i> DA PÁGINA INICIAL DO PORTAL COLLECTOR BRIDGE (2015/06/30).....	2
FIGURA 2-1 – <i>SCREENSHOT</i> DA PÁGINA INICIAL DO COLNECT	10
FIGURA 2-2 - <i>SCREENSHOT</i> DA PÁGINA DE UM UTILIZADOR AUTENTICADO DO COLLECTORS WEEKLY	11
FIGURA 2-3 - <i>SCREENSHOT</i> DA PÁGINA DE UM UTILIZADOR AUTENTICADO DO COLLECTORS QUEST	11
FIGURA 2-4 - <i>SCREENSHOT</i> DA PÁGINA DE UM UTILIZADOR AUTENTICADO DO KOLEKTADO.....	12
FIGURA 3-1 - CICLO DE VIDA DO <i>DESIGN THINKING</i>	19
FIGURA 3-2 – <i>SCRUM PROCES</i> [23].....	20
FIGURA 3-3 - ARQUITETURA DO RUBY ON RAILS [31]	25
FIGURA 3-4 – PLANEAMENTO DO PRIMEIRO SEMESTRE	28
FIGURA 3-5 - <i>SCREENSHOT</i> DO <i>DASHBOARD</i> DESENVOLVIDO.....	29
FIGURA 3-6 – PLANEAMENTO INICIAL DO SEGUNDO SEMESTRE.....	30
FIGURA 3-7 – PLANEAMENTO FINAL DO SEGUNDO SEMESTRE	31
FIGURA 3-8 – MATRIZ DE EXPOSIÇÃO DE RISCOS E A CLASSIFICAÇÃO DE CADA RISCO	32
FIGURA 4-1 - MODELO FÍSICO DA COLLECTORS BRIDGE	37
FIGURA 4-2 – MODELO LÓGICO DA COLLECTORS BRIDGE.....	38
FIGURA 5-1 – <i>SCREENSHOT</i> DA PÁGINA INICIAL ANTIGA DE UM CHAMPION NO <i>BACKOFFICE</i>	40
FIGURA 5-2 - DIAGRAMAS DE MODELOS E CONTROLADORES DA <i>SHOUTBOX</i>	41
FIGURA 5-3 - <i>SCREENSHOT</i> DA PÁGINA INICIAL DE UM CHAMPION NO <i>BACKOFFICE</i> COM AS MELHORIAS IMPLEMENTADAS	41
FIGURA 5-4 - FICHEIROS ADICIONADOS E ALTERADOS COM AS MELHORIAS DO <i>BACKOFFICE</i>	42
FIGURA 6-1 – PROTÓTIPO NÃO FUNCIONAL DO <i>MARKETPLACE</i>	46
FIGURA 6-2 - DIAGRAMA DE ESTADOS DE UM ARTIGO	49
FIGURA 6-3 - DIAGRAMA DE ESTADOS DE UMA LICITAÇÃO/COMPRA.....	50
FIGURA 6-4 - DIAGRAMA DOS MODELOS DO <i>MARKETPLACE</i>	52
FIGURA 6-5 – DIAGRAMA DE CONTROLADORES DO <i>MARKETPLACE</i>	53
FIGURA 6-6 – <i>SCREENSHOUT</i> DA LISTAGEM DE ARTIGOS NO <i>MARKETPLACE</i>	54
FIGURA 6-7 – <i>SCREENSHOUT</i> DA PÁGINA DE UM ARTIGO NO <i>MARKETPLACE</i>	54
FIGURA 6-8 - PAINEL DE COMPRA E VENDAS	57
FIGURA 6-9 – PAINEL DE UMA COMPRA.....	57
FIGURA 6-10 - PAYPAL CHAINED PAYMENTS.....	58
FIGURA 6-11 - DIAGRAMA DE SEQUÊNCIA DA CRIAÇÃO DE UM PEDIDO DE PAGAMENTO NO PAYPAL.....	59
FIGURA 6-12 – SEQUENCIA DE UM PAGAMENTO BEM-SUCEDIDO EXECUTADO POR UM COMPRADOR NA C’B	61
FIGURA 6-13 – DIAGRAMA DE SEQUÊNCIA DA TROCA DE MENSAGENS IPN E OBTENÇÃO DO <i>SHIPPING ADDRESS</i> ENTRE A C’B E PAYPAL.....	61
FIGURA 6-14 – FORMULA DAS CLASSIFICAÇÕES DE UTILIZADORES NO <i>MAREKETPLACE</i>	63
FIGURA 6-15 – ANATOMIA DO <i>MARKETPLACE</i>	66
FIGURA A-1 - <i>BURNDOWN CHART</i> DO 1º <i>SPRINT</i>	79

FIGURA A-2 - <i>BURNDOWN CHART</i> DO 2º <i>SPRINT</i>	82
FIGURA A-3- <i>BURNDOWN CHART</i> DO 3º <i>SPRINT</i>	84
FIGURA A-4 - <i>BURNDOWN CHART</i> DO 4º <i>SPRINT</i>	87
FIGURA A-5 - <i>BURNDOWN CHART</i> DO 5º <i>SPRINT</i>	90
FIGURA B-1 - EVOLUÇÃO DA MAGNITUDE DOS RISCOS DURANTE O PROJETO	95
FIGURA E-1 - VALOR DAS COMISSÕES NA PLATAFORMA COLLECTORS CORNER	110
FIGURA F-1 – JANELA DO PAYPAL VISTA PELO COMPRADOR NUM ATO DE PAGAMENTO	111
FIGURA F-2 – OPÇÕES DE PAGAMENTOS DE TAXAS PAYPAL	112
FIGURA F-3 - EXEMPLO DE TRANSAÇÃO NO MODO DE PAGAMENTO ACORRENTADO, SENDO O VENDEDOR O PRIMEIRO RECETOR E A C'B O SEGUNDO RECETOR.	113
FIGURA H-1 - PROCESSO ITERATIVO DO <i>PROTOTYPING</i>	120
FIGURA H-2 - <i>LINK</i> DE <i>ADVERTISING</i> NO MENU DO UTILIZADOR	121
FIGURA H-3 - PÁGINA DE <i>ADVERTISING</i>	122
FIGURA H-4 - PASSO 1 PARA ADICIONAR UM NOVO <i>ADVERTISE</i>	122
FIGURA H-5 - PASSO 2 PARA ADICIONAR UM NOVO <i>ADVERTISE</i>	123
FIGURA H-6 - PAINEL DE MONITORAMENTO DE UM <i>ADVERTISE</i>	123
FIGURA H-7 - PÁGINA INICIAL DO <i>MARKETPLACE</i>	125
FIGURA H-8 - <i>POPUP</i> DE ADIÇÃO DE UM ARTIGO AO <i>MARKETPLACE</i>	125
FIGURA H-9 - PÁGINA DE UM ARTIGO À VENDA	126
FIGURA H-10 - PRIMEIRO PASSO DA COMPRA POR PARTE DO COMPRADOR.....	126
FIGURA H-11 - CONFIRMAÇÃO DO PRIMEIRO PASSO DA COMPRA POR PARTE DO COMPRADOR	126
FIGURA H-12 - PÁGINA DE UM PRODUTO COLOCADO À VENDA, VISTA PELO SEU VENDEDOR	127
FIGURA H-13 - ARTIGO PAGO PELO COMPRADOR, À ESPERA DO ENVIO POR PARTE DO VENDEDOR (VISTA DO VENDEDOR).....	128
FIGURA H-14 - ULTIMO PASSO DO PROCESSO DE COMPRA/VENDA DE UM ARTIGO (CLASSIFICAÇÃO DOS DOIS UTILIZADORES)	128
FIGURA H-15 - PREÇO ACEITO PELO VENDEDOR (VISTO PELO COMPRADOR)	129
FIGURA H-16 - <i>POPUP</i> DE <i>CHECKOUT</i> DE ARTIGO	129
FIGURA H-17 - PÁGINA INICIAL DO <i>MYSTERY</i>	130
FIGURA H-18 - EXEMPLO DO FILTRO A APLICAR À LISTAGEM DOS ITENS	131
FIGURA H-19 - INSERÇÃO DE UM NOVO COLECIONÁVEL NO <i>MYSTERY</i>	131
FIGURA H-20 - PÁGINA RELATIVA A UM COLECIONÁVEL DO <i>MYSTERY</i> AINDA NÃO RESOLVIDO (VISTA POR UM UTILIZADOR).....	132
FIGURA H-21 - PÁGINA RELATIVA A UM COLECIONÁVEL DO <i>MYSTERY</i> JÁ RESOLVIDO (VISTA POR UM UTILIZADOR)	132
FIGURA H-22 - PÁGINA RELATIVA A UM COLECIONÁVEL DO <i>MYSTERY</i> (VISTA POR UM CHAMPION OU AUTOR)....	133
FIGURA H-23 - SEPARADOR <i>MYSTERY</i> DENTRO DE UM PORTAL	133

FIGURA H-24 - <i>LINK</i> PARA O UPGRADE NO MENU DE UTILIZADOR.....	134
FIGURA H-25 - EXEMPLO DA PÁGINA DE <i>UPGRADE</i>	135
FIGURA H-26 - <i>LINK</i> PARA GERIR AS FUNCIONALIDADES <i>PREMIUM</i> E FOTO DE PERFIL COM CRACHÁ.....	135
FIGURA H-27 - PÁGINA MY PREMIUM.....	135
FIGURA H-28 - BOTÃO DE MENSAGENS COM O SEU COMPORTAMENTO.....	136
FIGURA H-29 - MENSAGENS ENTRE DOIS UTILIZADORES.....	137
FIGURA H-30 - PÁGINA DE UTILIZADOR COM BOTÃO DE ENVIO DE MENSAGEM.....	137
FIGURA H-31 - NOVA MENSAGEM.....	138
FIGURA H-32 - EXEMPLO DE COMENTÁRIOS.....	139
FIGURA H-33 - EXEMPLO DE UMA POSSÍVEL INCORPORAÇÃO DA WIKIPEDIA NUM SEPARADOR DE UM ARTIGO ..	140
FIGURA H-34 - GALERIA DE UM ARTIGO COM 2 VÍDEOS ..	141
FIGURA H- 35 - VÍDEO EM EXECUÇÃO ..	141

Índice de Tabelas

TABELA 2-1 - CONCORRENTES DA COLLECTORS BRIDGE	9
TABELA 2-2 – COMPARAÇÃO DE FUNCIONALIDADE E ATRIBUTOS DAS PLATAFORMAS CONCORRENTES DA COLLECTORS BRIDGE (ATUALIZADO A 13/10/2014)	13
TABELA 2-3 – AVALIAÇÃO DA PERFORMANCE DAS PLATAFORMAS PELO YSLOW (REALIZADO A 20/10/2014).....	15
TABELA 2-4 – POSICIONAMENTO DAS PLATAFORMAS, SEGUNDO OS FATORES CRÍTICOS DE SUCESSO DA COLLECTORS BRIDGE	16
TABELA 5-1 - TESTES DA VELOCIDADE DE CARREGAMENTO DA PÁGINA INICIAL DO BACKOFFICE COM E SEM CACHE.....	43
TABELA 6-1 - TESTES DA VELOCIDADE DE CARREGAMENTO DA PÁGINA INICIAL DAS COLEÇÕES E <i>MARKETPLACE</i> COM E SEM <i>CACHE</i>	68
TABELA 6-2 - TESTES DA VELOCIDADE DE CARREGAMENTO DA PÁGINA DE DETALHE DE UM ITEM DAS COLEÇÕES E <i>MARKETPLACE</i> COM E SEM <i>CACHE</i>	68
TABELA A-1 - <i>USER STORIES</i> DO 1º <i>SPRINT</i>	79
TABELA A-2 - <i>USER STORIES</i> DO 2º <i>SPRINT</i>	81
TABELA A-3 - <i>USER STORIES</i> DO 3º <i>SPRINT</i>	84
TABELA A-4 - <i>USER STORIES</i> DO 4º <i>SPRINT</i>	87
TABELA A-5 - <i>USER STORIES</i> DO 5º <i>SPRINT</i>	90
TABELA B-1 – AVALIAÇÃO DOS RISCOS DURANTE O PROJETO.....	94
TABELA E-1 - COMPARAÇÃO DAS COMISSÕES ENTRE PLATAFORMAS.....	110

Lista de Acrónimos

API	Application Programming Interface
C'B	Collectors Bridge
CRUD	Create Read Update Delete
CSS	Cascading Style Sheets
DNS	Domain Name System
ER	Entity-Relationship
GMT	Greenwich Mean Time
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
IP	Internet Protocol
IPN	Instant Payment Notification
MVC	Model-View-Controller
MVP	Minimum Viable Product
REST	Representational State Transfer
SGDB	Sistema de Gestão de Base de Dados
SQL	Structured Query Language
URL	Uniform Resource Locator

CAPÍTULO 1

1. Introdução

Este relatório é a parte integrante escrita de um projeto de estágio, no qual se refletem as fases e tarefas que contribuíram para o resultado final do mesmo. O estágio é uma unidade curricular do segundo ano do Mestrado de Engenharia Informática do Departamento de Engenharia Informática da Universidade de Coimbra, que corresponde a um total de 42 ECTS, perfazendo um total de 336 horas no primeiro semestre e 840 horas no segundo semestre de trabalho do aluno.

O projeto foi desenvolvido na empresa Collectors Bridge, Lda., assentando-se na vertente de engenharia de *software*, iniciado em setembro de 2014 e finalizado em julho de 2015, tendo como objetivo a evolução da rede social Collectors Bridge, produto *core* da empresa. Os orientadores foram o Engenheiro Pedro Martins por parte da Collectors Bridge e o Professor Doutor Marco Vieira por parte do Departamento de Engenharia Informática da Universidade de Coimbra.

1.1. Entidade de acolhimento e a sua plataforma

Fundada em 2014 por João Ramos, Pedro Martins, Henrique Pereira e João Machado, vencedores do Arrisca-C 2014 com melhor plano de negócios e apoiada pela Incubadora do Instituto Pedro Nunes (IPN), a Collectors Bridge, Lda. é uma empresa responsável por gerir e desenvolver uma plataforma Web, sendo parte nuclear do negócio da mesma, e que se encontra disponível em “www.collectorsbridge.com”. A plataforma é um portal para colecionadores que almeja trazer a paixão associada ao colecionismo (de quaisquer tipo de artigos) para o mundo digital, encontrando-se no equilíbrio entre três fatores que a empresa considera como fatores críticos de sucesso numa plataforma deste domínio:

- Paixão – Está associada ao gosto de utilizar a solução (ser intuitiva e com *look&feel* agradável), permitindo que os colecionadores mostrem as suas coleções e deixem a sua marca na comunidade;
- Educação – Está associada à completude do catálogo, que será mantido e evoluído pelo dinamismo da comunidade;
- Mercado - É fundamental para dinamizar as trocas e comércio porque o desejo de completar as coleções é um fogo que tem de ser alimentado permanentemente.

A Collectors Bridge é uma plataforma simples e intuitiva que permite gerir e apresentar artigos colecionáveis. Está integrada com as redes sociais, permitindo a partilha e o usufruto dos “tesouros” com amigos e colegas. Cada tema no Collectors Bridge é representado sob a forma de um portal temático, que reúne os colecionares desse tema e assenta-se numa base de dados de artigos, gerida pela comunidade e dinamizada por *champions* (moderadores), conhecedores do domínio colecionável em questão. Também é possível às marcas ou personalidades públicas criarem uma página na plataforma, colocando as suas *milestones* e coleções. A plataforma foi lançada em novembro de 2014 estando atualmente a funcionar num estado beta aberto ao público e conta com quase 400 utilizadores, 25 marcas, 8 portais, 150 coleções contabilizando-se em cerca de 1800 objetos colecionáveis.

A Figura 1-1 mostra a página inicial de um utilizador autenticado, da Collectors Bridge.

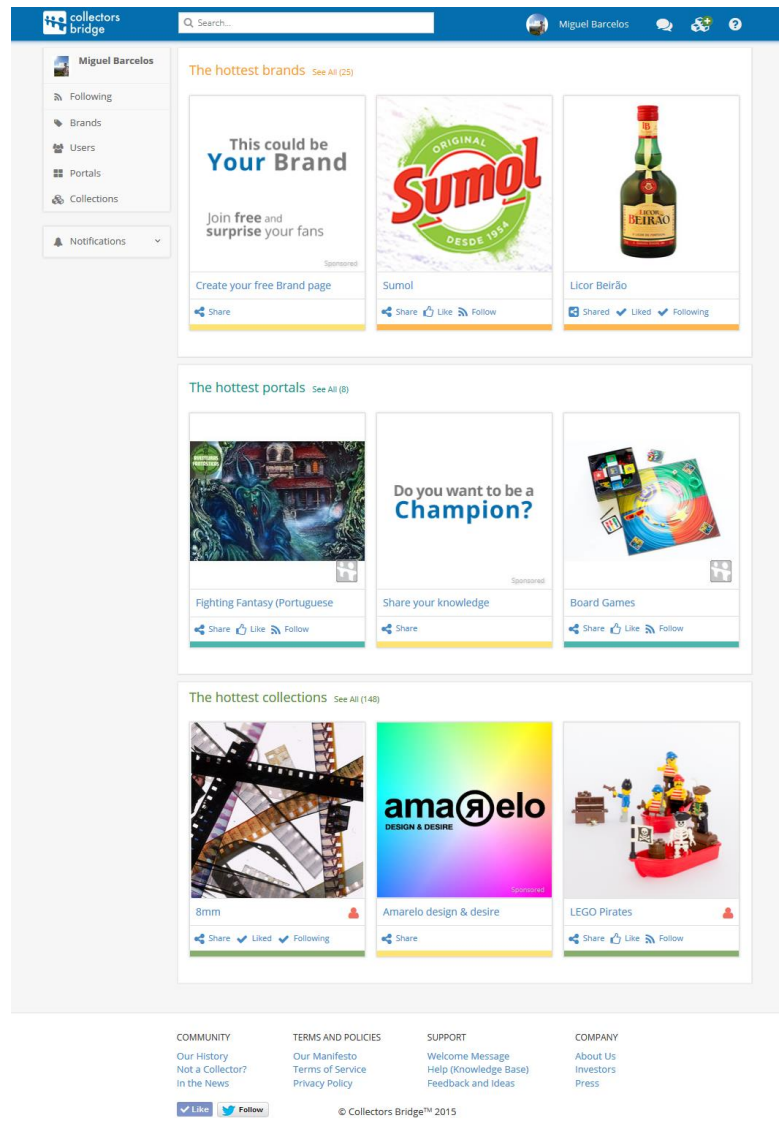


Figura 1-1 - Screenshot da página inicial do portal Collector Bridge (2015/06/30)

1.2. Objetivos do estágio

Sendo a Collectors Bridge uma plataforma que pretende ser a referência mundial do colecionismo no mundo digital, o principal objetivo do estágio foi evoluir a plataforma, cimentando o seu desenvolvimento, ou seja acrescentar ou melhorar funcionalidades.

Os objetivos do estágio foram divididos em duas partes. Na primeira parte, realizada no primeiro semestre consistiu no planeamento do estágio, definição da abordagem adotada para o projeto, e no desenvolvimento de protótipos não funcionais. A segunda parte, realizada no segundo semestre foi dedicada à implementação de pequenas melhorias ao *BackOffice* da Collectors Bridge e na adição de um novo módulo à plataforma que permitisse aos utilizadores transacionarem colecionáveis entre si.

Os objetivos das melhorias do *BackOffice* foram:

- Integração de uma shoutbox para *chamipions*;
- Pequenas melhorias de *design* e usabilidade do *BackOffice*;
- Adição de uma funcionalidade para gerir séries de um portal;

Os objetivos gerais para o módulo *marketplace* foram:

- Possibilidade do utilizador criar artigos para venda;
- Possibilidade do utilizador passar um artigo de uma coleção para venda;
- Possibilidade de um utilizador comprar artigos;
- Anonimato do comprador e vendedor;
- Perguntas e respostas sobre um determinado artigo;
- O preço dos itens ser fixo pelos vendedores, podendo ser negociável pelos compradores;
- Utilização do serviço PayPal para pagamentos;
- Pesquisa de artigos;
- Obtenção de uma pequena taxa para a Collectors Bridge por artigo comprado/vendido entre comprador e vendedor;

1.3. Estrutura do documento

Este documento encontra-se dividido por capítulos e anexos. Neste primeiro capítulo foi feita uma introdução do estágio, apresentado a empresa e os objetivos do projeto.

No segundo capítulo é apresentado o resultado de uma análise ao estado da arte do colecionismo e às plataformas existente na internet dedicadas a este tema, onde estas são analisadas e comparadas com a Collectors Bridge.

No terceiro capítulo é descrita a abordagem adotada no projeto de estágio, descrevendo as metodologias de trabalho selecionadas, a forma de testes adotada, as principais tecnologias envolvidas, o ambiente de desenvolvimento, o planeamento do estágio e por fim a forma adotada de avaliação de riscos.

No quarto capítulo são descritas todas as tarefas que contribuíram para a idealização e especificação, começando pela descrição das tarefas realizadas durante o *Design Thinking*, seguindo-se da identificação dos módulos selecionados para a implementação e por fim são apresentados os requisitos não funcionais e a arquitetura geral da plataforma.

No quinto capítulo são descritas as funcionalidades melhoradas no *BackOffice*, apresentando a motivação e problema, objetivos e requisitos, desenho do sistema e implementação e testes.

No sexto capítulo é descrito o módulo *marketplace* implementado neste estágio, apresentando a motivação e problema, objetivos e requisitos, desenho do sistema, implementação e testes e tarefas realizadas no lançamento para a produção.

No sétimo e último capítulo são descritas as principais conclusões, onde é feita uma reflexão do projeto realizado, sendo apresentadas diversas possíveis melhorias a fazer no futuro, dos conteúdos desenvolvidos.

Também estão presentes 9 anexos, sendo eles:

- A. User Stories implementadas – Este anexo descreve as *user stories* implementadas e os gráficos *burndown chart*, referentes a cada *sprint* realizado;
- B. Análise de riscos – Neste anexo está contida a análise de riscos efetuada durante o decorrer do projeto;
- C. Descrição dos Modelos – Neste anexo são descritas todos os modelos, desenvolvidas durante o estágio;
- D. Testes automatizados – Neste anexo é listado todos os testes unitários e de regressão implementados;
- E. Análise dos valores das comissões dos concorrentes – Este anexo é resultado da análise que ajudou a empresa a tomar uma decisão em relação à percentagem a cobrar por compra/venda no *marketplace*;

- F. Análise de soluções do PayPal – Neste anexo são analisadas as formas que o PayPal disponibiliza que permitem transações entre um comprador e vendedor, podendo ser retirada uma comissão para a Collectors Bridge;
- G. Manual de configuração do ambiente de trabalho – Este anexo contém um pequeno manual, criado antes de começar o desenvolvimento do código, que tem como objetivo ajudar os próximos desenvolvedores da Collectors Bridge a configurar e instalar mais facilmente o ambiente de desenvolvimento;
- H. Protótipos não funcionais – Este anexo contém todos os protótipos não funcionais desenvolvidos no primeiro semestre do estágio;
- I. Testes funcionais manuais - Neste anexo são listados todos os testes funcionais, que não foram automatizados.

CAPÍTULO 2

2. Estado da Arte

Neste capítulo é apresentado o resultado de uma análise ao estado da arte do colecionismo e às plataformas existente na internet dedicadas a este tema. Esta análise foi uma das tarefas realizadas durante a fase *needfinding* da metodologia *Design Thinking* (a ser apresentada e detalhada nos próximos capítulos), servindo assim para o estagiário e empresa conhecerem melhor as plataformas concorrentes. O capítulo começa por definir o colecionismo, depois são apresentadas diversas plataformas atualmente existentes, concorrentes à Collectors Bridge, sendo feita uma breve apresentação das quatro mais populares. Após a apresentação é feita uma discussão e comparação entre a Collectors Bridge e essas quatro plataformas mais populares concorrentes, onde é identificada uma lista de atributos e funcionalidades, e verificada a performance de carregamento das suas páginas iniciais, segundo uma lista de regras. O capítulo termina com a identificação do enfoque e posicionamento de cada solução, explicando como a Collectors Bridge cria mais valor sobre as mesmas.

2.1. O Colecionismo

O colecionismo é definido como o processo de adquirir e possuir coisas de forma ativa, seletiva e apaixonada, que fazem parte de um conjunto de objetos não idênticos e que não são utilizadas na sua forma usual [1]. Algumas das suas características são essenciais e transcendem o lugar ou tempo, outras são voláteis e correspondem ao contexto social em que estão inseridas [2]. O hábito de colecionar é tão antigo quanto a consciência humana [3] .

As motivações que levam as pessoas a colecionarem algo são a: representação do eu, prazer da caça de novos itens, competição com outros colecionadores, formar relacionamentos, busca pela imortalidade, nostalgia, desejo de suprimir frustrações e aprendizagem [4]. A nível económico o colecionismo move anualmente milhões de dólares. Segundo estatísticas existentes [5] só os Americanos gastam cerca de \$100.000 milhões anualmente em artigos colecionáveis, dos quais 20% em arte e antiguidade *high-end* e 80% num vasto conjunto de itens. Um terço dos Americanos são colecionadores e as categorias mais populares de artigos colecionáveis são:

- Moedas – 27 milhões de pessoas;
- Estatuetas/esculturas – 20 milhões;
- Cromos – 18 milhões;

- Bonecas – 16 milhões;
- Recordações – 16 milhões;
- Artigos de natal – 15 milhões;
- *Plush/beanbag toys* – 14 milhões;
- Figuras de cristal – 12 milhões;
- Pinturas/litografias – 10 milhões;
- Miniaturas – 10 milhões.

2.2. Plataformas de colecionismo concorrentes

Atualmente existem diversas plataformas *online* relacionadas com o colecionismo e concorrentes à Collectors Bridge. Com base no plano de negócios da empresa, *emails* de utilizadores e pesquisas em motores de busca, foram identificados vários possíveis concorrentes, apresentados na Tabela 2-1. A tabela apresenta também estatísticas sobre o número de *followers* do Twitter, *likes* do Facebook, utilizadores registados, ano de lançamento e país de origem.

Data: 13/10/2014	Twitter Followers	Facebook Likes	Utilizadores Registado	Ano de Lançamento	País
Collectors Weekly collectorsweekly.com	12900	28392	-	2007	USA
Collectors Quest collectorsquest.com	-	9821	-	2006	USA
Kolektado kolektado.com	272	8764	2013	2014	Israel
The Collection Room thecollectionroom.com	274	8204	-	-	USA
Colnect colnect.com	-	7424	82168	2003	Israel
Clctin clctin.com	-	5939	-	2013	Israel
Collectib.ly collectib.ly	185	5425	-	2013	Singapura
Collect.it collect.it	979	4000	-	-	USA
Delcampe delcampe.net	823	1973	-	2000	Bélgica

Curicon curicon.com	1768	1489	-	2011	Austrália
Kollectbox kollectbox.com	423	740	-	2013	Portugal
Colecção Virtual colegaovirtual.com.br	8	174	-	2012	Brasil

Tabela 2-1 - Concorrentes da Collectors Bridge

Das 12 plataformas listadas na tabela anterior, selecionou-se apenas as 4 mais populares (Collectors Weekly, Collectors Quest e Kolektado) para analisar neste capítulo. Como o sucesso da Collectors Bridge dependerá do número de aderências por parte dos utilizadores, foi utilizado esse número para definir o grau de popularidade de cada plataforma. Uma vez que quase nenhuma disponibiliza os dados sobre o número de utilizadores registados, foi necessário adicionar um segundo critério à medição do grau de popularidade, recorrendo às respetivas páginas nas redes sociais. Optou-se pelo número de *likes* do Facebook porque há dados em todas as plataformas e o número total na tabela é bastante superior ao número de *followers* do Twitter ($82345 > 17632$).

2.2.1. Colnect

O Colnect, desenvolvido e atualmente mantido por Amir Wald, é um *website* de catálogos de artigos colecionáveis, no qual estes são criados e geridos pelos seus utilizadores, permitindo de uma forma automática corresponder artigos às suas listas de posse, desejos e trocas com outros utilizadores. Todos os novos itens inseridos ou editados são sempre verificados por editores voluntários confiáveis. É oferecida a possibilidade do utilizador adquirir fisicamente artigos similares aos da plataforma com recurso ao *website* eBay. É também possível aos utilizadores contribuírem para a tradução do *website* para diversos idiomas, estando atualmente disponível em 62. Existe um serviço pago (*Premium Membership*) que possibilita mais funcionalidades ao utilizador, como por exemplo remoção de publicidade, exportação de listas para diversos formatos, entre outros [6] [7].

A Figura 2-1 ilustra a página inicial do Colnect.

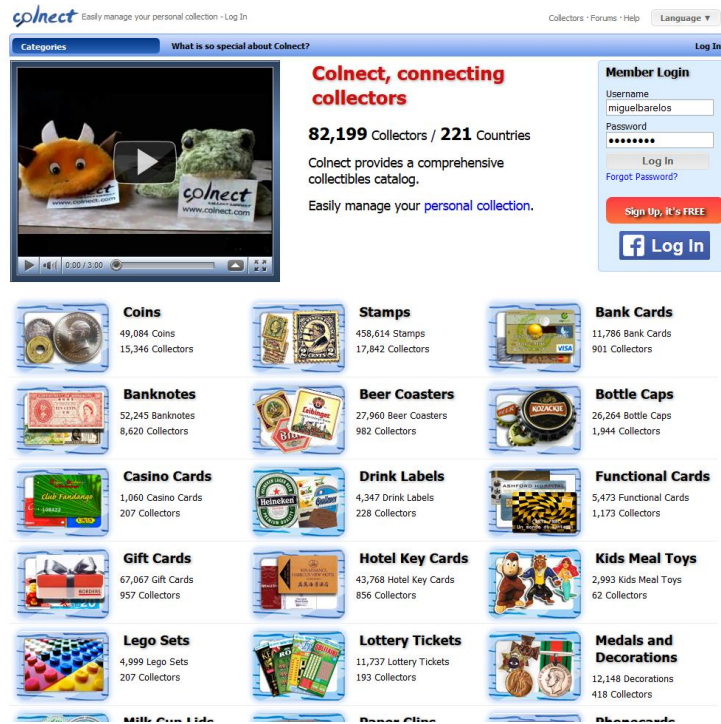


Figura 2-1 – Screenshot da página inicial do Colnect

2.2.2. Collectors Weekly

Lançado em 2007 pela empresa Market Street Media, atualmente sediada em São Francisco e dirigida por Dave Margulius, o Collectors Weekly é uma plataforma web para os interessados em antiguidade e artigos *vintage*. Tem como missão criar um lugar ótimo para explorar e aprender sobre o colecionismo, e uma *vitrine* para a paixão e conhecimento dos colecionadores de todos os lugares. Possui uma funcionalidade “*Show & Tell*” onde os utilizadores fazem *posts*, podendo a comunidade reagir fazendo “*Love*” ou “*Like*” aos mesmos. É permitido a compra e venda via eBay, mediante listagens de leilões automáticos por temática. Por fim, os colecionadores podem publicar artigos que são por vezes entrevistas com especialistas de antiguidades, e outras vezes mensagens ao estilo *blog* [8].

A Figura 2-2 ilustra a página inicial do *website*.

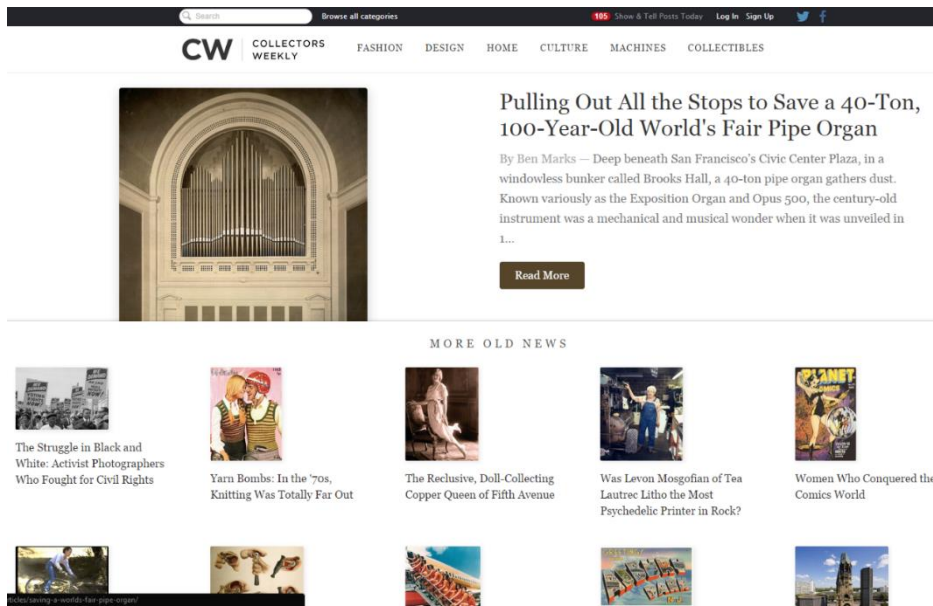


Figura 2-2 - Screenshot da página de um utilizador autenticado do Collectors Weekly

2.2.3. Collectors Quest

A Collectors Quest foi fundada pela Elizabeth Kressel, com o objetivo de ajudar todos os colecionadores de todos os tipos a tirar o máximo proveito das suas coleções. A plataforma permite aos seus utilizadores fazerem *upload* das suas coleções, conhecer e conversar com outros colecionadores, ver outras coleções, vender e comprar artigos colecionáveis e obter notícias globais do mundo do colecionismo [9] .

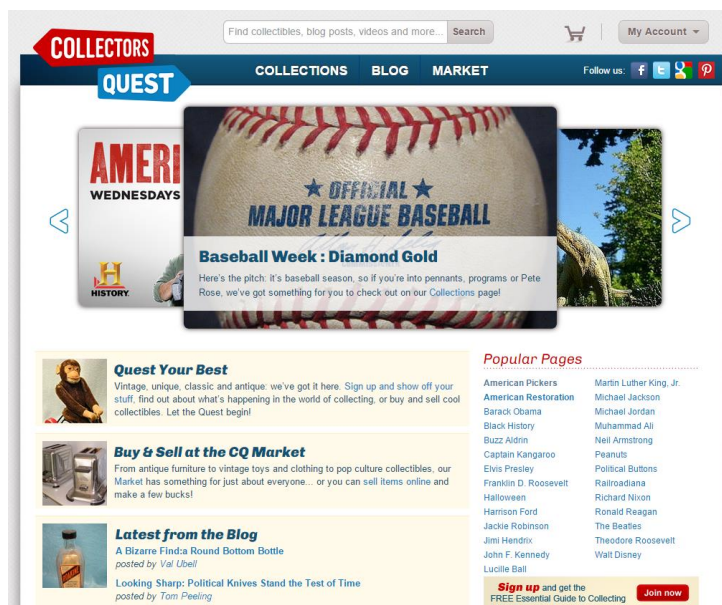


Figura 2-3 - Screenshot da página de um utilizador autenticado do Collectors Quest

2.2.4. Kolektado

Lançado em 2014 por Daniel Abramov e Vladimir Kartaviy, estando atualmente numa versão *beta*, o Kolektado afirma-se como uma comunidade mundial de colecionadores [10]. Possui funcionalidades de gestão de coleções, notificações em tempo real, pesquisa de utilizadores próximos geograficamente, trocas e vendas de artigos com outros utilizadores por leilão ou preço fixo, canais de discussão, pesquisa avançada através de filtros, entre outros [11]. A Figura 2-4 seguinte ilustra um exemplo da página inicial de um utilizador registado.

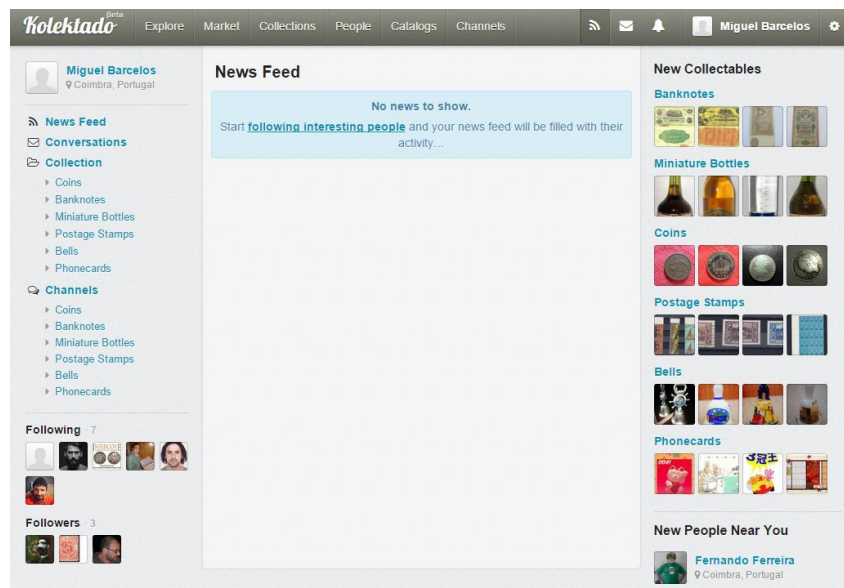


Figura 2-4 - Screenshot da página de um utilizador autenticado do Kolektado

2.3. Comparação e discussão das plataformas

Após identificadas as plataformas concorrentes, experimentou-se todas de modo a perceber quais os conceitos, atributos e funcionalidades das mesmas, e de que forma estas se distinguem da Collectors Bridge. Esta análise encontra-se na totalidade num ficheiro da empresa e resumida apenas às quatro plataformas mais populares na Tabela 2-2. Os critérios contidos na primeira coluna da tabela são referentes às diversas funcionalidades, onde algumas foram extraídas do plano de negócios da empresa e as restantes identificadas pelo estagiário durante a experimentação. São utilizados os qualificadores “sim” quando a plataforma em causa implementa o critério, “não” quando não implementa e “parcial” quando implementa de forma parcial, como por exemplo, utilizando um recurso externo. A análise à Collectors Bridge está feita de acordo com as funcionalidades esperadas na sua primeira versão pós beta.

	Colnect	Collectors Weekly	Collectors Quest	Kolektado	Collectors Bridge
Marketplace	Parcial (eBay)	Parcial (eBay)	Sim	Parcial	Sim
Possibilidade de registo e autenticação pelas redes sociais	Sim	Não	Sim	Sim	Sim
Possibilidade de envio de mensagens privadas entre utilizadores	Sim	Não	Sim	Sim	Sim
HTTPS	Parcial (Market)	Não	Não	Não	Sim
Nº de idiomas	62	1	1	53 (Google Translate)	2
Número fixo de categorias	Não	Sim	Sim	Sim	Não
Itens colaborativos	Sim	Não	Não	Não	Sim
Canais de discussão em cada categoria ou fórum	Sim	Não	Não	Sim	Não
Vista da coleção pessoal	Sim	Parcial	Sim	Sim	Sim
Museu virtual por categoria	Sim	Não	Não	Sim	Sim
Comentários de itens	Não	Sim	Sim	Sim	Sim
Blog genérico sobre colecionismo	Não	Sim	Sim	Não	Não
Rating de utilizadores	Sim	Não	Não	Não	Não
Possibilidade de seguir outros utilizadores	Não	Sim	Não	Sim	Sim
Possibilidade de fazer “like” em itens	Não	Sim	Não	Sim	Sim
Publicidade	Sim	Não	Não	Não	Sim
Otimização para mobile	Não	Não	Não	Não	Não
Notificações em tempo real	Não	Não	Não	Sim	Não
Mostra utilizadores próximos geograficamente	Não	Não	Não	Sim	Não
Pesquisa geral	Não	Sim	Sim	Não	Sim

Tabela 2-2 – Comparação de funcionalidade e atributos das plataformas concorrentes da Collectors Bridge (atualizado a 13/10/2014)

Como se pode verificar pela tabela anterior existem diversas diferenças entre as plataformas. Todas elas possuem um *marketplace*, no qual a Colnect e a Collectors Weekly o implementam com recurso externo ao eBay, a Kolektado apenas manda uma mensagem para o utilizador possuidor do item, em que o modo de transação é livre e fora da plataforma, e por fim a Collectors Quest e a Collectors Bridge implementam dentro da plataforma.

Com o objetivo de simplificar os processos de registo e autenticação, hoje em dia muitos *websites* simplificam esse processo através das redes sociais [12]. Verificou-se que esta técnica

só não está presente na Collectors Weekly. Na Kolektado e a Colnect só é possível utilizando o Facebook, no entanto a foto de perfil não é herdada na Kolektado. Por fim na Collectors Quest e a Collectors Bridge é possível não só pelo Facebook, mas também pelo Twitter e Google.

Com a exceção da Collectors Weekly, a possibilidade de troca de mensagens privadas entre utilizadores está presente em todas as plataformas. No entanto foi possível observar que todas elas, exceto a Collectors Bridge não utilizam comunicação por HTTPS, o que pode ser prejudicial para os utilizadores. A Collectors Quest utiliza HTTPS apenas no seu *marketplace*.

Em relação ao número de idiomas, todas as plataformas tem como idioma principal o Inglês. A Colnect permite que todos os utilizadores colaborem na tradução da plataforma, no que resulta atualmente em 62 traduções. A Kolektado permite outros idiomas com o uso da API do Google Translate [13]. Esta opção permite traduzir um *website* de forma automática para diversos idiomas, no entanto observou-se que por vezes a tradução pode ser feita de forma errada em alguns conteúdos da página, e aumenta a sua velocidade de carregamento, pois primeiro é carregada na língua original e só depois é traduzida.

O número fixo de categorias significa a possibilidade da comunidade adicionar novas categorias de colecionáveis, estando apenas presente na Colnect e Collectors Bridge.

Itens colaborativos significa a possibilidade de um item ser adicionado por um utilizador X e editado pelos utilizadores Y e Z, ao estilo da Wikipedia. Este conceito encontra-se na Colnect e na Collectors Bridge.

Verificou-se que existe um fórum genérico na Colnect e canais de discussão por categoria de colecionáveis na Kolektado. Sendo duas plataformas com utilizadores de diferentes idiomas, o Colnect resolveu este problema criando tópicos para cada idioma da plataforma, deixando uma possibilidade de tradução automática em todos os tópicos do fórum com recurso ao Google Translate. Na Kolektado, o utilizador seleciona os idiomas que pretende visualizar em cada canal de discussão. No entanto se tiver a utilizar a tradução automática será tudo traduzido automaticamente.

Na vertente mais social das cinco plataformas é possível visualizar a presença de diversas funcionalidades como ver a coleção de um determinado utilizador, comentar, fazer *like e share* a itens, seguir outros utilizadores e ser seguido, receber notificações em tempo real e ver os utilizadores mais próximos geograficamente.

Tendo em conta que os dispositivos móveis são cada vez mais comuns na sociedade, testou-se as plataformas num *smartphone* com a finalidade de observar o comportamento do

layout em baixas resoluções. Verificou-se que nenhuma das plataformas se adapta a telas pequenas e também não existe aplicações mobile das mesmas para os dispositivos móveis.

Por fim observou-se que apenas a Collectors Weekly e a Collectors Quest possuem uma pesquisa geral e sempre presente em todas as páginas da plataforma. As outras apenas possuem em determinadas partes da plataforma, como por exemplo categorias ou utilizadores.

Com a finalidade de verificar e analisar a performance carregamento da página inicial (com um utilizador autenticado) da Collectors Bridge, comparando com as restantes quatro plataformas (nas mesmas condições da Collectors Bridge), foram efetuados testes com recurso à ferramenta YSlow [14]. Esta ferramenta analisa *websites*, atribuindo uma classificação e sugerindo formas para melhorar o seu desempenho, com base num conjunto de 23 regras para páginas *web* de alta performance [15]. A Tabela 2-3 contém a avaliação global das cinco plataformas, no qual a pontuação varia entre 0 (mau) e 100 (bom).

	Pontuação
Colnect	87
Collectors Weekly	88
Collectors Quest	82
Kolektado	79
Collectors Bridge	76

Tabela 2-3 – Avaliação da performance das plataformas pelo YSlow (realizado a 20/10/2014)

Como se pode verificar as cinco plataformas obtiveram classificações semelhantes, no entanto a Collectors Bridge é a que possui o pior desempenho. Os maiores problemas encontrados, que contribuíram para a pontuação da Collectors Bridge foram na não utilização de *Content Delivery Network*, o numero elevado de pedidos HTTP que poderá ser solucionado combinando múltiplos ficheiros de *scripts* e *CSS* num único e utilizando *image maps*, os *headers* não terem expiração no qual poderá ser resolvido adicionando um parâmetro de controlo de *cache*, a não utilização de JavaScript no fim da página HTML, a utilização de mais de 4 domínios, a não compressão de algumas componentes de texto e por fim a não redução da escala de diversas imagens da plataformas.

2.4. Enfoque e posicionamento das plataformas

O enfoque da Colnect está no seu portal multilingue, assente num catálogo de artigos tipo *wiki*, com uma boa comunidade de utilizadores de colecionáveis tradicionais. A interface com

o utilizador poderia ser bem melhor e interliga-se com o eBay. A Collectors Bridge cria mais valor do que esta solução, cobrindo os itens colecionáveis não tradicionais, diferenciando-se pela simplicidade e facilidade de utilização, com uma interface intuitiva que não requer grande curva de aprendizagem.

O enfoque do Collectors Quest assenta-se no seu *marketplace*, estando diferenciado da Collectors Bridge por ter custos por listar artigos.

O enfoque da Kolektado está na sua rede social. A Collectors Bridge cria mais valor do que esta solução, cobrindo qualquer tipo de categoria de artigos colecionáveis.

O enfoque do Collectors Weekly está no seu *blog*, focando-se em artigos individuais, com uma boa interação com a comunidade e interliga-se com o eBay. A Collectors Bridge cria mais valor do que esta solução e as duas anteriores, pois disponibiliza uma base de dados com todos os itens colecionáveis de um tema, contendo por isso informação educativa aos colecionadores mais ávidos, permitindo fazer uma avaliação realista de coleções individuais.

Por fim, a Tabela 2-4 sumariza o posicionalmente, segundo os três fatores críticos de sucesso da Collectors Bridge (**E**ducação, **P**aixão e **M**ercado), das plataformas analisadas neste capítulo. Como se pode ver, qualquer uma das quatro plataformas concorrentes não cobre de uma forma total os três fatores críticos, estando a Collectors Bridge planeada para cobrir totalmente. Esta análise de posicionalmente foi elaborada em colaboração com a empresa.

Colnect	Collectors Weekly	Collectors Quest	Kolektado	Collectors Bridge

Tabela 2-4 – Posicionamento das plataformas, segundo os fatores críticos de sucesso da Collectors Bridge

CAPÍTULO 3

3. Abordagem

Neste capítulo é descrita a abordagem adotada no projeto de estágio. Começa por descrever as metodologias selecionadas, justificando a escolha das mesmas, seguindo-se da descrição do ambiente de desenvolvimento e das principais tecnologias envolvidas no projeto, depois é apresentado o planeamento do estágio para ambos os semestres, acompanhado por um diagrama de *Gantt*, e por fim são identificados os principais riscos do projeto.

3.1. Metodologias de trabalho

Tendo em conta que o estágio esteve dividido em dois semestres, em que o primeiro teve como objetivos o planeamento, idealização especificação, e o segundo teve como objetivo implementação, optou-se por utilizar diferentes metodologias. Não se optou por utilizar uma metodologia clássica durante todo o projeto, como por exemplo o modelo *Waterfall* [16], pois a plataforma esteve em constante desenvolvimento, com várias alterações de requisitos com base no *feedback* dos seus utilizadores e prioridades da empresa.

No primeiro semestre optou-se por utilizar a metodologia de inovação, *Design Thinking*. Esta escolha está relacionada com o âmbito do estágio que se intitula como “Evolução da rede social Collectors Bridge”, no qual a empresa deu liberdade para investigar em que partes ou como a plataforma poderia evoluir, resultando assim num conjunto de funcionalidades a implementar no segundo semestre pelo estagiário ou equipa de desenvolvimento.

Para o segundo semestre selecionou-se a metodologia SCRUM de forma adaptada, pois o desenvolvimento do projeto de estágio foi realizado apenas por uma pessoa.

Devido à impossibilidade do orientador de estágio da empresa acompanhar diariamente, de forma presencial o projeto, definiu-se que o estagiário deveria enviar um *email* WOD (*Work Of the Day*) todos os dias em que estivesse na empresa com o que fez, o que irá fazer no outro dia, dificuldades e artefactos que necessitam de aprovação.

Nas duas subsecções seguintes são apresentadas as duas metodologias, sendo descritas e explicadas como foram utilizadas e adaptadas ao estágio.

3.1.1. *Design Thinking*

A metodologia *Design Thinking* [17] [18] surgiu para resolver problemas e descobrir novas soluções de modo criativo e foi desenvolvida pelo professor Rolf Faste da Stanford School, que definiu e popularizou o conceito de “design thinking” como uma forma de ação criativa. Foi adaptada à administração por David M. Kelley, colega de Faste em Stanford e fundador da IDEO [19], uma empresa norte-americana de consultoria de *design* reconhecida mundialmente, que apesar de não ter inventado o termo foi uma das primeiras formadoras de opinião sobre o tema.

O *Design Thinking* foca-se no desenvolvimento de boas soluções, com novas funcionalidades, criando novas experiências, valor e significado para os consumidores. É um processo de inovação com o foco centrado no consumidor, obtendo as suas impressões sobre produtos, serviços e processos, decifrando as suas ambições em novas soluções. Não se trata apenas de conhecer uma ideia, mas sim de a concretizar, testando-a através da prototipagem.

Como mostra a Figura 3-1, tudo começa com a identificação de um problema a ser resolvido e após definir esse problema diverge-se com o objetivo de ganhar novos *insights* (conclusões dedutivas). Esses *insights* são analisados e convertidos em novas ideias, testando-as sempre em protótipos, convergindo até chegar à solução inovadora, ou seja um protótipo final. Este processo pode ser faseado mas nunca pensado de uma forma obrigatória sequencial. Pelo contrário, cada fase permeia a outra, fazendo parte de um todo coerente. Existem vários autores que sugerem várias fases (variando normalmente entre 4 e 6) e com diversas designações possíveis para cada. Com o objetivo de simplificar o uso desta metodologia no projeto de estágio, agregou-se algumas etapas, ficando assim com as seguintes fases:

- ***Needfinding*** – Esta é uma fase de pesquisa e aquisição de novos *insights* com a finalidade de determinar novos requisitos de um novo conceito ou ideia. Esses *insights* não se obtêm através de questionários, mas sim observando, escutando, interagindo nos seus ambientes para se ter uma percepção dos seus sentimentos, preocupações e problemas ou seja, desenvolvendo uma empatia pelos possíveis utilizadores do novo conceito ou ideia;
- ***Brainstorming*** – Depois de ter vários *insights* são feitas sessões de *brainstorming* com a equipa que está a desenvolver o novo conceito ou ideia. Isto é um processo de grupo em que se analisa os diversos *insights*, promovendo a procura de novas soluções, através de métodos que facilitam a geração de novas ideias;

- **Prototyping** – As novas ideias são implementadas de forma iterativa num modelo preliminar, possibilitando a avaliação de uma determinada abordagem a partir da qual poderão sugerir novas ideias consequentes.

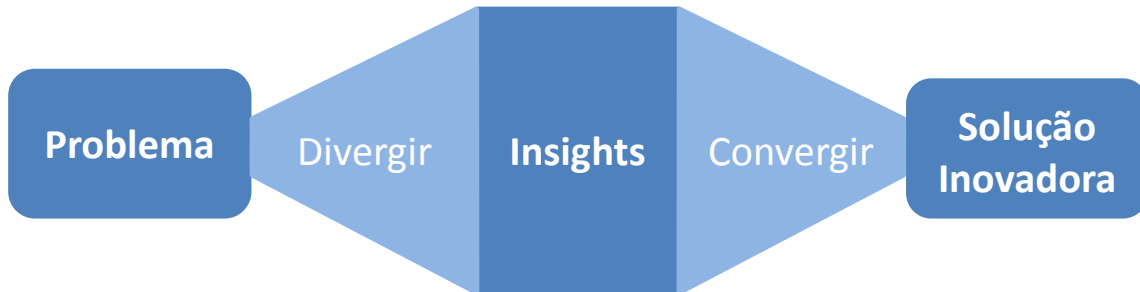


Figura 3-1 - Ciclo de vida do *Design Thinking*

3.1.2. SCRUM

O SCRUM [20] [21] [22] é um conjunto simples de princípios e práticas que ajuda equipas a entregar produtos em ciclos curtos, permitindo um *feedback* rápido, melhoria contínua e rápida adaptação à mudança. É líder do desenvolvimento *Agile*, formalizado para projetos de desenvolvimento de *software*, sendo facilmente adaptado a outras áreas. Foi a apresentado por Ken Schwaber e Jeff Sutherland na conferência de OOPLA de 1995.

Existem três diferentes papéis na metodologia SCRUM:

- **Scrum Team** – A equipe é responsável pela entrega do produto, sendo tipicamente constituída entre 5 a 9 pessoas. É auto-organizada e fazem diferentes tarefas como por exemplo analisar, projetar, desenvolver, testar, documentar, entre outras.
- **Scrum Master** – É o elemento responsável por orientar e treinar a equipa, assegurando que esta siga os valores e práticas do SCRUM. Também a protege, assegurando que a mesma não se comprometa excessivamente com o que é capaz de realizar durante um *Sprint*. Facilita as reuniões da equipa, as negociações com o *Product Owner* e remove as barreiras organizacionais para dentro da equipa.
- **Product Owner** – É a autoridade final dos requisitos, responsável por maximizar o valor do trabalho que a *Scrum Team* desenvolve, definindo e priorizando os elementos que compõem o *Product Backlog*.

A figura seguinte ilustra todo o processo do desenvolvimento de *software* com SCRUM.

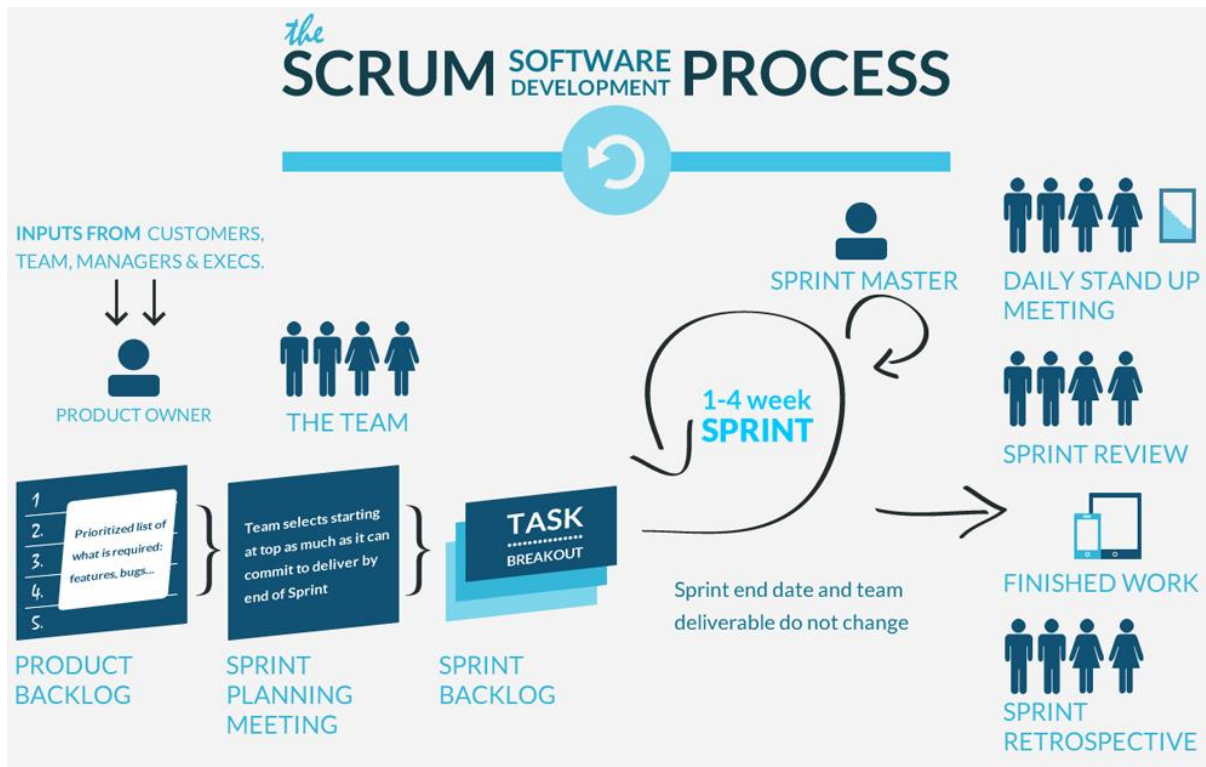


Figura 3-2 – Scrum Proces [23]

Como se pode observar pela Figura 3-2, o processo inicia-se com a preparação do *Product Backlog* pelo *Product Owner*. O *Product Backlog* é uma lista ordenada de todos os requisitos desejados para um produto. A lista não precisa de estar completa desde o início de um projeto, sendo normalmente incrementada e alterada à medida que se aprende mais sobre o produto e os seus utilizadores. Os *itens* da lista normalmente são representados por *user stories*. As *users stories* [24] são simples descrições das *features*, contada a partir da pessoa que deseja a nova funcionalidade no sistema, seguindo tipicamente o modelo: Como <tipo de utilizador>, eu quero <alguns objetivos> para que <alguma razão>.

Depois de estar definido o *Product Backlog* faz-se uma reunião com o objetivo de planear o *Sprint*, no qual o *Product Owner* prioriza os itens do *Product Backlog* e a equipa seleciona quais os que irá implementar durante o *sprint* a iniciar, sendo que esses são transferidos para o *Sprint Backlog*. Um *sprint* é um ciclo com duração típica entre duas e quatro semanas, que representa um *Time Box* dentro do qual um conjunto de atividades deve ser executada.

A cada dia de um *sprint* a equipa faz uma reunião de 15 minutos, designada por *Daily Scrum*, para que a equipa de desenvolvimento possa sincronizar as atividades e criar um plano para as próximas 24 horas. Durante a reunião cada elemento responde a três questões: O que fez hoje? O que vai fazer amanhã? Existe algum obstáculo que impeça o incumprimento da meta do *sprint*?

Ao terminar um *sprint* é esperado um incremento no produto. É realizada uma reunião de avaliação sendo tipicamente realizada uma demonstração das novas funcionalidades do produto. Os elementos participativos na reunião normalmente são a *Scrum Team*, *Scrum Master*, *Product Owner*, gestores, clientes e desenvolvedores de outros projetos.

Adaptação do SCRUM ao projeto

Sendo o SCRUM uma metodologia ágil para equipas e o projeto de estágio ser composto apenas por um elemento, foi decidido adaptar esta metodologia, simplificando-a. O estagiário desempenhou as funções de *Scrum Master* e *Scrum Team*, enquanto o orientador de estágio da empresa desempenhou a função de *Product Owner*. O estagiário também contribuiu no desenvolvimento do *Product Backlog*, no qual foi priorizado e selecionado apenas pelo orientador da empresa. Os itens que contiveram no *Product Backlog* foram requisitos de alto nível extraídos dos protótipos não funcionais, desenvolvidos pelo estagiário durante a fase de *Design Thinking*.

Os *sprints* tiveram uma duração média de três semanas, começando com uma reunião entre o estagiário e o orientador, onde se dissidiu quais as *user stories* a implementar no *sprint*, atribuindo prioridades e estimando o esforço necessário para a realização da mesma. A técnica utilizada para estimação de esforço foi executada pelo estagiário e orientador de estágio sendo baseada na *Planning Poker* [25], funcionando da seguinte forma: escolhe-se a menor história (a que se acha que se levará menos tempo a concretizar) atribuindo-se a classificação 2, numa escala de complexidade baseada na sequência de Fibonacci (0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89). A partir daí a avaliação das restantes histórias é feita, comparando com a história menor. Se uma história for superior a 20, será considerada muito grande e irá ser necessário quebra-la em várias. A velocidade do *sprint* é o número total de pontos das histórias do *sprint*. Após o primeiro *sprint* foi possível saber de uma forma mais precisa o tempo relativo a cada unidade, pois já se tinha dados para comparar.

O desenvolvimento de cada *sprint* foi iniciado após estar definido o *Sprint Backlog* e terminado após todas as *user stories* serem aprovadas pelo orientador de estágio, segundo os critérios de aceitação das mesmas.

De forma a controlar o progresso de cada *sprint*, foi construído gráficos *Burndown chart* [26], com medições de 2 em 2 dias. Estes gráficos mostram no eixo do Y o número de pontos do *sprint* e no eixo do X o nº de dias do *sprint*.

O Anexo A deste relatório contém todas as *user stories* com os respectivos critérios de aceitação, e gráficos de *Burndown chart* respetivos a cada *sprint* do estágio.

3.2. Testes

Até ser iniciado o desenvolvimento do estágio, a empresa apenas fazia testes exploratórios em duas fases:

- Alfa – Nesta fase os testes ocorrem num servidor pré-produção da Collectors Bridge, restrito a apenas alguns utilizadores, onde estes utilizam e verificam se as funcionalidades implementadas estão como o desejado;
- Beta – Nesta fase os testes ocorrem no servidor de produção da Collectors Bride (atualmente de livre acesso), onde os utilizadores podem dar feedback ou relatar algum problema ou erro encontrado.

A nível mais funcional, como por exemplo testes funcionais automatizados era inexistente, o que poderia e pode ser um problema na adição de novas funcionalidades, pois não há garantias mínimas que ao adicionar uma nova funcionalidade as já existentes continuem a funcionar corretamente. Para ter uma garantia que pelo menos as funcionalidades implementadas funcionam e continuarão a funcionar corretamente no futuro decidiu-se implementar dois tipos de testes:

- Testes unitários – Para a implementação destes testes utilizou-se a gem RSpec¹, sendo utilizada para testar procedimentos atômicos definidos nos modelos do MVC, verificando se estes funcionam bem em situações no qual recebem bons ou maus *inputs*;
- Testes de regressão – Para a implementação destes testes utilizou-se a gem Capybara², no qual esta possibilita a simulação de um utilizador navegando num *web browser*, verificando assim a presença e o comportamento dos vários elementos. Esta gem foi utilizada para verificar o correto funcionamento de todas as funcionalidades possíveis de serem testadas, presentes nas *user stories* implementadas. Para cada funcionalidade é dado cenários no formato Dado <contexto>, Quando <evento/ação>, <resultado desejado> verificando assim se a funcionalidade age corretamente, testando a integração dos vários componentes (base de dados, modelos, controladores, vistas, JavaScript, entre outros).

¹ <https://github.com/rspec/rspec-rails>

² <https://github.com/jnicklas/capybara>

Também é utilizada a gem `FactoryGirl`³, que permite criar representações de objetos em memória e agilizar a sua criação na base de dados de testes. A listagem dos testes efetuados encontra-se presente no Anexo D deste relatório.

As funcionalidades que não foram possíveis de testar de uma forma automática por algum tipo de restrição (por exemplo eventos que ocorram fora da plataforma, funcionalidades que requerem uma utilização humana, entre outros), fez-se uma lista de testes com o mesmo formato dos testes de regressão, para se poder testar manualmente. Esta lista encontra-se disponível no Anexo I deste relatório.

A nível não funcional foi testado a velocidade de carregamento das páginas e a compatibilidade com os diversos *browsers*. O resultado destes testes está presente na seção de testes dos capítulos 5 e 6.

As funcionalidades desenvolvidas também passaram pela fase Alfa nas últimas duas semanas de estágio, no qual foram feitos testes exploratórios por vários elementos da empresa. Nestes testes, além da verificação do comportamento correto em situações de utilização normal do sistema, os testadores também testaram condições alternativas e caminhos incomuns de forma a poder encontrar possíveis erros no sistema.

Testes de usabilidade estão fora do âmbito deste estágio.

3.3. Tecnologias

Nesta secção serão apresentadas de forma sucinta as principais tecnologias envolvidas no projeto, sendo elas Ruby on Rails, MySQL, Bootstrap e JQuery. Como a Collectors Bridge está desenvolvida sobre estas tecnologias, toda implementação realizada neste estágio foi com recurso às mesmas, funcionando assim como uma restrição na implementação.

3.3.1. Ruby on Rails

Rails [27] [28] é um *framework* de desenvolvimento de aplicações *web* escritas na linguagem Ruby, gratuita e *opensource* sob a licença MIT. Foi extraído por David Hansson de um projeto seu na Basecamp e lançada ao público pela primeira vez em 2005. Está desenhado para tornar a programação de aplicações *web* mais fácil, pressupondo o que cada programador precisa para começar, permitindo assim escrever menos código.

³ https://github.com/thoughtbot/factory_girl

O Ruby [29] é uma linguagem de programação orientada ao objeto, criada por Yukihiro Matsumoto e tornada pública em 1995. Algumas das suas características são:

- Todas as variáveis são objetos, até mesmo os tipos primitivos;
- É visto como uma linguagem flexível, uma vez que permite aos utilizadores alterar partes da linguagem;
- Suporta somente heranças simples, no entanto existe o conceito de módulos, em que estes são coleções de métodos;
- Não necessita de declarações de variáveis e usa convenções de nomes para denotar o âmbito das variáveis;
- Permite blocos de código passados como parâmetros para métodos;
- Capacidade de tratamento de exceções e possui *um mark-and-sweep garbage collector* para todos os objetos Ruby.

O Ruby on Rails [30] usa *Representational State Transfer* (REST) para *web services*, suporta vários sistemas de gestão de base de dados (MySQL, Oracle, MS SQL Server, PostgreSQL, IBM DB2, entre outros), usa por padrão o *Model-View-Controller* (MVC) na sua arquitetura, e a sua filosofia possui dois grandes princípios orientadores, sendo eles:

- *Don't Repeat Yourself (DRY)* – É um princípio de desenvolvimento de *software* que diz que cada pedaço de conhecimento num sistema deve possuir uma representação única, ser autoritária e livre de ambiguidades em todo o sistema.
- *Convention over configuration* – É um princípio de desenvolvimento de *software* procura diminuir o número de decisões que os desenvolvedores precisam de tomar, ganhando simplicidade sem perder a flexibilidade.

As características acima descritas estão distribuídas pelas seguintes componentes:

- Action Mailer – Esta componente é responsável pelo serviço de processamento de *emails*;
- Action Pack – Esta componente fornece as camadas de *controller* e *view* do modelo MVC e é composta por três módulos:
 - Action Dispatch – Lida com encaminhamento de *requests* do *web browser* e processa os *requests* em torno do HTTP. Depois do processamento é encaminhado para o Action Controller correspondente;
 - Action Controller – Contem ações de controlo sobre os *models* e *views*;
 - Action View – É chamada pelo Action Controller e efetua a apresentação da página web solicitada;

- Active Model – Define a interface entre o Action Pack e os módulos do Active Record.
- Active Record – Esta componente fornece o mapeamento objeto-relacional de classes. Ela constrói a camada *model*, que liga as tabelas da base de dados com as respetivas classes Ruby. Fornece também ferramentas para implementar as funcionalidades de CRUD (criação, leitura, atualização e exclusão) da base de dados;
- Active Resource – É utilizado para gerir a ligação entre os *web services* RESTfull e os objetos;
- Active Support – É uma coleção de várias classes e extensões de bibliotecas padrões que são uteis para o desenvolvimento em Ruby on Rails;

A figura seguinte ilustra como as componentes acima descritas se relacionam entre si:

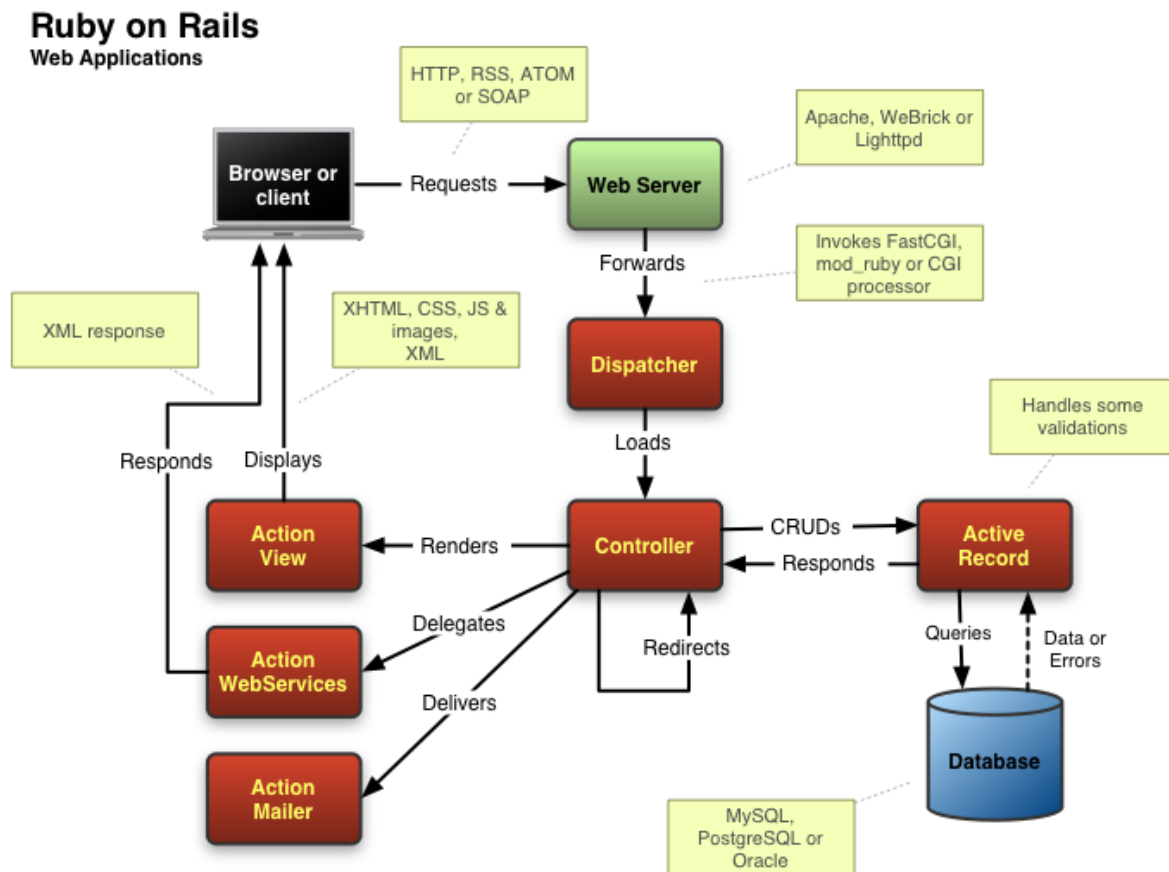


Figura 3-3 - Arquitetura do Ruby on Rails [31]

Esta linguagem de programação e *framework* é utilizada em toda a lógica de negócio da Collectors Bridge. No entanto há outras tecnologias alternativas, como por exemplo o Django⁴ e CakePHP⁵.

3.3.2. MySQL

MySQL [32] é um sistema de gestão de bases de dados (SGDB) *open source* sob a licença GPL (GNU General Public License), que utiliza a linguagem SQL como interface. É desenvolvido, distribuído e apoiado pela Oracle Corporation.

As principais características do MySQL são [33]:

- Portabilidade e compatibilidade, pois suporta praticamente qualquer plataforma atual;
- Suporta muitos tipos de dados;
- A nível de segurança, é um sistema de privilégios e senhas;
- É escalável;
- O servidor pode apresentar mensagens de erro em diferentes idiomas e possui suporte total para vários conjuntos de caracteres;
- Possui vários *softwares* de cliente por linha de comandos como por exemplo `mysqldump` e `mysqladmin` e graficamente como por exemplo o MySQL Workbench.

Este SGDB é utilizado na persistência dos dados da Collectors Bridge, no entanto existem alternativas como por exemplo o PostgreSQL⁶ e SQL Server⁷.

3.3.3. Bootstrap

O Bootstrap [34] é um *framework front-end open source* que tem como objetivo facilitar o desenvolvimento *front-end* das páginas web. Foi criado em 2010 no Twitter por Mark Otto e Jacob Thornton antes de ser um *framework open-source*, sendo denominado como *Twitter Blueprint*. Em agosto de 2010 o Twitter lançou-o como *open source* sob a licença do MIT, estando atualmente na versão v3.3.1. Está escrito em HTML, CSS, LESS, SASS e JavaScript, é compatível com as versões mais recentes de todos os principais *browsers* e suporta *web*

⁴ <https://www.djangoproject.com/>

⁵ <http://cakephp.org/>

⁶ <http://www.postgresql.org/>

⁷ <http://www.microsoft.com/pt-br/server-cloud/products/sql-server/>

design responsivo ou seja, o layout das páginas ajusta-se automaticamente, tendo em conta as características do dispositivo a ser usado.

Este *framework* está sendo utilizado no *front-end* da Collectors Bridge, no entanto existem várias alternativas semelhantes como por exemplo o Foundation⁸ ou o INK⁹ da Sapo.

3.3.4. jQuery

jQuery [35] é uma biblioteca de JavaScript rápida, pequena e rica em recursos, tendo como lema “*write less, do more*”. Foi criada por John Resin, sendo atualmente desenvolvida pela jQuery Foundation. É *open source* sob a licença do MIT, é compatível com diversos web browsers e as principais características são a manipulação do HTML/DOM, CSS e eventos, possibilita fazer efeitos e animações, e possui AJAX.

O jQuery está a ser utilizado no *front-end* da Collectors Bridge, no entanto há alternativas como ExtJS¹⁰ e MooTools¹¹.

3.4. Ambiente de desenvolvimento e testes

A implementação foi realizado numa máquina com sistema operativo Xubuntu 14.01¹², sendo utilizado como IDE o RubyMine 7¹³. Como a Collectors Bridge ainda não possui um servidor para desenvolvimento e testes, foi utilizado o servidor por defeito do Rails (WEBrick¹⁴) em *localhost*. Para testar funcionalidades que requeriam um IP público (por exemplo o envio de notificações por parte do PayPal para o servidor de desenvolvimento) foi utilizado um *software* designado por No-IP¹⁵, em que atribui um DNS ao IP dinâmico da máquina de desenvolvimento.

A Collectors Bridge possui um servidor de pré-produção que serve para testar e validar tudo antes de ir para o servidor de produção. Foi neste servidor que as funcionalidades implementadas obtiveram a validação final pela equipa da Collectors Bridge.

⁸ <http://foundation.zurb.com/>

⁹ <http://ink.sapo.pt/>

¹⁰ <http://www.sencha.com/products/extjs/>

¹¹ <http://mootools.net/>

¹² <http://xubuntu.org/>

¹³ <https://www.jetbrains.com/ruby/whatsnew/>

¹⁴ <http://ruby-doc.org/stdlib-2.2.0/libdoc/webrick/rdoc/WEBrick.html>

¹⁵ <https://www.noip.com/>

Como repositório de código foi utilizado o Bitbucket¹⁶ e como controlador de versões foi utilizado o Git¹⁷.

Para testar os pagamentos com o PayPal durante o desenvolvimento, foi utilizada a sua *sandbox*¹⁸, no qual esta simula o *website* real, oferecendo praticamente todos os serviços disponíveis no mesmo.

3.5. Planeamento

Nesta secção é apresentado o planeamento do projeto de estágio, dividido em dois semestres.

3.5.1. Primeiro semestre

No primeiro semestre trabalhou-se dois dias por semana na empresa. A figura seguinte ilustra o planeamento efetuado para o primeiro semestre, sobre a forma de um diagrama de *Gantt*.

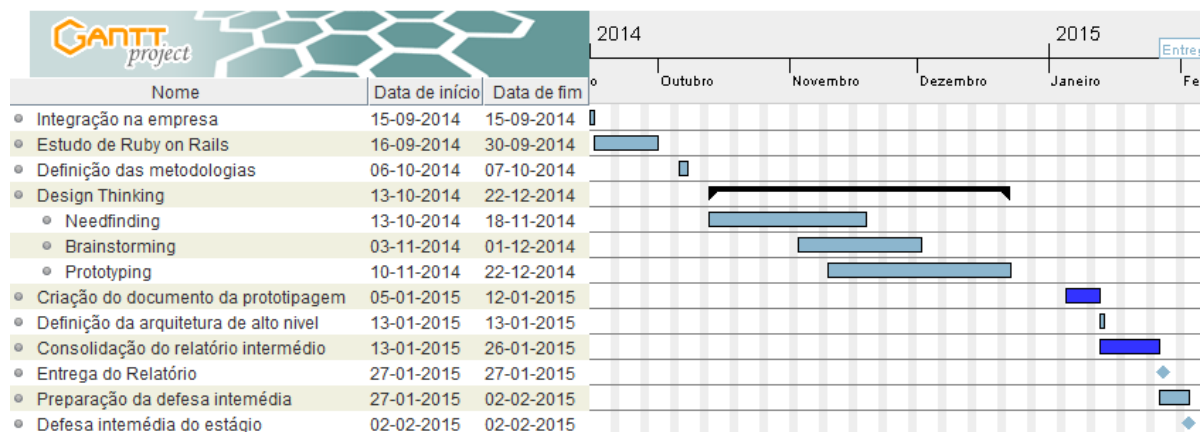


Figura 3-4 – Planeamento do primeiro semestre

O estágio iniciou-se com a integração do estagiário na empresa e como este não tinha qualquer conhecimento da tecnologia *Ruby on Rails*, foram fornecidos alguns tutoriais com o objetivo de conhecer minimamente as tecnologias envolvidas no projeto. Para verificação dos conhecimentos adquiridos foi pedido para desenvolver um pequeno *dashboard* que mostrasse dados estatísticos da plataforma entre um intervalo de tempo, com recurso às APIs do Google Analytics¹⁹ e MailChimp²⁰. O resultado está ilustrado na figura seguinte. O *dashboard* não foi

¹⁶ <https://bitbucket.org/>

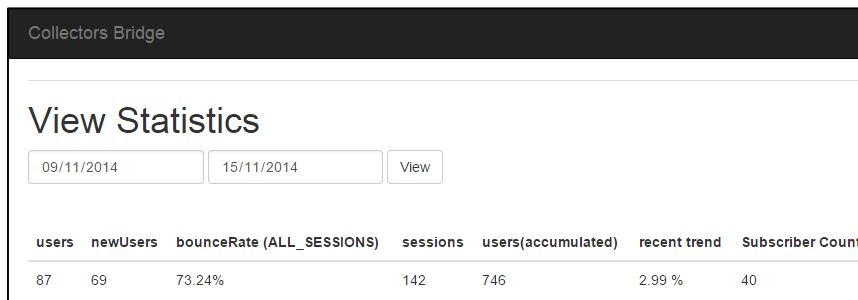
¹⁷ <https://git-scm.com/>

¹⁸ <https://www.sandbox.paypal.com/>

¹⁹ <https://developers.google.com/analytics/>

²⁰ <https://apidocs.mailchimp.com/>

para produção, no entanto partes do código implementado foram reutilizadas noutras funcionalidades por parte de elementos da empresa.



users	newUsers	bounceRate (ALL_SESSIONS)	sessions	users(accumulated)	recent trend	Subscriber Count
87	69	73.24%	142	746	2.99 %	40

Figura 3-5 - Screenshot do dashboard desenvolvido

Depois de definidas as metodologias iniciou-se a fase de *Desing Thinking*, no que terminou na criação de um documento (Anexo H) contendo todos os protótipos não funcionais desenvolvidos e a relativa especificação. Por fim foi feita a arquitetura geral da Collectors Bridge.

O relatório intermédio foi desenvolvido aos poucos ao longo de todo o semestre, no entanto foi reservado na última semana, espaço para a consolidação e correções do mesmo.

3.5.2. Segundo semestre

No segundo semestre trabalhou-se nos cinco dias uteis da semana no local da empresa. Este semestre foi dedicado à implementação. A Figura 3-6 mostra o diagrama de Gantt feito inicialmente e a Figura 3-7 mostra as alterações feitas posteriores. Como se pode observar houve redução de uma semana no primeiro *sprint* e foi adicionado uma atividade que consistiu em configurar o ambiente de desenvolvimento e contextualização no código do projeto. As *user stories* previstas de serem implementadas no primeiro *sprint* não foram afetadas com a redução do mesmo, e a atividade realizada na primeira semana deu origem ao Anexo G deste relatório, que tem como objetivo de ajudar os próximos desenvolvedores da Collectors Bridge a configurar e instalar mais facilmente o ambiente de desenvolvimento da plataforma. Esta modificação surgiu devido a mitigar um risco que consistia ao não acesso ao código por parte do estagiário antes de incitar o segundo semestre. No Anexo B é possível obter mais detalhes sobre esse risco mitigado.

Outras alterações foram no desaparecimento do sexto *sprint*, aumento do tempo para a escrita e revisão deste relatório, inclusão das tarefas de implementação dos testes automatizados e lançamento em pré-produção e produção das funcionalidades implementadas. Decidiu-se implementar os testes automatizados apenas no fim do desenvolvimento porque as

funcionalidades desenvolvidas só iriam poder entrar em produção quando a empresa contratada pela Collectors Bridge deixasse o projeto (junho), e como as funcionalidades implementadas poderiam sofrer alterações de *sprint* para *sprint*, então foi uma forma de otimizar o tempo, utilizando-o para o desenvolvimento. A tarefa de lançamento em produção das funcionalidades implementadas consistiu no processo de aprovação do *marketplace* por parte do PayPal e nos testes exploratórios realizados na fase alfa no servidor pré-produção da Collectors Bridge, por vários elementos da empresa onde estes encontraram pequenos erros que foram corrigidos.

As funcionalidades implementadas nos *sprints* foram:

- *Sprint 1* – Incidiu-se nas melhorias e adição de novas funcionalidades ao *BackOffice*;
- *Sprint 2* – Incidiu-se no desenvolvimento do desenho do sistema e implementação da estrutura inicial do *marketplace* na plataforma;
- *Sprint 3* – Incidiu-se no desenvolvimento diversas funções no *marketplace*. No final do *sprint* podia-se licitar artigos, rejeitar licitações, editar detalhes de artigos, cancelar vendas, fazer perguntas e dar respostas, ou seja quase todas as ações do diagrama de estados de um artigo, até ao ato de pagar;
- *Sprint 4* – Incidiu-se no desenvolvimento de várias funcionalidades no *marketplace*, como por exemplo pagamentos, históricos de estados, classificação de utilizadores e feedbacks. No final do *sprint* já se podia fazer compras e vendas no *marketplace*;
- *Sprint 5* – Este *sprint* incidiu-se nas funcionalidades de notificações, expirações de licitações/compras, *watchlist*, partilhas nas redes sociais e colocação de artigos no *marketplace* através de itens de coleções.

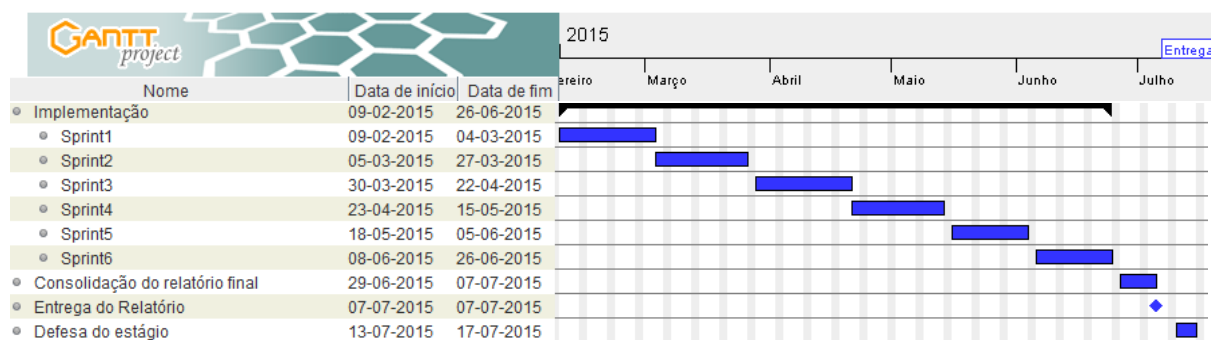


Figura 3-6 – Planeamento inicial do segundo semestre

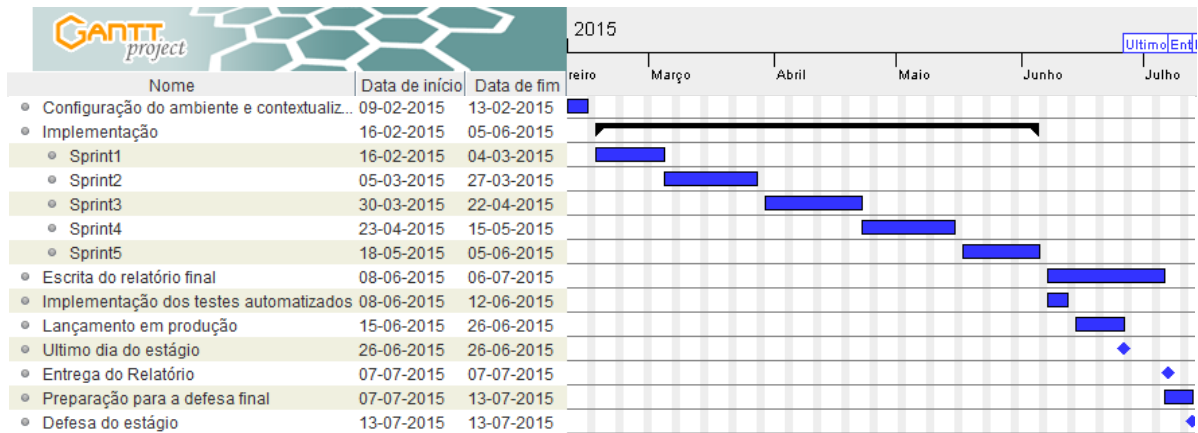


Figura 3-7 – Planeamento final do segundo semestre

3.6. Riscos

Existem sempre riscos associado a um projeto de *software*, nos quais podem ter diferentes origens e contribuir para o insucesso de um projeto. Prevenção e mitigação são estratégias que podem diminuir o impacto no projeto. No contexto deste projeto de estágio foram feitas 8 avaliações dos riscos, o que originou a identificação de 7 riscos, nos quais apenas um persiste atualmente. Esta análise encontra-se disponível detalhadamente no Anexo B deste relatório. A primeira análise de riscos foi em dezembro de 2014 e sempre que o projeto iniciou uma nova fase foi feita uma nova análise. O processo para a identificação e análise dos riscos consistiu numa sessão de brainstorming entre o orientador da empresa e estagiário. Por cada risco identificado foi atribuído um identificador, um título, uma descrição dizendo em que consiste, elaborado um plano de mitigação, atribuída uma probabilidade (baixa, média, alta e muito alta), um grau de impacto (baixo, médio, alto e muito alto) e uma magnitude (baixa, moderada, alta e crítica). A classificação da probabilidade e o grau de impacto foram atribuídos pela perceção do estagiário e orientador da empresa. A magnitude foi extraída pela Figura 3-8, no qual esta contém uma matriz de exposição de riscos que relaciona a probabilidade com o grau de impacto.

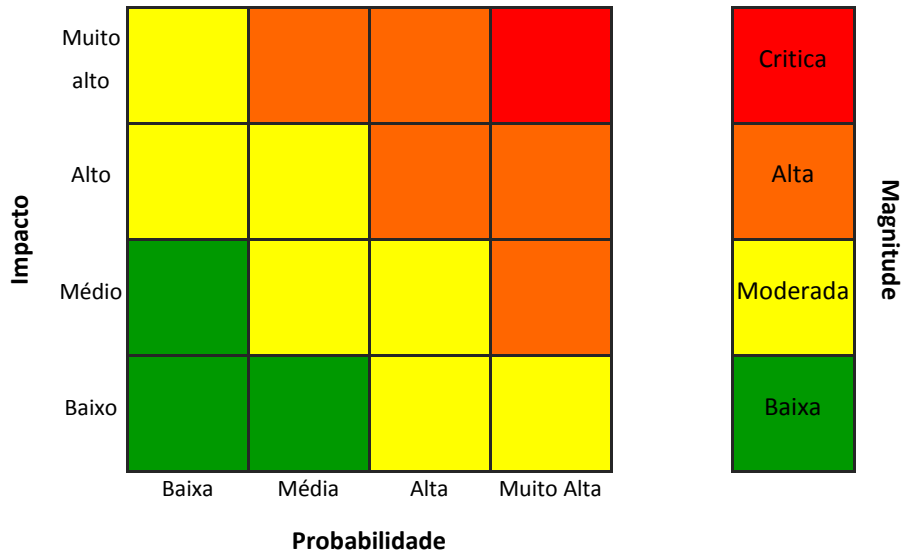


Figura 3-8 – Matriz de exposição de riscos e a classificação de cada risco

CAPÍTULO 4

4. Idealização e especificação

Neste capítulo são descritas todas as tarefas realizadas no projeto que contribuíram para a idealização e especificação durante o primeiro semestre do estágio. O capítulo começa com a descrição das tarefas realizadas durante o *Design Thinking*, seguindo-se da identificação dos módulos selecionados para a implementação, e por fim são apresentados os requisitos não funcionais e a arquitetura geral da plataforma.

4.1. *Design Thinking*

O projeto iniciou-se com a aplicação da metodologia *Design Thinking*, apresentada no capítulo anterior. Esta primeira etapa do projeto teve como objetivo investigar, propondo melhorias e novas funcionalidades para a plataforma, no qual resultou numa seleção de módulos, uns implementados no segundo semestre e outros armazenados na lista de desejos da empresa para o futuro. Como referido anteriormente, faseou-se a aplicação da metodologia em três.

4.1.1. *Needfinding*

Esta primeira fase teve como objetivo a aquisição de novos conhecimentos com a finalidade de determinar os problemas e possíveis melhorias. Começou com a leitura de alguma documentação sobre a empresa e a plataforma, e a experimentação da sua versão demo, ainda num estado de desenvolvimento, com o objetivo de contextualização no projeto. De seguida iniciou-se uma investigação das plataformas similares, resultando no segundo capítulo deste relatório com a descrição das quatro plataformas mais populares e numa atualização da folha de cálculo da empresa, contendo todos os concorrentes. Também com o objetivo de ganhar mais conhecimento no domínio da empresa e do colecionismo, assistiu-se a um *Talk* do CEO da Collector Bridge, numa aula da disciplina de PGI da Licenciatura de Engenharia Informática da FCTUC, e foi-se com a empresa ao evento Lisboa Games Week 2014. No *talk* foi abordado todo o processo de criação de uma *startup* e um produto novo, com o foco na criação da Collector Bridge. O Lisboa Games Week é uma feira de videojogos que se realizou no Parque das Nações de 6 a 9 de novembro. Este evento foi do interesse da empresa, pois continha uma parte de *retro gaming* com vários colecionadores desta área servindo assim para atrair

potenciais utilizadores para a plataforma e obter ideias de novas funcionalidades que poderiam ser uteis.

Durante toda esta fase o estagiário foi *beta tester*, acompanhado o lançamento da plataforma, e também teve acesso aos *emails* de *feedback* dos utilizadores. Por fim, foi investigado certas áreas relacionadas com os módulos decididos na fase seguinte (*brainstorming*) como por exemplo, no módulo da publicidade foi investigado as métricas da publicidade *online*, o funcionamento da publicidade nas redes sociais, *blogs* e *websites* concorrentes.

4.1.2. Brainstorming

Esta fase foi composta por várias sessões de *brainstorming* entre os elementos do projeto, com a finalidade de discutir ideias de como a plataforma poderia evoluir. Com base no trabalho e atividades participadas, sugeriu-se diversas melhorias e novas funcionalidades. Muitas delas como não eram prioritárias ficaram apenas registadas em texto na lista de desejos futuros da empresa, como por exemplo um sistema de recomendação de colecionáveis ou um *feed* do que aconteceu na plataforma. Outras melhorias sugeridas como eram muito prioritárias foram logo delegadas à equipa de desenvolvimento, como por exemplo a fusão de dois conceitos da plataforma (*showcase* e *collection*). Por fim a empresa selecionou algumas propostas e ideias para se demonstrar em protótipos não funcionais na fase seguinte.

4.1.3. Prototyping

Esta fase consistiu em aplicar as ideias recolhidas na fase anterior, de forma iterativa num modelo preliminar, dando origem posterior ao Anexo H – “Protótipos não Funcionais” deste relatório. Os módulos especificados no documento são:

Módulo Publicidade v2 – Este módulo tem como objetivo automatizar todo o processo de criação e gestão de publicidade por parte das marcas e utilizadores da plataforma, que atualmente é feito de forma manual em *backoffice*, tendo como requisitos principais:

- Possibilidade de uma marca ou utilizador inserir publicidade de forma automática na plataforma;
- Possibilidade de monitorização pelos autores de cada publicidade;
- Implementar um algoritmo de disputa de publicidade.

Módulo Marketplace v1 – Este modulo tem como objetivo a adição do 3º fator crítico de sucesso da Collectors Bridge ou seja, a criação de um *marketplace* dentro da plataforma que permita a transação de colecionáveis entre utilizadores.

Módulo Mystery v1 – Este módulo tem como objetivo fomentar o fator crítico de sucesso da educação na plataforma. Deverá permitir a qualquer utilizador da plataforma inserir colecionáveis, sabendo ou não a que portal pertencem, questionando a comunidade com objetivo de saber informações sobre o mesmo. Ao ter uma resposta conclusiva sobre o colecionável, o utilizador proprietário ou *champion* de um portal deverá dar como resolvido a questão colocada.

Módulo Freemium v1 – Este modulo tem como objetivo acrescentar o modelo de negócio *Freemium* na plataforma, ou seja, oferecer alguns recursos aos utilizador normal e criar um novo tipo de utilizador (*premium*) com funcionalidades exclusivas. As funcionalidades propostas são:

- Preferência nos resultados das pesquisas;
- Crachá *premium* na imagem de perfil do utilizador;
- Possibilidade de remoção da publicidade para o utilizador em toda a plataforma;
- Possibilidade de remoção da publicidade na página do utilizador para qualquer utilizador da plataforma;
- Saber quem viu o seu perfil;
- Imprimir as suas coleções.

Módulo Message v1 – Este módulo tem como objetivo adicionar a possibilidade dos utilizadores da plataforma comunicarem entre si.

Módulo External Content v1 – Este módulo surge num dos problemas que a Collectors Bridge está a ter na sua fase de crescimento. Assim sendo, pretende-se adicionar a possibilidade aos *champions* poderem incorporar conteúdo externo (ex. Wikipedia), sem ter de violar direitos de autor, dentro dos itens de um portal.

Módulo Comentários v2 – Este módulo tem como objetivo acrescentar as seguintes funcionalidades ao módulo de comentários já existente na plataforma:

- Possibilidade de inserir imagens nos comentários;
- Deverá fazer o *preview* do conteúdo do conteúdo de um URL inserido nos comentários.

Módulo Vídeo v1 – Este módulo tem como objetivo acrescentar a possibilidade de adição de vídeos à galeria referente a um colecionável, com a restrição de manter o formato quadrangular na visualização da lista de *thumbnails*.

4.2. Módulos selecionados para a implementação

Inicialmente dos 8 módulos acima descritos, apenas 5 estavam previstos para serem implementados no segundo semestre, onde o estagiário implementaria os módulos Publicidade v2 e *Marketplace v1* e uma empresa contratada pela Collectors Bridge implementaria os módulos *Message v1*, Comentários v2 e *External Content v1*. No entanto dado o feedback de utilizadores e prioridades administrativas da empresa, esta previsão foi alterada, resultando assim numa implementação do módulo *Marketplace v1* por parte do estagiário e implementação do módulo *Message v1* por parte da empresa contratada. Todos os restantes módulos ficaram na lista de desejos da empresa, a serem implementados no futuro.

Além do módulo *Marketplace v1* a ser implementado pelo estagiário, a empresa propôs o melhormente de pequenas funcionalidades ao BackOffice, estando descritas no capítulo seguinte.

4.3. Requisitos não funcionais

Os requisitos não funcionais testáveis para a implementação deste projeto de estágio são os mesmos da plataforma já existente e devem ser transversais em todas as suas partes. Estes são:

- Eficiência – Uma vez que a plataforma exigirá constantes trocas de informação e processamento, o tempo de execução das operações deve ser reduzido, de modo a obter-se uma eficiência aceitável. Considera-se como aceitável quando a plataforma responde em menos de 2 segundos em 90% dos casos, exceto no *upload* e *download* de conteúdos, que irá depender da dimensão dos mesmos e na comunicação com serviços externos;

- Compatibilidade – A plataforma deverá ser compatível nos *web browsers* mais comuns da atualidade, sendo eles Google Chrome 40, Mozilla Firefox 35, Internet Explorer 11 e Safari 7.1.2;

4.4. Arquitetura geral da plataforma

Do ponto de vista físico, a Collectors Bridge é constituída por cliente e servidor. Na parte do cliente (dispositivos que o utilizador usa para aceder à aplicação) são utilizadas as tecnologias HTML, CSS, JavaScript, Bootstrap e JQuery, suportadas por um *Web Browser* que comunica com o servidor por HTTP. O servidor é composto por duas instâncias. Uma é composta por um servidor Apache que por sua vez possui um servidor Phusion Passenger, em que é utilizado a *framework* Rails 4.1.4 e a linguagem Ruby 2.1.2. A outra instância é composta por um servidor MySQL. A figura seguinte ilustra o modelo físico.

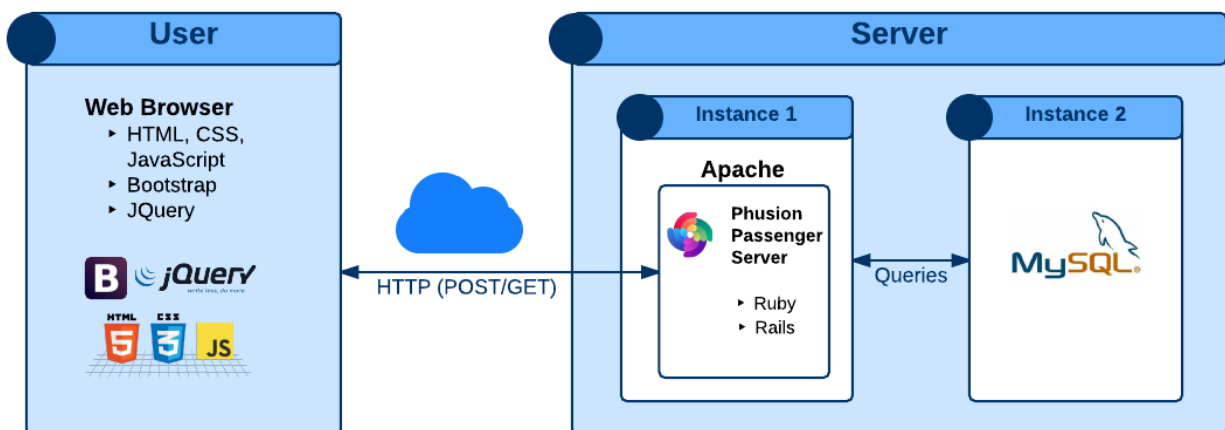


Figura 4-1 - Modelo físico da Collectors Bridge

Do ponto de vista lógico, pode-se ver a aplicação em duas perspectiva (lado do cliente e lado do servidor), tendo uma arquitetura de três camadas. A camada de cliente é responsável por apresentar a informação ao utilizador. A camada lógica é responsável por todos os processos lógicos, implementando o modelo MVC. Esta possui um *router* que recebe os *requests* e encaminha para o respetivo *controller*. O *controller* é responsável por interrogar o *model*, podendo selecionar uma *view* e responder ao pedido do cliente, o *model* é responsável por manipular a base de dados e a *view* é responsável por gerar a interface para o cliente. A utilização deste modelo possui diversas vantagens como a reutilização de código, fácil manutenção, maior facilidade no desenvolvimento em paralelo, entre outras. Por fim, a camada da persistência é responsável pela pelo armazenamento de dados.

A implementação a realizada foi integrada em todas as camadas do modelo lógico. A figura seguinte ilustra esse modelo.

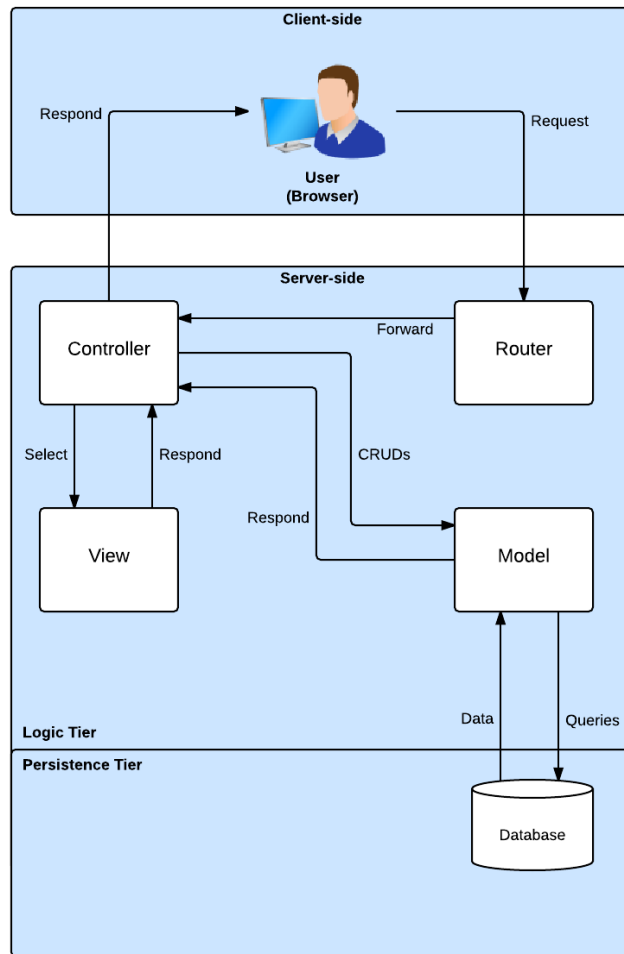


Figura 4-2 – Modelo lógico da Collectors Bridge

CAPÍTULO 5

5. Melhorias do BackOffice

Este capítulo descreve as melhorias e novas funcionalidades efetuadas ao *BackOffice* da Collectors Bridge durante o primeiro *sprint* do estágio. O capítulo inicia-se com uma descrição da motivação e problema, seguindo-se da identificação dos objetivos e requisitos, apresentação do desenho do sistema, explicação da implementação, e por fim são descritos os testes efetuados.

5.1. Motivação e problema

A Collectors Bridge possui um *BackOffice*, acedido por apenas dois tipos de utilizadores, champions e administradores. Os champions apenas tem acesso aos conteúdos dos seus portais e utilizam-no para gerir os mesmos. Os administradores possuem acesso a tudo, desde utilizadores, portais, conteúdos, entre outros, nos quais podem geri-los, ou seja, conseguem fazer CRUD (*Create, Read, Update e Delete*) a praticamente todos os conteúdos da plataforma.

Existiam alguns problemas e necessidades, principalmente para champions ao utilizarem o *BackOffice*. O primeiro problema estava relacionado com a diferença de *design* gráfico entre o *FrontOffice* e *BackOffice*. O segundo estava relacionada com a usabilidade nas caixas de pesquisa, ou seja quando um champion estava a editar um portal e pesquisava um artigo carregando na tecla “Enter”, o formulário era submetido em vez de apresentar somente resultados. O terceiro problema relacionava-se com a falta de uma funcionalidade que gerisse de forma rápida as séries do seu portal. E por fim a necessidade de ter um sistema de comunicação entre champions de um mesmo portal. Foram estas necessidades e problemas que deram origem aos requisitos deste capítulo.

A seguinte mostra um *screenshot* da página inicial de um champion no *BackOffice*, antes de ter sido implementado as melhorias.

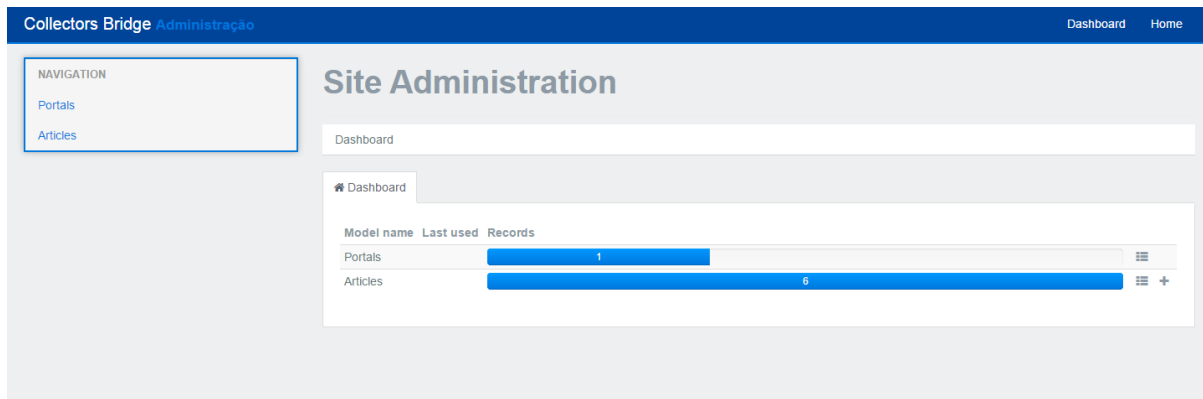


Figura 5-1 – Screenshot da página inicial antiga de um champion no *BackOffice*

5.2. Objetivos e requisitos

Os objetivos das melhorias e funcionalidades abordadas neste capítulo estão relacionados em combater os problemas descritos na secção anterior, sendo eles a integração de uma shoutbox para *champions*, melhorias de *design* e usabilidade do e adição de uma funcionalidade para gerir séries de um portal no *backoffice*.

Tal como no restante trabalho desenvolvido, no início de cada *sprint* houve uma análise dos requisitos das funcionalidades ou módulo a serem implementados, feito através de *user stories*. As *user stories* são histórias de utilização simplificadas que descrevem uma funcionalidade ou passo em específico. No Anexo A é possível ver todas as *user stories* implementadas no projeto, sendo apenas referentes a este capítulo as *user stories* identificadas com categoria *Backoffice*.

5.3. Desenho do sistema

A nível do desenho do sistema apenas fez-se alterações com a integração da shoutbox para os champion. Como se pode ver pela Figura 5-2, acrescentou-se um modelo que representa uma mensagem e adicionou-se um controlador responsável por todas as suas ações relativas à shoutbox. No Anexo C é possível visualizar mais detalhes do modelo adicionado.

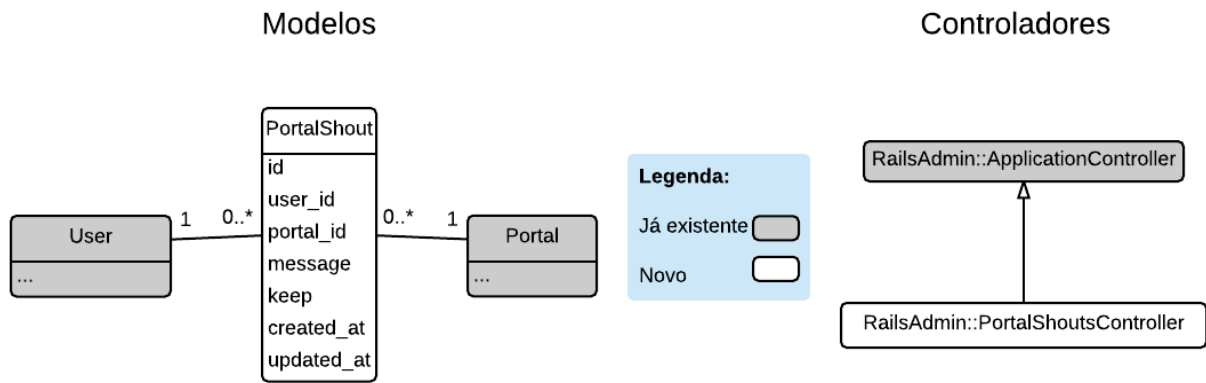


Figura 5-2 - Diagramas de Modelos e Controladores da *shoutbox*

5.4. Implementação

O *BackOffice* está construído com recurso à gem *RailsAdmin*²¹, no qual esta fornece uma interface para gerenciar dados. A autorização de conteúdos é feita com recurso à gem *Cancan*²².

A Figura 5-3 mostra um *screenshot* com o resultado final da implementação. Como se pode ver a nível de aparência gráfica colocou-se as mesmas cores e dimensões do *FrontOffice* e o logotipo da *Collectors Bridge*, sendo alterado apenas o ficheiro que contém o CSS.

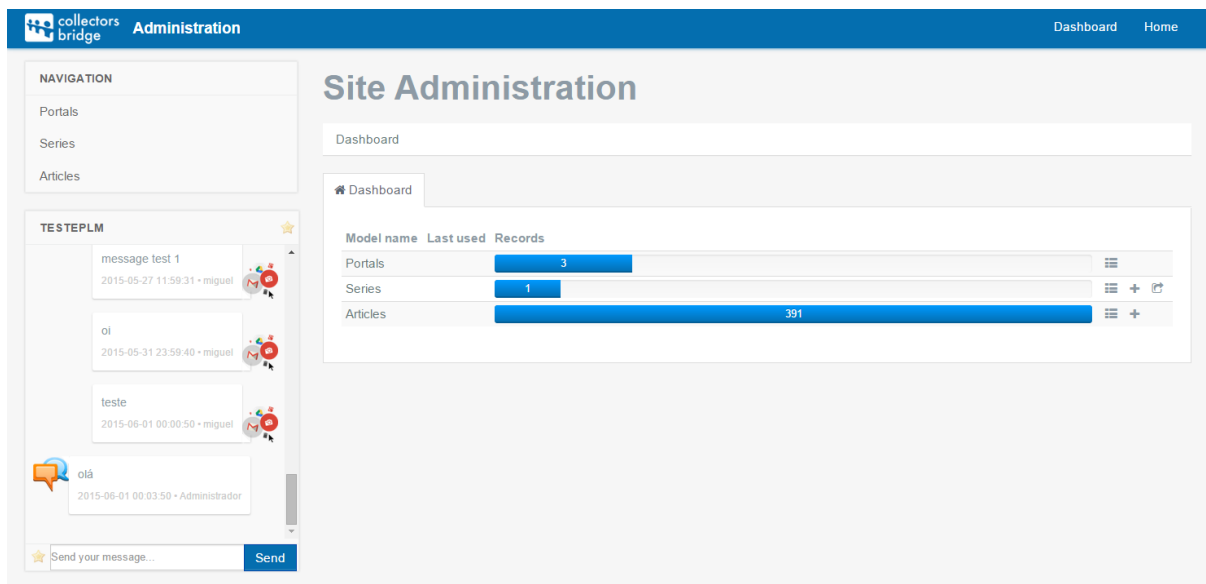


Figura 5-3 - *Screenshot* da página inicial de um champion no *BackOffice* com as melhorias implementadas

Para evitar a submissão de formulários ao carregar na tecla “enter” foi adicionado uma função JavaScript que capta o evento da tecla e ignora-o.

²¹ https://github.com/sferik/rails_admin

²² <https://github.com/ryanb/cancan>

Para adicionar a funcionalidade de gestão de séries de um portal configurou-se o RailsAdmin para apresentar dados do respetivo Modelo (PortalGroup) e as autorizações no Cancan (administrador pode ver e editar todas as séries de todos os portais e o champion apenas as séries dos seus portais).

A integração da *shoutbox* que já existia implementada mas de uma forma isolada da plataforma foi a tarefa mais complexa desta série pequenas melhorias. Decidiu-se que esta deveria ficar sempre visível, sendo debaixo do menu o local escolhido. Se o utilizador for *champion* de vários portais irá surgir várias *shoutbox*. A *shoutbox* que já existia carregava todas as mensagens do utilizador de segundo a segundo por GET, no entanto na integração melhorou-se com a obtenção apenas das mensagens que não estão carregadas. Nesta verificação é enviada por Ajax o ID da última mensagem no qual o servidor responde com as novas mensagens num formato JSON e por JavaScript é colocado nas *shoubox* respetivas.

A Figura 5-4 mostra os principais ficheiros e diretórios alterados ou adicionados, com as funcionalidades descritas neste capítulo.

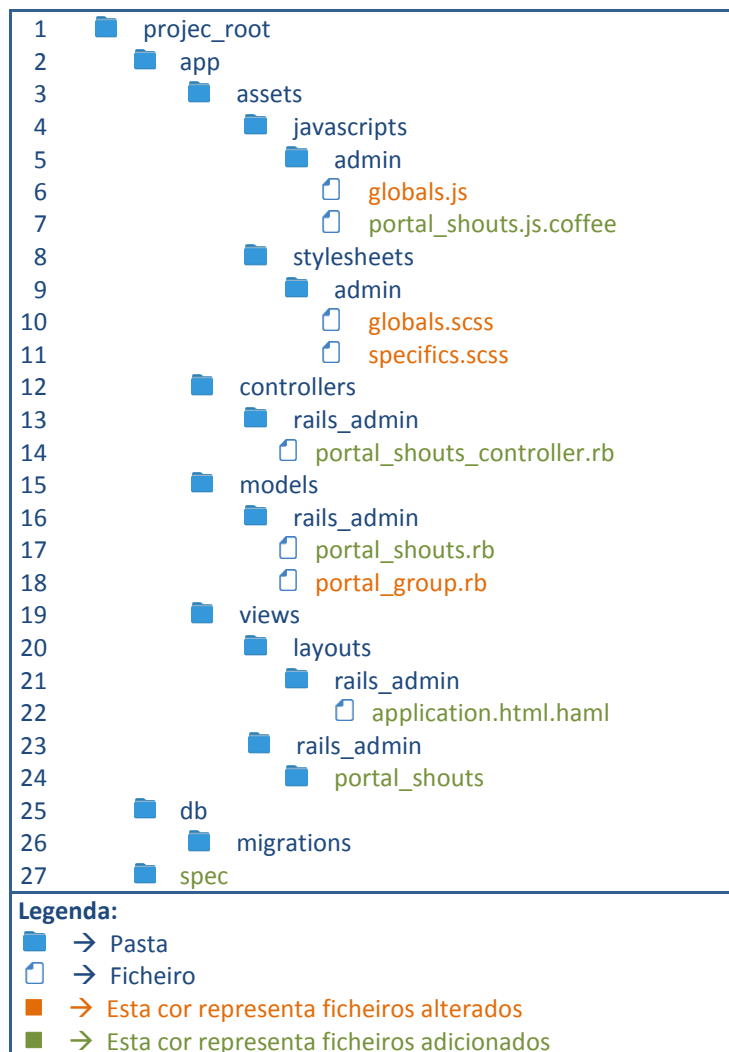


Figura 5-4 - Ficheiros adicionados e alterados com as melhorias do *BackOffice*

Os ficheiros adicionados/alterados são:

- 6 | `globals.js` – Contém todo o JavaScript do *BackOffice*;
- 7 | `portal_shouts.js.coffe` – Contém o JavaScript apenas referente à *shoutbox*;
- 10 | `globals.scss` – Contém o CSS mais global do *BackOffice*;
- 11 | `specifics.scss` – Contém o CSS mais específico do *BackOffice*;
- 14 | `portal_shouts_controller.rb` – Controlador da *shoutbox*;
- 17 | `portal_shouts.rb` – Modelo da *shoutbox*;
- 18 | `portal_group.rb` – Modelo das séries dos portais;
- 22 | `application.html.haml` – Layout do *BackOffice*
- 24 | `portal_shouts` – *Views* utilizadas na *shoutbox*;
- 27 | `spec` – Contém todos os testes automatizados da *shoutbox*.

5.5. Testes

Os testes unitários e de regressão implementados para as funcionalidades e melhorias descritas neste capítulo encontram-se no Anexo D. Passaram 4 dos 4 testes unitários e 8 dos 8 testes de regressão.

A compatibilidade das funcionalidades também foi testada nos 4 *browsers* descritos no capítulo anterior, sendo que não foi encontrada qualquer incompatibilidade.

A nível de velocidade de carregamento foi medido o tempo de carregamento da página inicial do *BackOffice* e comparado com o antes e depois da implementação. As medições foram realizadas ao servidor de pré-produção da Collectors Bridge, que se encontra em Portugal e no lado do cliente (também localizado em Portugal) foi utilizado o *browser* Google Chrome v43, numa largura de banda disponível com 12.4Mbps de *download* e 0.7Mbps de *upload*. A tabela seguinte mostra os resultados obtidos, sendo que cada resultado apresentado é uma média de 5 medições. Como se pode observar, a velocidade de carregamento não foi muito afetada com as alterações efetuadas, sendo aceitável, segundo o requisito não funcional da eficiência.

		Original	Alterado
Velocidade em segundos sem cache	Média	1.40s	1.46s
	Desvio padrão	0.14s	0.11s
Velocidade em segundos com cache	Média	0.90s	0.98s
	Desvio padrão	0.06s	0.05s

Tabela 5-1 - Testes da velocidade de carregamento da página inicial do BackOffice com e sem cache

CAPÍTULO 6

6. Módulo Marketplace

Este capítulo descreve o módulo *marketplace*, que foi desenvolvido nos últimos quatro *sprints* do estágio, estando atualmente a funcionar em produção de uma forma invisível ao público, e com lançamento e divulgação previstas para o dia 10 de julho de 2015.

O capítulo inicia-se com uma descrição da motivação e problema, seguindo-se da identificação dos objetivos e requisitos, apresentação do desenho do sistema, explicação de como foi feita a implementação, quais os testes efetuados, e por fim são descritas as tarefas realizadas no processo de lançamento em produção.

6.1. Motivação e problema

Como foi descrito anteriormente, a Colletctor Bridge assenta-se no equilíbrio entre três fatores crítico de sucesso (educação, paixão e mercado), estando apenas dois deles (educação e paixão) cobertos pela plataforma antes da realização deste estágio. O terceiro fator crítico (mercado) encontrava-se planeado para ser coberto por um *marketplace* dentro da plataforma, onde os seus utilizadores pudessem transacionar artigos colecionáveis entre si. Havia duas possibilidades de implementação, onde a primeira seria integrando com um *marketplace* já existente (por exemplo eBay²³) e a segunda criando um de raiz. Como a empresa queria obter uma pequena percentagem por transação efetuada, então foi descartada a primeira hipótese e é neste contexto que nasce o módulo *marketplace*, Este módulo é um dos mais importantes na para empresa, pois é o único que atualmente pode gerar receitas de uma forma automática.

6.2. Objetivos e requisitos

Os objetivos gerais, postos pela empresa para esta primeira versão do módulo *marketplace* foram:

- Possibilidade do utilizador criar artigos para venda;
- Possibilidade do utilizador passar um artigo de uma coleção para venda;
- Possibilidade de um utilizador comprar artigos;
- Anonimato do comprador e vendedor;

²³ <http://www.ebay.com/>

- Perguntas e respostas sobre um determinado artigo;
- O preço dos itens ser fixo pelos vendedores, podendo ser negociável pelos compradores;
- Utilização do serviço PayPal para os pagamentos;
- Pesquisa de artigos.
- Obtenção de uma pequena taxa para a Collectors Bridge por artigo comprado/vendido entre comprador e vendedor;

Estes objetivos foram transformados num protótipo não funcional (Figura 6-1), realizado no primeiro semestre, tendo pequenos ajustes no início do segundo *sprint* do desenvolvimento. Nele foi possível navegar pelo *marketplace* e executar as diversas ações antes de se começar a desenvolver código, e foi a partir daí que se extraiu as *user stories* para a construção do mesmo.

O valor da percentagem cobrada pela Collectors Bridge por cada compra/venda efetuada no *marketplace* é 2.5% do valor total do artigo vendido (com custos de envio incluídos). No Anexo E do presente relatório encontra-se uma análise que se fez aos valores das comissões, cobradas por diversos concorrentes. Foi com base nesta análise que a empresa tomou uma decisão da percentagem a cobrar.

As 54 *user stories* implementadas referentes a este capítulo encontram-se no Anexo A deste relatório, estando identificadas com a categoria “Marketplace”. No mesmo anexo também é possível visualizar quais as *user stories* que foram implementadas em cada *sprint* e os respetivos gráficos *burndown chart*.

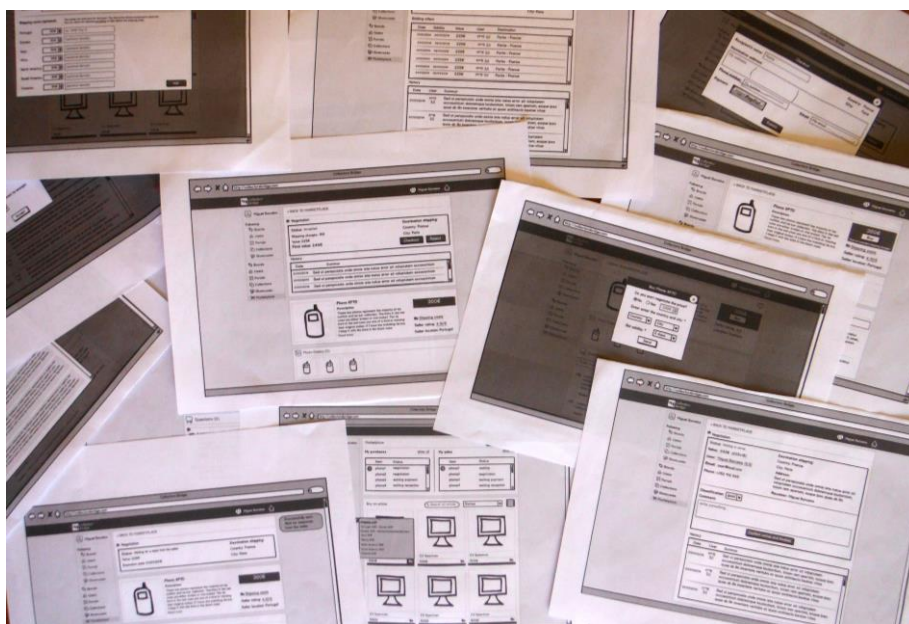


Figura 6-1 – Protótipo não funcional do *marketplace*

6.3. Desenho do sistema

Nesta secção é descrito o desenho do sistema do *marketplace*, sendo apresentados os estados de um artigo e licitação/compra, controladores e modelos adicionados.

6.3.1. Estados de um artigo e licitação/compra

Para se perceber de uma forma rápida e fácil quais os estados de um artigo e de uma licitação/compra no *marketplace* e as ações que os despertam, desenhou-se dois diagramas de estados (artigo na Figura 6-2 e licitação/compra na Figura 6-3). Um artigo é algo que pode ser vendido por um utilizador no *marketplace*, enquanto uma licitação é uma proposta de compra sobre o mesmo, que ao ser aceite por um vendedor de um artigo passa a compra.

Como se pode observar nos diagramas, todos os estados contém um nome e um número, onde apenas o número é salvo na base de dados. Depois a associação entre esses dois elementos é feita através de ficheiro *helper* da aplicação em que contém um *hash*, onde associa o nome ao número, e ao utilizar os estados no código é usado o nome através desse *hash*. Um *hash* em Ruby é uma generalização de um *array* que em vez de apenas permitir índices, como um *array*, é permitido que qualquer objeto seja utilizado como um "índice". Esta forma foi utilizada a pensar numa possível escalabilidade do *marketplace*, ou seja se no futuro adicionar mais estados, ser mais fácil a alteração das associações entre número e nome pois estas estão todas centralizadas num único ficheiro.

Os atores que podem despertar alterações de estado através de ações num artigo são:

- Vendedor – é o dono do artigo;
- Potencial comprador – é um utilizador que pode fazer uma licitação ao artigo;
- Comprador selecionado – é o dono da licitação que o vendedor aceitou;
- Sistema – é a plataforma Collectors Bridge.

Os estados possíveis num artigo são:

- canceled – significa que o artigo foi cancelado pelo vendedor e já não se encontra disponível para venda e visível ao público;
- unpublished – significa que o artigo encontra-se criado, mas ainda não foi publicado para venda no *marketplace*;
- waiting_bids – significa que o artigo se encontra publicado no *marketplace* mas ainda não possui licitações;

- with_offers – significa que o artigo possui pelo menos uma licitação;
- accepted – significa artigo possui uma licitação que foi aceite pelo vendedor e aguarda resposta do comprador selecionado;
- paid_pending – significa que o pagamento foi executado pelo comprador selecionado, mas ainda encontra-se pendente ou seja, o dinheiro ainda não chegou ao vendedor;
- paid_completed – significa que o artigo foi pago pelo comprador selecionado e o dinheiro já se encontra no vendedor;
- sent – significa que o artigo foi enviado pelo vendedor;
- received – significa que o artigo já chegou ao comprador selecionado;
- completed – significa que o processo de compra/venda foi concluído;
- failed – significa que a compra venda falhou por alguma razão.

O artigo apenas é público quando está num estado de publicado e sem nenhum pagamento o processo de compra concluído.

Em relação a uma licitação/compra os atores podem despertar alterações de estados através de ações são:

- Comprador – dono da licitação/compra;
- Vendedor – dono do artigo em que a licitação é feita;
- Sistema – plataforma Collectors Bridge.

A nível de estados numa licitação/compra, apenas difere de um artigo quando esta está pendente. Quando é aceite pelo comprador esta passa a compra e possui os mesmos estados do artigo, diferenciando-se apenas caso o pagamento falhe. Os estados diferentes do artigo são:

- canceled – significa que a licitação foi aceite pelo vendedor, mas rejeitada posteriormente pelo comprador;
- refused – significa que a licitação foi recusada pelo vendedor;
- expired – significa que a licitação foi expirada pelo sistema;
- waiting_reply – significa que a licitação foi feita e encontra-se à espera de resposta por parte do vendedor;

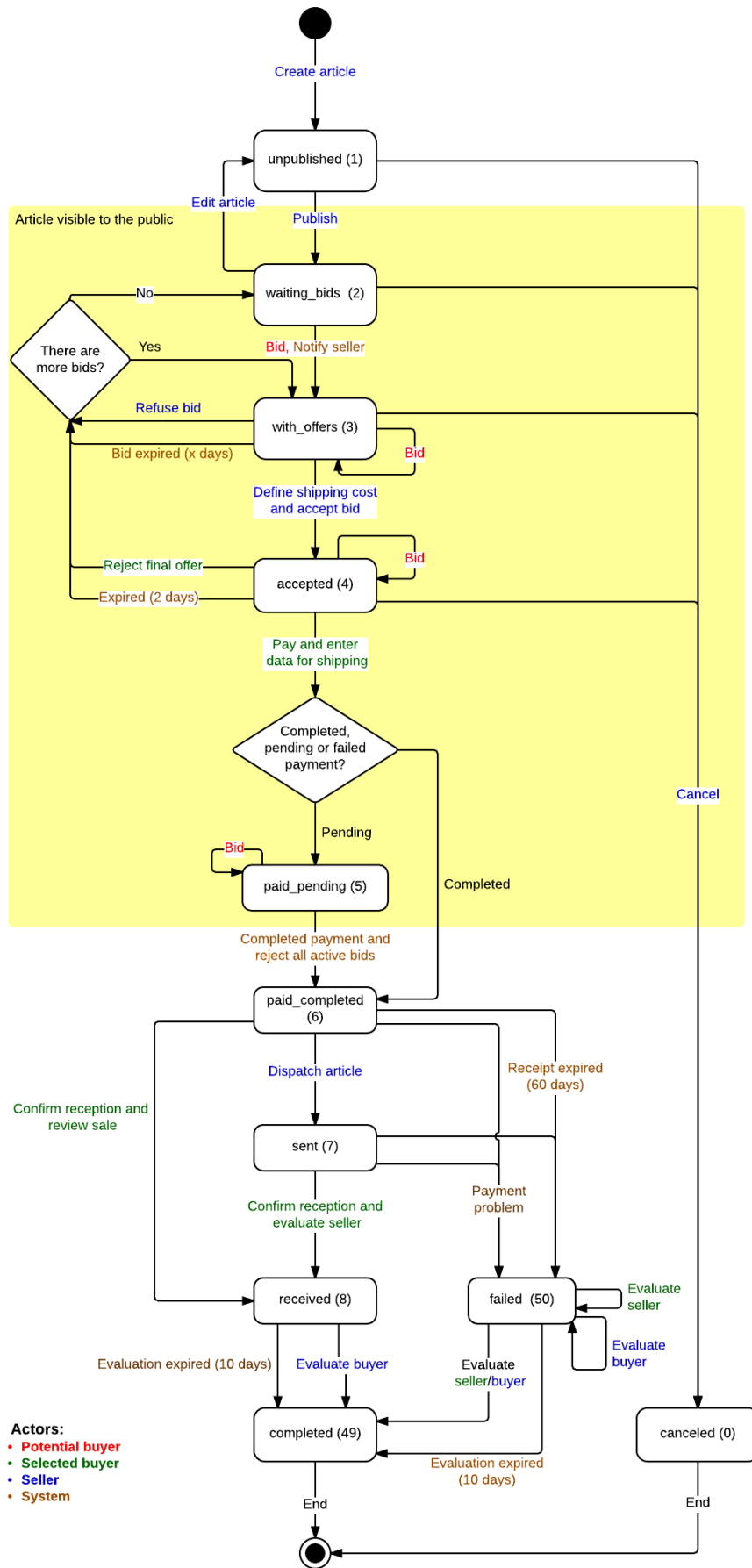


Figura 6-2 - Diagrama de estados de um artigo

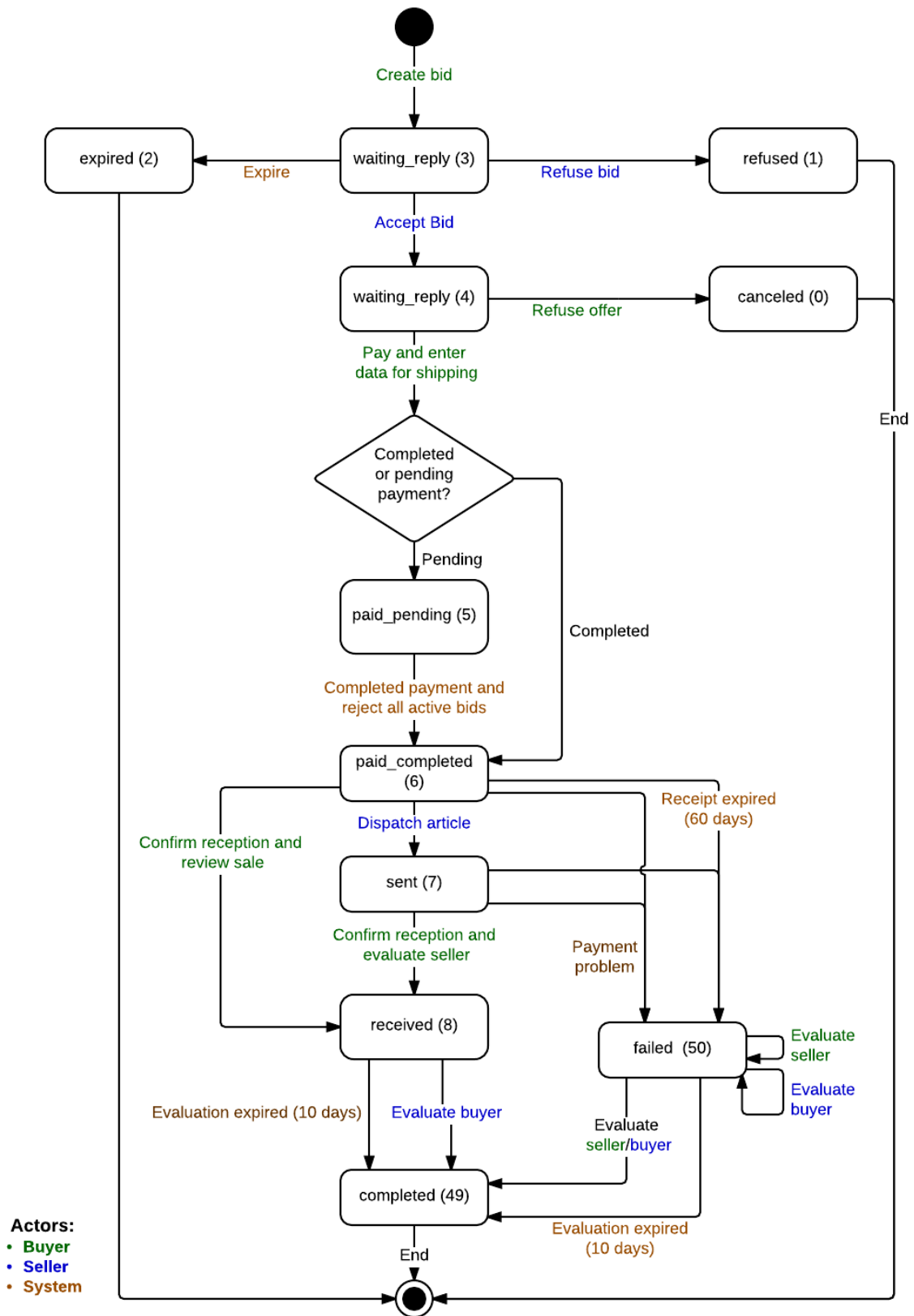


Figura 6-3 - Diagrama de estados de uma licitação/compra

6.3.2. Modelos e controladores

A Figura 6-4 mostra o diagrama de modelos do *marketplace*, estando descritos de forma detalhada no Anexo C deste relatório. Como se pode visualizar foram utilizados apenas 2 modelos já existentes na plataforma e adicionados 13 novos, sendo que os modelos de utilização exclusiva no *marketplace* estão identificados com “Market” antes do nome para facilitar a identificação dos mesmos. Nos modelos já existentes apenas estão representados os novos atributos adicionados. Em alguns casos existe redundância nas relações entre modelos com o objetivo de aumentar o desempenho do sistema. Um exemplo encontra-se no modelo MarketPurchase em que este se relaciona diretamente com o utilizador vendedor e comprador. No entanto esta informação poderia ser obtida através dos modelos MarketBid e MarketArticle, mas como a pontuação dos utilizadores (detalhada na próxima secção) é utilizada muito frequentemente e o seu calculo ser realizado através do modelo MarketPurchase, uma relação direta para o utilizador faz com que desempenho seja aumentado, pois se essa relação não existisse implicaria uma maior complexidade da pesquisa, afetando assim o desempenho do sistema.

Os modelos presentes no diagrama são:

- User – Representa um utilizador da plataforma;
- Category – Representa uma categoria da plataforma;
- MarketHistoryBid – Representa um dado histórico de uma licitação;
- MarketHistoryArticle – Representa um dado histórico de um artigo;
- MarketArticleImage – Representa uma imagem de um artigo;
- MarketPurchase – Representa uma compra/venda;
- MarketBid – Representa uma licitação;
- MarketArticle – Representa um artigo do *marketplace*;
- MarketCurrency – Representa um tipo de moeda de possível utilização no *marketplace*;
- MarketShipping – Representa um custo de envio de um artigo para uma zona;
- MarketZones – Representa uma zona do planeta (por exemplo um continente ou o país que se encontra o artigo a ser vendido);
- MarketQuestion – Representa uma questão a um artigo no *marketplace*;
- MarketWatchlis – Representa um artigo em *watch*;

- Country – Representa um país, sendo este utilizado apenas no *marketplace* e de possível utilização na restante plataforma;
- Region – Representa uma subdivisão de um país, sendo este utilizado apenas no *marketplace* e de possível utilização na restante plataforma;

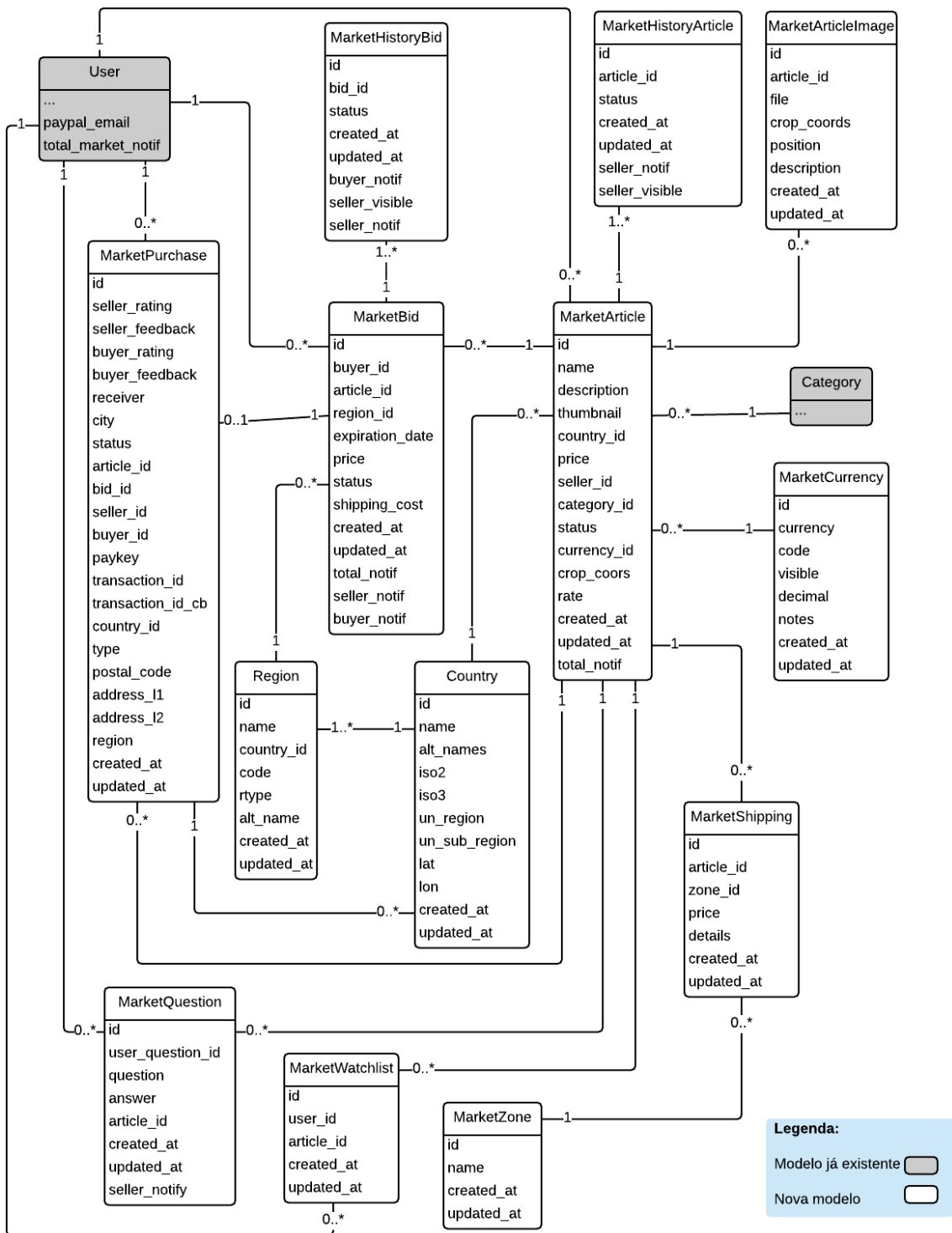


Figura 6-4 - Diagrama dos modelos do marketplace

Em relação aos controladores do *marketplace*, apenas foram adicionados dois em que um controla todas as ações efetuadas dentro do mesmo e outro controla as notificações dos pagamentos, recebidas atualmente pelo PayPal. O controlador MarketplaceController herda o BaseController, no qual este é responsável realizar algumas ações antes, como por exemplo verificar se o utilizador encontra-se autenticado, entre outras. A Figura 6-5 mostra o diagrama de controladores.



Figura 6-5 – Diagrama de controladores do *marketplace*

6.4. Implementação

Esta secção tem como objetivo apresentar a forma como foram implementadas as diversas funcionalidades que constituem o *marketplace*.

6.4.1. Navegação no *marketplace*

A navegação no *marketplace* assemelha-se à restante plataforma, sendo acedida pelo menu lateral, contendo uma página para listagem de artigos (exemplo na Figura 6-6) e outra para detalhes (exemplo na Figura 6-7). Na listagem de artigos foi reaproveitado das outras páginas, o código que apresenta os artigos em formato de cartões, mantendo o botão de partilha nas redes sociais (funcionalidade também reaproveitada) e adicionado outro botão para *watchlist*, no qual este permite colocar ou retirar um artigo da *watchlist*. Estas duas últimas ações são feitas assincronamente por Ajax. Para a paginação é utilizado a gem `will_paginate`²⁴ (também utilizada nas restantes páginas) e a opção de escolher quantas linhas de artigos a apresentar foi também reaproveitada.

²⁴ https://github.com/mislav/will_paginate

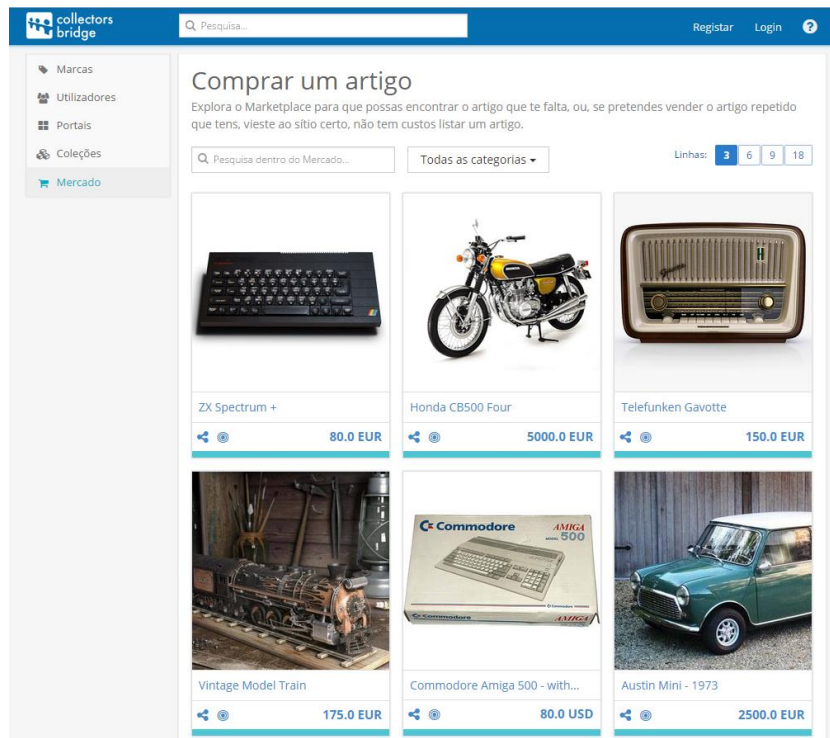


Figura 6-6 – Screenshot da listagem de artigos no marketplace

A página de um artigo no marketplace é semelhante à página de um artigo de uma coleção, ou seja é apresentada em blocos de informação (por exemplo descrição, galeria fotográfica, entre outros). Nela também é permitido um utilizador registado colocar questões sobre o artigo ao vendedor. Para haver anonimato entre os utilizadores a identificação é feito com a primeira e ultima letra dos mesmos e não é passado nenhum ID para o cliente.

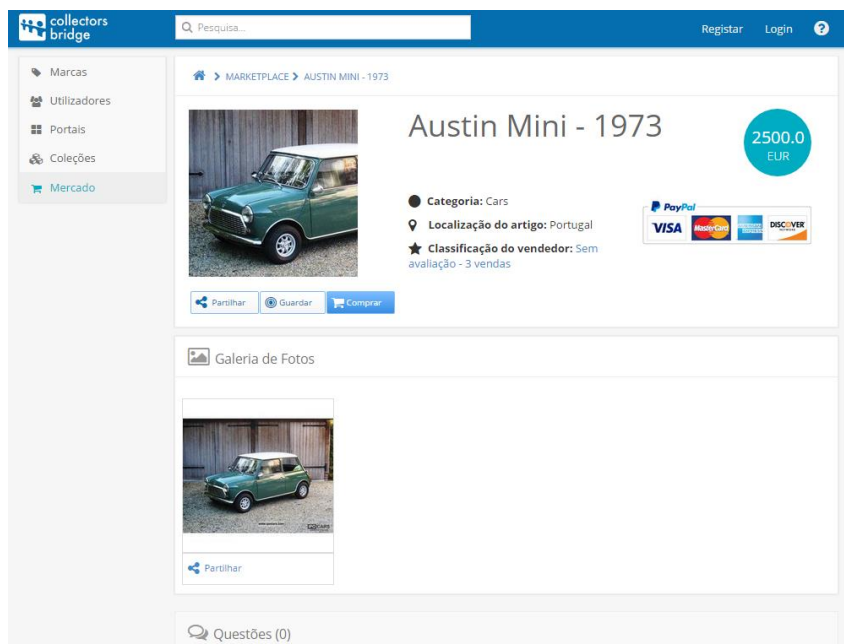


Figura 6-7 – Screenshot da página de um artigo no marketplace

6.4.2. Inserção, publicação, edição e cancelamento de artigos

Inserção e publicação

Existem duas formas de inserir artigos no *marketplace*, sendo pela sua página inicial ou copiando um item de uma coleção do utilizador. Em qualquer uma destas formas é apenas criado um artigo com um nome, descrição e imagem, envolvendo de seguida um segundo passo que permite ao utilizador colocar mais informações sobre o artigo, como imagens, preço, entre outros. Apenas é permitido a publicação dos artigos com todos os campos obrigatórios preenchidos.

Para a inserção de imagens no *marketplace* (galeria ou imagem de capa) foi utilizado um *uploader* já existente na plataforma, no qual este utiliza a gem *CarrierWave*²⁵ para carregamento entre cliente-servidor, e a gem *MiniMagick*²⁶ para recriar 5 versões, em diferentes dimensões da imagem, mantendo assim a coerência em toda a plataforma. As versões das imagens são:

- Imagem original - não sofre qualquer alteração, apenas serve para serem criadas as versões que são usadas na plataforma;
- Croppable - transformação da imagem para que seja possível efetuar sempre o *crop* corretamente;
- Original_cropped - imagem original com o *crop*;
- Base - imagem já com o *crop* feito mas em dimensão mais pequena (limite máximo de 1920x1080), disponível a visualização no *viewer*;
- Smallest - versão minúscula para futuras implementações, tamanho fixo 63x63;
- Thumbnail - versão utilizada nos *cards*, tamanho fixo 240x240.

Para o utilizador poder fazer *crop* das imagens depois de inseridas, foi utilizado o jQuery Image Cropping Plugin (Jcrop)²⁷, no qual este também é utilizado nas restantes partes da plataforma.

Com a finalidade de ter controlo e coerência dos dados geográficos e tipos moedas no *marketplace*, foi decidido adquiri-los e tê-los previamente salvos na base de dados para depois o utilizador poder selecciona-los em vez de os escrever livremente. Os dados do tipo de moeda foram obtidos no *website* do PayPal²⁸ e como o número de casas suportados por cada tipo não

²⁵ <https://github.com/carrierwaveuploader/carrierwave>

²⁶ <https://github.com/minimagick/minimagick>

²⁷ <http://deepliquid.com/content/Jcrop.html>

²⁸ https://developer.paypal.com/docs/classic/api/currency_codes/#paypal

é igual, foi criado um atributo com essa indicação, sendo os valores dos preços posteriormente apresentados ao cliente de acordo com o número de casas decimais. Os países e as suas respetivas subdivisões foram obtidos num *website*²⁹, estando sob a norma ISO 3166. Estes dados foram salvos em ficheiros de texto e para automatizar a população na base de dados, foi criado um script que lê os ficheiros e insere na base de forma automática.

Para inserção dos custos de envio pelo vendedor havia algumas hipóteses, sendo de forma manual onde o vendedor coloca o valor do custo de envio para cada proposta que recebe, de uma forma mais genérica em que o vendedor configura um preço para determinadas zonas ou utilizando uma API que automatiza o processo de decisão do valor. A utilização de uma API seria a solução ideal para este problema e analisou-se algumas, mas como eram pagas ou tinham alguma restrição, não se optou por esta solução. Selecionou-se a hipótese de um preço de custo de envio por licitação. Não se optou pela hipótese de colocação de um preço fixo por zona, pois em alguns casos os preços da mesma zona podem mudar (por exemplo Portugal continental e Açores ou Europa e Reino Unido). No entanto esta hipótese é utilizada de uma forma parcial (apenas indicativa e opcional), sendo possível ao vendedor dizer o preço para determinadas zonas do globo, com um campo de texto livre, em que este pode indicar alguma nota ou restrição (por exemplo vende com o preço X para Portugal continental e com o preço Y para os Açores).

Edição e cancelamento

Como a Collectors Bridge não cobra aos utilizadores para inserir artigos, decidiu-se que os mesmos podem permanecer listados por tempo indeterminado no *marketplace*, sendo possível uma edição dos seus dados apenas se estes não tiverem qualquer licitação pendente. Quando entram no modo de edição ficam no estado de não publicado, ficando assim não visíveis ao público.

Por razões de segurança, empresa decidiu que não seria possível ao utilizador eliminar de forma permanente artigos do *marketplace*, no entanto é possível cancelar desde que não existem pagamentos a decorrer sobre os mesmos, ficando assim não visíveis ao público. Se existir licitações ativas ao cancelar um artigo, estas são rejeitadas automaticamente.

6.4.3. Compras e vendas

²⁹ <https://gist.github.com/mindplay-dk/4755200>

O painel de controlo de compra e vendas de um utilizador está disponível na página inicial do *marketplace* (Figura 6-8). Este painel é uma *partial view* do Ruby on Rails que é renderizado se tiver um utilizador autenticado, podendo assim ser reaproveitada em outros futuros locais da plataforma.



Figura 6-8 - Painel de compra e vendas

A informação de uma compra ou venda de um artigo é disponível na página do mesmo, através da adição de um painel antes da restante informação (Figura 6-9). À semelhança do painel compra/vendas anterior, estes também são *partial view* do Ruby on Rails, sendo carregado o respetivo painel (vendedor ou comprador) ao renderizar a página de um artigo. A informação apresentada no painel altera com o estado da compra/venda.

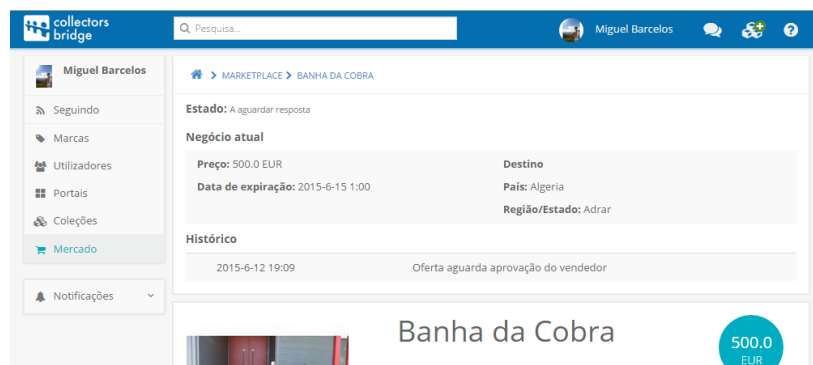


Figura 6-9 – Painel de uma compra

6.4.4. Pagamentos

Um dos objetivos no *marketplace* era os pagamentos serem feitos através do PayPal, no qual a Collectors Bridge deveria ficar com uma pequena percentagem por cada comprar/venda realizada.

Depois de uma análise à documentação do PayPal e às suas diferentes API's para integração [36], selecionou-se a API Adaptive Payments [37] [38], pois esta permite num único pagamento, transacionar dinheiro para diferentes destinatários. No entanto dentro desta API existem diversos modos para obter esse fim e para ajudar a empresa a decidir qual o modo a

utilizar nos pagamentos do *marketplace*, fez-se uma análise, resumindo as diferentes opções que poderiam interessar. Esta análise encontra-se disponível no Anexo F deste relatório. O modo selecionado para integração foi Chained Payments e funciona da forma como está ilustrado na Figura 6-10 ou seja, quando o utilizador paga um artigo ao vendedor, o PayPal retira automaticamente a comissão definida (2.5%) e envia para a Collectors Bridge. O vendedor também paga as taxas do PayPal.



Figura 6-10 - PayPal Chained Payments

Para se integrar no código da plataforma, foi utilizado gem PayPal Adaptive Payments SDK³⁰ que providencia uma implementação da API Adaptive Payments na linguagem Ruby.

Os pedidos de pagamento são criados no PayPal quando uma licitação é aceite pelo vendedor. A Figura 6-11 ilustra o processo de comunicação entre a plataforma e PayPal na criação de um pedido de pagamento bem-sucedido. Na primeira comunicação é utilizado o método `Pay`³¹, no qual a plataforma envia um pedido para criação de um pagamento, passando os dados para o mesmo ou seja, os *emails* das contas de PayPal da Collectors Bridge e vendedor com os respetivos montantes, entre outras configurações necessárias. Caso esteja tudo correto o PayPal manda uma mensagem com uma chave do pagamento e mais alguns parâmetros. Ao receber esta mensagem bem-sucedida, a plataforma envia outra mensagem utilizando o método `SetPaymentOptions`³², onde é requerido a obrigatoriedade do comprador colocar a sua morada para o envio do artigo e adicionado alguma informação sobre o produto a ser vendido, como por exemplo o nome, ID, descrição, entre outros. Depois o PayPal responde que foi bem-sucedido. Caso o PayPal responda numa dessas mensagens com insucesso, a plataforma guarda

³⁰ <https://github.com/paypal/adaptivepayments-sdk-ruby>

³¹ https://developer.paypal.com/docs/classic/api/adaptive-payments/Pay_API_Operation/

³² https://developer.paypal.com/webapps/developer/docs/classic/api/adaptive-payments/SetPaymentOptions_API_Operation/

em *log* essa mensagem, para caso seja necessário investigar no futuro. Alguns dos erros que podem ocorrer que contribuem para este tipo de mensagens, são por exemplo o vendedor colocar um *email* inválido ou a sua venda está a ser realizada num tipo de moeda que não é aceite pela sua conta do PayPal.

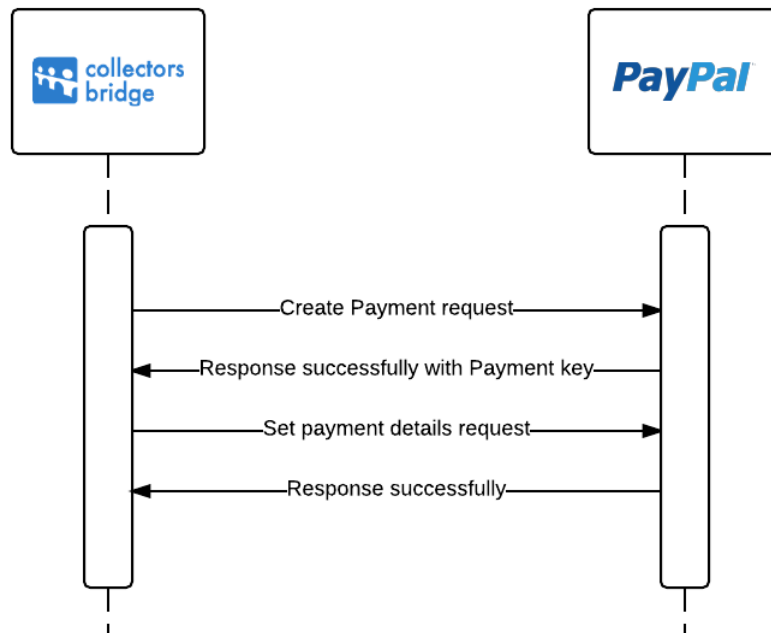


Figura 6-11 - Diagrama de sequência da criação de um pedido de pagamento no PayPal

Depois do pagamento ser criado no PayPal é atualizado a base de dados e alterados os estados do artigo e compra, ficando assim disponível para o comprador efetuar o pagamento. Para se poder efetuar pagamentos, foi incluído um botão de checkout no qual este abre uma mini janela no *browser* com a página de *login* do PayPal. Ao efetuar o login com uma conta do PayPal, é pedido que se selecione uma morada para receber o artigo e pagar. Terminado o pagamento, a janela é fechada e a página que estava antes na plataforma é atualizada podendo estar já atualizada com os dados do pagamento. A Figura 6-12 ilustra esta sequência. Quando o pagamento é efetuado o PayPal envia um *email* para intervenientes e uma mensagem IPN (*Instant Payment Notification*) [39] [40] para a plataforma dizendo o estado do pagamento e outras informações. Esta mensagem normalmente é enviada antes do comprador voltar à plataforma, no entanto poderá haver uns segundos de atraso. Para evitar a possibilidade da mensagem por parte do PayPal atrasar-se e o comprador cancelar a compra do artigo quando este se encontra pago, a opção de cancelamento da compra é bloqueada por 6 minutos, depois do comprador abrir a mini janela de pagamento no PayPal.

Cada chave de pagamento do PayPal tem uma validade de 3 horas para se poder efetuar o pagamento. Como um dos requisitos do *marketplace* é a possibilidade de um comprador ao ter a sua oferta aceite, poder pagar um artigo num máximo de 2 dias, esta opção de pagamento deveria se manter por mais tempo e não 3 horas. Para ultrapassar este problema utilizou-se a gem Clockwork³³ que executa tarefas agendadas e que neste contexto é verificado de minuto a minuto se tem alguma chave de pagamento expirada. Se sim, e se compra estiver no período permitido pela Collectors Bride para realizar o pagamento, o sistema volta a fazer um pedido de uma nova chave ao PayPal, permitindo assim que o pagamento fique disponível por mais 3 horas. Esta gem já era utilizada pela plataforma para executar diversas tarefas como por exemplo agendar *emails* para os utilizadores.

A Figura 6-13 ilustra a troca de mensagens IPN e no recebimento de uma notificação de pagamento concluído e obtenção do endereço do recetor, entre o PayPal e a plataforma. Para isto funcionar foi implementado um controlador apenas para estar à escuta destas mensagens, a serem recebidas por POST. Quando uma mensagem é recebida, é respondido com o código HTTP 200 dizendo que a mesma foi recebida corretamente. Se o PayPal e não receber resposta ou receber outro código, ele tentará reenviar mais 15 vezes. Depois de recebida a mensagem a plataforma reenvia a mesma para um endereço do PayPal no qual este verifica a assinatura e responde se é válida ou não.

As mensagens IPN são enviadas sempre que há uma atualização nos pagamentos, por exemplo se o pagamento falhou, está pendente, concluído, entre outros. No caso de receber uma notificação a dizer que o pagamento falhou, o sistema volta a pedir uma nova chave de pagamento pelo método descrito anteriormente, pois a chave em que o pagamento falhou não é mais válida pelo PayPal. No caso de receber uma notificação dizendo que o pagamento foi concluído, a plataforma pede através do método `GetShippingAddresses`³⁴ o endereço de destino e atualiza na base de dados. Se receber uma mensagem a dizer que o pagamento encontra-se pendente, apenas é atualizado os estados do artigo e da compra na base de dados. Por fim se receber uma mensagem a dizer que o dinheiro foi devolvido ao comprador, estando a compra/venda num estado de pagamento completo, é atualizado o estado na base de dados para falhado.

³³ <https://github.com/tomykaira/clockwork>

³⁴ https://developer.paypal.com/docs/classic/api/adaptive-payments/GetShippingAddresses_API_Operation/

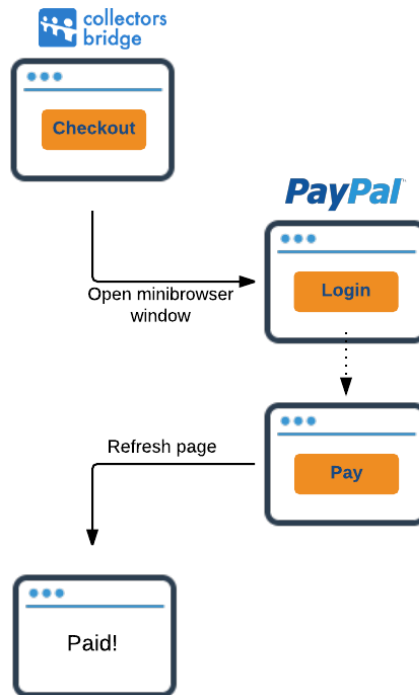


Figura 6-12 – Sequencia de um pagamento bem-sucedido executado por um comprador na C’B

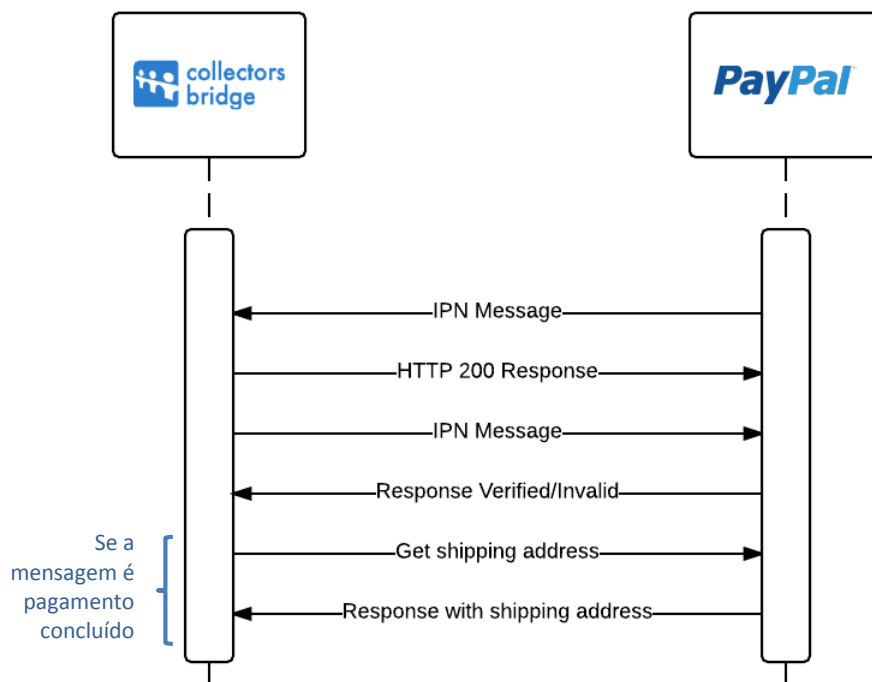


Figura 6-13 – Diagrama de sequência da troca de mensagens IPN e obtenção do *shipping address* entre a C’B e PayPal

As mensagens IPN enviadas ou por enviar encontram-se disponíveis também na área de utilizador da Collectors Bridge, no *website* do PayPal. Todas as mensagens IPN recebidas pelo PayPal são registadas em *log* e caso o sistema receba uma mensagem IPN que não tenha nenhuma compra/venda associada ou simplesmente ocorrer algum erro inesperado, é enviado um *email* para a Collectors Bridge a relatar o sucedido para rapidamente ser resolvido.

6.4.5. Proteção do vendedor e comprador

Uma das preocupações para os utilizadores ao utilizarem *marketplaces online* para vender ou comprar artigos são as fraudes e a proteção contra as mesmas. Por exemplo, se um utilizador paga um artigo ao efetuar uma compra num *marketplace* e depois o artigo chegar diferente do que era anunciado ou simplesmente não chegar? Para proteger o *marketplace* da Collectors Bridge contra esse tipo de situações foi utilizado a proteção de vendedores e compradores do PayPal [39]. Esta proteção cobre todos os artigos físicos que possam ser enviados por correio e não sejam proibidos por lei. Para integrar esta solução à plataforma é necessário ter uma morada de envio no PayPal, por isso foi necessário obrigar o comprador colocar a morada de receção do artigo no ato de pagamento, na janela do PayPal.

Para o comprador usufruir desta proteção, deverá iniciar a sessão na sua conta PayPal e abrir uma disputa no prazo máximo de 180 dias após o seu pagamento para dar início a um processo de proteção do comprador. Se não receber uma resposta do vendedor ou não conseguir chegar a uma solução, deverá escalar a disputa para uma reclamação no prazo máximo de 20 dias após a abertura da disputa original. Desta forma, o PayPal irá investigar a reclamação e tomar uma decisão sobre a sua elegibilidade de acordo com o Contrato do Utilizador. Caso ganhe a disputa o PayPal devolverá o montante da compra na sua totalidade.

No caso do vendedor se for alvo de uma reclamação por parte do comprador, este deverá enviar para o PayPal uma prova digital ou física de entrega da empresa de transporte, que deve incluir a data de envio, a morada do comprador, a data de entrega e também poderá ser pedido mais algumas informações por parte do PayPal.

6.4.6. *Feedback* e classificações

Sistemas de *feedback* e classificação de utilizadores são funcionalidades muito comuns em diversos *marketplaces online*, no qual podem transmitir aos outros utilizadores se um utilizador é confiável ou não. No caso da Collectors Bridge foi implementado esse sistema tanto para os vendedores como os compradores, ou seja, cada utilizador que utilize o *marketplace* tem duas classificações diferentes (uma como vendedor e outra como comprador). A classificação como vendedor é calculada pelas avaliações dos compradores que compraram os seus artigos, no qual estes disseram no fim de cada compra, se a mesma correu bem (positiva) ou mal (negativa), e a do comprador é calculada da mesma forma mas pelas avaliações feitas pelos vendedores no fim das suas vendas. A Figura 6-14 mostra como é

calculado a classificação, numa escala de 0 a 5, sendo apenas atribuída caso o número total avaliações seja inferior a 5, evitando assim influências drásticas nas primeiras avaliações. No fim de cada processo de compra/venda também existe a opção de dar feedback entre os dois utilizadores.

$$Rating = \frac{p}{t} \times 5, \quad t > 5$$

$$total (t) = positivos (p) + negativos (n)$$

Figura 6-14 – Formula das classificações de utilizadores no *marketplace*

6.4.7. Expirações

Com o objetivo de licitações e processos de compra/venda não ficarem eternamente com um estado pendente no sistema, implementou-se um sistema que os expira de forma automática. Para tal utilizou-se a gem Clockwork, já utilizada nos pagamentos descritos anteriormente. Neste caso concreto esta gem está a ser utilizada para correr de hora a hora um método que verifica quais as licitações ou compras que a data é inferior à atual do sistema, e as que forem inferiores são expiradas. Como foi mostrado na Figura 6-2 da secção anterior, existem quatro tipos de expiração:

- Expiração da licitação – esta expiração acontece quando uma licitação se encontra à espera de resposta por parte de vendedor e a data de expiração é definida pelo autor da licitação, sendo restrita entre 2 a 30 dias futuros;
- Expiração de uma licitação aceite pelo vendedor – esta expiração acontece quando o artigo foi aceite pelo vendedor e não obteve resposta por parte do autor da licitação. O número de dias até expirar são 2 a partir do momento em que a licitação foi aceite;
- Expiração de um artigo pago – este tipo de expiração ocorre quando um artigo é pago ou enviado, e não confirmada a receção por parte do comprador. O número de dias até ser expirado são 60 a partir do dia em que o artigo foi pago ou enviado.
- Expiração da avaliação e feedback – esta expiração ocorre quando um artigo foi recebido ou quando uma compra/venda falhou, não havendo uma avaliação e feedback dos intervenientes. O número de dias até expirar são 10.

6.4.8. Notificações

Para avisar os utilizadores do *marketplace* que ocorreram mudanças de estado (não executadas pelos próprios) ou houve alguma pergunta/resposta, nas suas compras ou vendas, foram implementadas notificações. Estas notificações são de duas formas: por *email* e dentro da plataforma.

Nas notificações por *email* é utilizado a gem Delayed Job ³⁵. Esta gem é responsável por executar tarefas de forma assíncrona de longa duração.

As notificações dentro da plataforma aparecem em 3 sítios diferentes:

- Menu lateral da plataforma junto à opção “Marketplace” – é apresentado o número total de todas as notificações do *marketplace*;
- Painéis de vendedor e comprador – é indicado por cada item dos painéis o número total de notificações sobre o item;
- Página do artigo – é apresentado uma indicação nos sítios respetivos da página do artigo. Por exemplo se tem uma nova licitação, pergunta, ou houve algum avanço no negócio corrente.

Sempre que o utilizador entra na página do artigo em que possui notificações, estas são decrementadas do número total de notificações do *marketplace*. Não se juntou este tipo de notificações com as restantes notificações sociais da plataforma, pois a empresa considera estas mais prioritárias, merecendo assim separação das restantes.

6.4.9. Estrutura e organização do código

Ícones

Todos os ícones utilizados no *marketplace*, assim como nas restantes partes da plataforma são do *website* Font Awesome³⁶, no qual este disponibiliza centenas de ícones sob licença MIT³⁷ e um CDN para o carregamento rápido dos mesmos.

Idiomas e fusos horários

Com o objetivo da globalização da Collectors Bridge, todas as *strings* utilizadas na plataforma são armazenadas num ficheiro (um por idioma) no formato chave-valor.

³⁵ https://github.com/collectiveidea/delayed_job

³⁶ <http://fontawesome.github.io/Font-Awesome/>

³⁷ <http://opensource.org/licenses/mit-license.html>

Atualmente a plataforma está disponível nos idiomas inglês e português. A escolha do idioma a ser carregado é feita automaticamente nos pedidos HTTP, no qual a plataforma verifica que idioma está contido no cabeçalho do pedido e se for PT o ficheiro a ser carregado é o português, se não é o de inglês.

Para que se mostre a hora local do utilizador na plataforma, todas as horas e datas armazenadas na base de dados contêm o GMT do servidor, sendo que quando apresentadas ao cliente, o jQuery transforma para a hora do sistema do utilizador, com base no GMT do servidor e cliente.

Validações

As validações dos *inputs* dos utilizadores para o sistema são sempre feitos em dois lados:

- Servidor – Este verifica se o utilizador tem autorização para efetuar o que pretende e se sim, verifica se os dados que introduzido obedecem às regras previamente estabelecidas, como por exemplo, se um número é positivo, se é um campo obrigatório, entre outras;
- Cliente – Estas validações são executadas por JavaScript, utilizando o plugin jQuery Form Validator³⁸, que verifica se todos os campos introduzidos obedecem às regras, antes de ser submetido para o servidor.

Anatomia da aplicação

O código implementado no módulo *marketplace* segue estrutura padrão da *framework* Rails, estando coerente com o restante código da plataforma. A figura seguinte ilustra apenas os principais ficheiros e diretórios adicionados ou alterados durante o desenvolvimento do *marketplace*.

³⁸ <http://formvalidator.net/>



Figura 6-15 – Anatomia do *marketplace*

Como se pode observar foram adicionados e alterados vários ficheiros/diretórios:

- 5 | jcrop_controllers – Neste diretório é possível encontrar os dois *croppers* de imagens utilizados no *marketplace*;
- 6 | marketplace.js.erb – Este ficheiro contém todo o código JavaScript utilizado no *marketplace*;
- 8 | marketplace.scss – Este ficheiro contém todos os estilos utilizados exclusivamente no *marketplace*;
- 9 | controllers – Neste diretório é possível encontrar os dois controladores descritos no desenho do sistema do *marketplace*;

- 10 | models – Neste diretório é possível encontrar todos os modelos criados para o *marketplace*;
- 12 | mailer.rb – Neste ficheiro foram adicionados os métodos para envio de emails do *marketplace*;
- 14 | marketplace_helper.rb – Este ficheiro contém funções mais genéricas de possível utilização em diferentes zonas do código do *marketplace*;
- 15 | uploaders – Neste diretório estão contidos os dois *uploaders* utilizados para o carregamento de imagens;
- 17 | marketplace – Neste diretório estão armazenadas todas as *views* criadas para o *marketplace*;
- 18 | mailer – Este diretório contém os *templates* de *emails* a ser enviados pelo *marketplace*;
- 20 | locales – Neste diretório é possível encontrar os ficheiros de idiomas (EN e PT) utilizados em toda a plataforma e no *marketplace*;
- 21 | application.rb - Este ficheiro contém configurações globais do *marketplace*, como por exemplo a percentagem da comissão da Collectors Bridge nas compras/vendas, número de dias das expirações, entre outras;
- 22 | paypal.yml – Este ficheiro contém a configuração utilizada pela API do PayPal;
- 23 | routes.rb – Este ficheiro contém as rotas de toda a plataforma;
- 25 | migrations – Neste diretório foram adicionados as migrações feitas à base de dados com a inclusão do *marketplace* na plataforma;
- 26 | datasets – Este diretório contém ficheiros com dados para ser populados na base de dados;
- 27 | seeds.rb – Este ficheiro contém um *script* responsável por popular a base de dados;
- 29 | paypal_interface.rb – Este ficheiro contém um módulo responsável por utilizar os métodos da API do PayPal;
- 31 | clockwork.rb – Este ficheiro contém os métodos executados de forma agendada (por exemplo as expirações de licitações);
- 32 | spec – Este diretório contém todos os testes automatizados.

6.5. Testes

Os testes unitários e de regressão implementados para este capítulo encontram-se no Anexo D. Passaram 53 dos 53 testes unitários e 65 dos 65 dos testes de regressão. Nas

funcionalidades que não foram possíveis de testar de forma automática fez-se testes de forma manual, estando descritos no Anexo I. Passaram 32 dos 32 testes manuais.

A compatibilidade das funcionalidades também foi testada nos 4 *browsers* descritos nos requisitos não funcionais, sendo que não foi encontrada qualquer incompatibilidade.

A nível de velocidade de carregamento foi medido o tempo de carregamento da página inicial do *marketplace*, com um utilizador autenticado, havendo 9 artigos listados, 10 artigos vendidos e 10 comprados no seu *dashboard*, e também a página de detalhes de um artigo listado, contendo uma foto na galeria. Para ter algo de referência para se poder comparar, mediu-se também a velocidade da página inicial das coleções com um utilizador autenticado, havendo 9 artigos listados, e a página de detalhes de um item de uma coleção, tendo uma foto na galeria. As medições foram realizadas ao servidor de pré-produção da Collectors Bridge que se encontra em Portugal e no lado do cliente (também localizado em Portugal) foi utilizado o *browser* Google Chrome v43, numa largura de banda disponível com 12.4Mbps de *download* e 0.7Mbps de *upload*. As tabelas seguintes mostram os resultados obtidos, sendo que cada resultado apresentado é uma média de 5 medições.

		coleções	marketplace
Velocidade em segundos sem cache	Média	4.79s	4.41s
	Desvio padrão	0.24s	0.29s
Velocidade em segundos com cache	Média	3.99s	3.66s
	Desvio padrão	0.51s	0.28s

Tabela 6-1 - Testes da velocidade de carregamento da página inicial das coleções e *marketplace* com e sem *cache*

		coleções	marketplace
Velocidade em segundos sem cache	Média	5.03s	4.74s
	Desvio padrão	0.29s	0.26s
Velocidade em segundos com cache	Média	4.30s	3.77s
	Desvio padrão	0.25s	0.47s

Tabela 6-2 - Testes da velocidade de carregamento da página de detalhe de um item das coleções e *marketplace* com e sem *cache*

Como se pode ver a velocidade de carregamento das páginas do *marketplace* está superior ao desejado e isto acontece devido a uma série de fatores, em que a resolução encontra-se fora do âmbito do estágio. Alguns desses fatores e sugestões para a correção e melhoramento são:

- O *layout* geral da plataforma, que é carregado sempre em cada pedido contém um excesso de informação não utilizada como por exemplo o tutorial inicial de utilização da Collectors Bridge. Este tutorial que se encontra de forma invisível, sendo apenas mostrado ao carregar num botão, ocupa quase metade do HTML carregado. Uma

melhoria poderia ser a colocação desse tutorial a ser carregado de forma assíncrona e apenas quando solicitado;

- Existe algumas más práticas de implementação, presentes em muitas partes da plataforma (incluindo partes que são sempre carregadas independentemente da página), que também contribui para o aumento do tempo de carregamento. Um exemplo é a utilização de *queries* $N + 1$ o que se pode resolver muito facilmente com apenas 2 *queries* nessas situações;
- Excesso de código JavaScript é encontrado muito frequentemente nos ficheiros HTML (incluindo o layout geral) da plataforma, fazendo que sejam sempre carregadas com os pedidos HTTP. Uma melhoria seria colocar todo esse código em apenas um ficheiro JavaScript comprimido, fazendo com que esse ficasse em *cache*, não necessitando de carregamentos constantes;
- A utilização de vários serviços externos, como por exemplo o Google Analytics ou AddThis aumenta o número de pedidos HTTP, atrasando o carregamento das páginas;
- Outra melhoria que diminuiria bastante a velocidade de carregamento das páginas seria um carregamento parcial das mesmas, evitando que todo o *layout* e ficheiros associados fossem carregados. Uma forma fácil de implementação seria através da gem PJax³⁹ que atualmente é utilizada no *BackOffice*.

6.6. Lançamento em produção

O lançamento do conteúdo desenvolvido para produção envolveu uma série de tarefas. Uma delas foi a submissão da plataforma no PayPal, de modo a obter uma chave para se poder utilizar o serviço com dinheiro real. Além de toda a informação requerida sobre a empresa, esta tarefa envolveu uma descrição de tudo o que é utilizado pela API do PayPal, assim como a criação de contas de testes e pequenos tutoriais para os aprovadores do PayPal poderem testar. A plataforma foi aprovada passado poucos dias após a submissão.

Também foi realizado por parte de um elemento da empresa, uma revisão do texto inglês *marketplace*, assim como sua tradução para português.

Por fim houve uma série de testes exploratórios, realizada por diversos elementos da empresa, envolvendo duas fases. Na primeira foi utilizado o modo de dinheiro virtual do PayPal

³⁹ https://github.com/rails/pjax_rails

e na segunda o modo de dinheiro real. Em qualquer uma das fases foi encontrado pequenos erros, no qual foram corrigidos.

Atualmente o *marketplace* encontra-se a funcionar em produção e escondido ao público. O lançamento está previsto para o dia 10 de julho de 2015, e até lá a empresa está contactando colecionadores para colocarem colecionáveis à venda numa fase inicial.

CAPÍTULO 7

7. Conclusão

Neste último capítulo é feita uma conclusão e reflexão de todo o projeto realizado durante o estágio, sendo apresentado sugestões de trabalho futuro sobre as funcionalidades desenvolvidas.

7.1. Conclusão e reflexão

Este projeto de estágio, dividido em dois semestres, teve como principal objetivo a evolução da rede social Collectors Bridge, plataforma que pretende ser a referência mundial do colecionismo no mundo digital.

A primeira parte do estágio, decorrida no primeiro semestre teve como objetivos o planeamento, idealização e especificação do trabalho desenvolvido no segundo semestre. O uso da metodologia *Desing Thinking* na idealização foi uma mais-valia, pois ajudou a perceber todo domínio do problema, facilitando a definição e demonstração em protótipos não funcionais de novas melhorias e funcionalidades na plataforma. A utilização de protótipos não funcionais foi uma grande vantagem, pois além da empresa compreenderem melhor o que poderia ser feito, permitiu detetar e corrigir de forma rápida, algumas falhas de conceito, antes de entrar na implementação, poupando assim recursos futuros.

A segunda parte do estágio teve como objetivo a implementação de algumas melhorias no *BackOffice* da Collectors Bridge e a implementação de um módulo designado com *marketpace*, que permite aos utilizadores da plataforma transacionarem colecionáveis entre si e sem custos de listagem. Foi utilizado o serviço PayPal como método de pagamento, onde é cobrado uma taxa de 2.5% por cada compra/venda efetuada. Um dos maiores desafios, mas também motivação foram na linguagem de programação e *framework* utilizadas, visto que não se tinha qualquer conhecimento e experiencia nas mesmas.

Pode-se afirmar que o estágio foi bastante positivo, pois além de todo o conhecimento e experiencia adquiridas com o projeto teve-se a oportunidade de acompanhar de perto o processo inicial de uma *start-up* juntamente com o lançamento de uma nova plataforma no mercado. O módulo implementado será de grande importância, pois é uma fonte de receitas para a empresa e poderá ter grande impacto no futuro da mesma. A empresa planeou o seu lançamento ao público para o dia 10 de julho, juntamente com campanhas publicitárias por *newsletter*, Facebook e Google AdWords.

Por fim pode-se afirmar que todos os objetivos propostos foram cumpridos.

7.2. Trabalho futuro

Esta primeira versão do módulo *marketplace* está preparada para funcionar no estado inicial da plataforma, ou seja, com um número reduzido de utilizadores e de compras e vendas. No entanto existem algumas melhorias a nível de funcionalidades mais específicas que podem ser importantes para um próximo passo com mais utilizadores, como por exemplo uma página dedicada aos artigos que vendeu e comprou, no qual se possa pesquisar, filtrar por estado da compra/venda e data, entre outros. Outras melhorias que também se consideram importantes e úteis na próxima versão são:

- Colocar as notificações respetivas ao *marketplace* em tempo real dentro da plataforma;
- Adicionar um conversor de moedas possibilitando ao utilizador saber qual o preço de um artigo na moeda que utiliza;
- Agilizar o processo de compra possibilitando compras sem ter de ser aceites pelo vendedor. A única coisa que impede neste momento este agilização é os valores dos custos de envio, no qual pode ser resolvido utilizando uma API que calcule os mesmos ou implementar um sistema que permita ao vendedor configurar o preço de país para país;
- Adicionar outros métodos de pagamentos além do PayPal, como por exemplo o Stripe⁴⁰;
- Adicionar a possibilidade de trocas em que permita aos utilizadores em vez de proporem um preço, proporem um artigo para troca, no qual seria cobrado uma taxa por troca;
- Adicionar mais filtros à listagem de artigos no *marketplace*, como por exemplo filtrar por preço, popularidade, país, entre outros.

Em relação ao trabalho futuro do que foi melhorado no *BackOffice*, a melhoria mais evidente está no método do carregamento das mensagens na shoutbox. Atualmente estão a ser carregadas por GET de segundo a segundo, funcionando bem com muito poucos utilizadores e pouca utilização dos mesmos (o que é o caso atual). Mas com uma escalabilidade será necessário alterar a forma de obtenção das mensagens para *sockets*, evitando assim os pedidos constantes ao servidor.

⁴⁰ <https://stripe.com/>

Referências Bibliográficas

- [1] M. C. Farina, G. L. Toledo e G. B. Corrêa, “Colecionismo: Uma perspectiva abrangente sobre o comportamento do consumidor,” em *IX SEMEAD*, São Paulo, 2006.
- [2] T. C. COSTA, “Colecionando o invível: o reordenamento de mundo a partir de objetos de descarte,” Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2012.
- [3] J. R. Lopes, “Colecionismo e ciclos de vida: uma análise sobre percepção, duração e transitoriedade dos ciclos vitais,” *Horizontes Antropológicos* 16, 2010.
- [4] M. V. Cantagallo, O. M. Pépece e A. E. Pieske, “Colecionados & Colecionadores: motivação e escolha de temas no universo da Filatelia,” em *Seminário Internacional de Estudos e Pesquisas em Consumo*, São Paulo, 2012.
- [5] C. Quest, “Collecting Fact Sheet,” [Online]. Available: <http://www.collectorsquest.com/pages/press-room/collecting-fact-sheet/>. [Acedido em 12 10 2014].
- [6] R. Carthy, “Colnect Is A No-Frills Collectibles Marketplace And Wiki. Someone Wake Up David Cowan!,” Techcrunch, 14 06 2009. [Online]. Available: <http://techcrunch.com/2009/06/14/colnect-is-a-no-frills-collectibles-marketplace-and-wiki-someone-wake-up-david-cowan/>. [Acedido em 14 10 2014].
- [7] Colnect, “Collecting Help,” [Online]. Available: <http://colnect.com/en/help>. [Acedido em 14 10 2014].
- [8] D. Margulius, “Welcome! |Collectors Weekly,” Collectors Weekly, [Online]. Available: <http://www.collectorsweekly.com/about/>. [Acedido em 14 10 2014].
- [9] Collectors Quest, “About Us | Collectors Quest,” [Online]. Available: <http://www.collectorsquest.com/pages/about-us/?ref=footer>. [Acedido em 14 10 2014].
- [10] CrunchBase, “Kolektado | CrunchBase,” [Online]. Available: <http://www.crunchbase.com/organization/kolektado>. [Acedido em 14 10 2014].
- [11] Kolektado, “Welcome to Kolektado,” [Online]. Available: <http://kolektado.com/>. [Acedido em 14 10 2014].
- [12] M. Stelzner, “How Social Login Enhances Websites,” 22 05 2012. [Online]. Available: <http://www.socialmediaexaminer.com/how-social-login-enhances-membership-websites/>. [Acedido em 21 10 2014].

- [13] “Google Translate API,” [Online]. Available: <https://cloud.google.com/translate/docs>.
- [14] M. Duran, “YSlow,” [Online]. Available: <http://yslow.org/>.
- [15] M. Duran, “YSlow Ruleset Matrix,” [Online]. Available: <http://yslow.org/ruleset-matrix/>.
- [16] “Waterfall Model,” [Online]. Available: <http://www.waterfall-model.com/>. [Acedido em 27 11 2014].
- [17] T. Brown, Design Thinking - Uma metodologia poderosa para decretar o fim das velhas ideias, Campus - RJ, 2010.
- [18] R. Cohen, “Design Thinking: A Unified Framework For Innovation,” Forbes, 31 03 2014. [Online]. Available: <http://www.forbes.com/sites/reuvencohen/2014/03/31/design-thinking-a-unified-framework-for-innovation/>. [Acedido em 06 10 2014].
- [19] IDEO, “IDEO | A Design and Innovation Consulting Firm,” [Online]. Available: <http://www.ideo.com/>.
- [20] Scrum.Org and ScrumInc, “The Scrum Guide,” [Online]. Available: <http://www.scrumguides.org/scrum-guide.html>. [Acedido em 27 11 2014].
- [21] Scrum.org, “What is Scrum?,” [Online]. Available: <https://www.scrum.org/Resources/What-is-Scrum>. [Acedido em 27 11 2014].
- [22] Scrum Alliance, Inc, “Learn About Scrum,” [Online]. Available: <https://www.scrumalliance.org/why-scrum>. [Acedido em 27 11 2014].
- [23] Maxxor, “SCRUM Software Development Process,” [Online]. Available: <https://www.maxxor.com/software-development-process>. [Acedido em 27 11 2014].
- [24] Mountain Goat Software, “User Stories and User Story Examples by Mike Cohn,” [Online]. Available: <http://www.mountaingoatsoftware.com/agile/user-stories>. [Acedido em 27 11 2013].
- [25] Mountain Goat Software, “Planning Poker: Agile Estimating Made Easy,” [Online]. Available: <http://www.mountaingoatsoftware.com/tools/planning-poker>. [Acedido em 25 01 2015].
- [26] Mountain Goat Software, “Release Burndown Chart,” [Online]. Available: <http://www.mountaingoatsoftware.com/agile/scrum/release-burndown>. [Acedido em 18 01 2015].

- [27] [rubyonrails.org](http://guides.rubyonrails.org/getting_started.html#what-is-rails-questionmark), “Getting Started with Rails — Ruby on Rails Guides,” [Online]. Available: http://guides.rubyonrails.org/getting_started.html#what-is-rails-questionmark. [Acedido em 28 12 2014].
- [28] “Ruby on Rails,” [Online]. Available: <http://rubyonrails.org/>. [Acedido em 29 12 2014].
- [29] “About Ruby,” [Online]. Available: <https://www.ruby-lang.org/en/about/>. [Acedido em 30 12 2014].
- [30] A. Mejia, “Ruby on Rails Architectural Design,” 11 08 2011. [Online]. Available: <http://adrianmejia.com/blog/2011/08/11/ruby-on-rails-architectural-design/>. [Acedido em 29 12 2014].
- [31] Niwatori, “Ruby on Rails Diagram,” [Online]. Available: <https://picasaweb.google.com/Dikiwinky/Ruby#5116531304417868130>.
- [32] Oracle Corporation, “1.3.1 What is MySQL?,” [Online]. Available: <http://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>. [Acedido em 15 01 2015].
- [33] “1.3.2 The Main Features of MySQL,” Oracle Corporation, [Online]. Available: <http://dev.mysql.com/doc/refman/4.1/en/features.html>. [Acedido em 15 01 2015].
- [34] “Bootstrap · The world's most popular mobile-first and responsive front-end framework.,” [Online]. Available: <http://getbootstrap.com/>. [Acedido em 26 12 2014].
- [35] The jQuery Foundation, “jQuery,” [Online]. Available: <http://jquery.com/>.
- [36] PayPal, “Existing Developer Products - PayPal Developer,” [Online]. Available: <https://developer.paypal.com/docs/classic/products/>. [Acedido em 08 06 2015].
- [37] PayPal, “Adaptive Payments API Getting Started - PayPal Developer,” [Online]. Available: https://developer.paypal.com/docs/classic/adaptive-payments/gs_AdaptivePayments/. [Acedido em 2015 06 08].
- [38] M. A. Russell, “4. Adaptive Payments (Simple, Parallel, and Chained Payments) - PayPal APIs: Up and Running, 2nd Edition,” O'Reilly Media, Inc., 04 2012. [Online]. Available: <https://www.safaribooksonline.com/library/view/paypal-apis-up/9781449321666/ch04.html>. [Acedido em 08 06 2015].
- [39] PayPal, “IPN and PDT Variables - PayPal Developer,” [Online]. Available: <https://developer.paypal.com/webapps/developer/docs/classic/ipn/integration-guide/IPNandPDTVariables/>. [Acedido em 09 06 2015].

- [40] PayPal, “Instant Payment Notification Guide,” 10 07 2012. [Online]. Available: https://www.paypalobjects.com/webstatic/en_US/developer/docs/pdf/ipnguide.pdf. [Acedido em 09 06 2015].
- [41] PayPal, “Transferências seguras para compradores e vendedores - PayPal Portugal,” [Online]. Available: <https://www.paypal.com/pt/webapps/mpp/paypal-safety-and-security>. [Acedido em 08 06 2015].

Anexo A - *User Stories* implementadas e gráficos *Burndown Chart*

Neste presente anexo são apresentadas as *user stories* e os gráficos *burndown chart*, referentes a cada *sprint* realizado. Os dias contabilizados nos gráficos são apenas os dias que se trabalhou, excluindo assim fins-de-semana e feriados. As medições dos pontos foram realizadas de 2 em 2 dias. Por cada *sprint* é dado uma pequena descrição do que se fez e data em que se realizou. Cada *user story* é acompanhada por um identificador, categoria a que pertence, critérios de aceitação e número de pontos estimados.

As categorias presentes são:

- BackOffice – esta categoria representa todas as funcionalidades referentes ao *backoffice* da Collectors Bridge;
- Marketplace – esta categoria representa todas funcionalidades referentes ao módulo *marketplace*.

Os atores presentes são:

- Champion – é um tipo de utilizador responsável por gerir um portal, no qual possui uma área restrita no *backoffice* para esse fim;
- Administrador – é um tipo de utilizador que tem todos os privilégios de um utilizador autenticado, mais a possibilidade de acesso e gestão do *backoffice*;
- Utilizador – é um utilizador que não está autenticado no sistema, no qual pode visualizar todas as áreas públicas da plataforma;
- Utilizador autenticado - é um utilizador que está autenticado no sistema, no qual pode fazer tudo o que um utilizador faz (ver tudo o que é público), e aceder a diversas áreas restritas dentro da plataforma;
- Vendedor – é um utilizador que possui um artigo para venda no *marketplace*;
- Comprador – é um utilizador que possui uma licitação/compra no *marketplace*
- Membro da C'B – é um recurso humano da Collectors Bridge.

Sprint 1

Este *sprint* teve início no dia 15 de fevereiro de 2015 e terminou no dia 4 de março de 2015, totalizando-se em 13 dias de trabalho e 37 pontos. As 7 *user stories* (Tabela A-1) concretizadas incidiram-se nas melhorias e adição de novas funcionalidades ao *BackOffice*. Como se pode observar pela Figura A-1 o *sprint* teve um início um pouco lento, pois estava-se ainda em fase de conhecimento e adaptação do que já existia criado no *BackOffice*.

ID	Categoria	User Story	Crítérios de Aceitação	Pontos
1	BackOffice	Como champion Eu quero um shoutbox no BackOffice Para que possa comunicar com outros champions	a) Deverá ser reaproveitado o código da shoutbox existente; b) As mensagens deverão ser descarregadas de forma assíncrona de 1 em 1 segundos por GET e em formato JSON; c) A shoutbox deverá ficar debaixo do menu de navegação do BackOffice;	13
2	BackOffice	Como champion Eu quero ver sempre todas as shoutbox no qual sou champion em qualquer área do BackOffice Para que a comunicação fique mais fácil	a) A geração de shoutbox deverá ser dinâmica, de acordo com o número de portais em que o utilizador é champion; b) As mensagens contidas nas diversas shoutbox do champion deverão ser recebidas em apenas um único JSON.	3
3	BackOffice	Como champion Eu quero poder dizer para guardar ("keep") uma mensagem antes de a enviar numa shoutbox Para que que ela possa ser filtrada posteriormente	a) A opção de "keep" deverá estar junto à caixa de texto da shoutbox;	3
4	BackOffice	Como champion Eu quero uma funcionalidade que filtre por "keep" na shoutbox Para que consiga visualizar de forma mais rápidas as mensagens "keep"	a) A aplicação do filtro deverá ser do lado do cliente, não necessitando assim um novo pedido ao servidor;	3
5	BackOffice	Como champion Eu quero que se ignore ao pressionar a tecla "Enter" na caixa de pesquisa de champions para portais Para que possa primeiro pesquisar sem guardar as informações de um portal	a) Ao pressionar a tecla "Enter" nas caixas de pesquisas das edições/inserções do BackOffice, não deverá submeter o formulário.	2

6	BackOffice	<p>Como champion Eu quero uma melhoria na aparência do BackOffice Para que fique mais semelhante ao FrontOffice</p>	<p>a) As melhorias deverão ser apenas alterações no ficheiro de CSS do BackOffice; b) Deverá ser utilizado o mesmo esquema de cores do FrontOffice; c) A largura da página deverá continuar adaptativa ao tamanho da janela (ao contrário do FrontOffice); d) A altura da barra de topo do BackOffice deverá ser igual à do FrontOffice; e) Deverá ser mostrado o logótipo da Collectors Bridge na barra de topo.</p>	5
7	BackOffice	<p>Como champion Eu quero a possibilidade de gerir séries dentro de um portal Para que possa saber pela série num portal, quais os artigos que lhe pertencem</p>	<p>a) A gestão de séries deverá ser acessível ao administrador e champion; b) O administrador deverá conseguir ver todas as séries de todos os portais; c) O champion deverá conseguir ver apenas as séries dos portais que é champion;</p>	8

Tabela A-1 - User Stories do 1º sprint

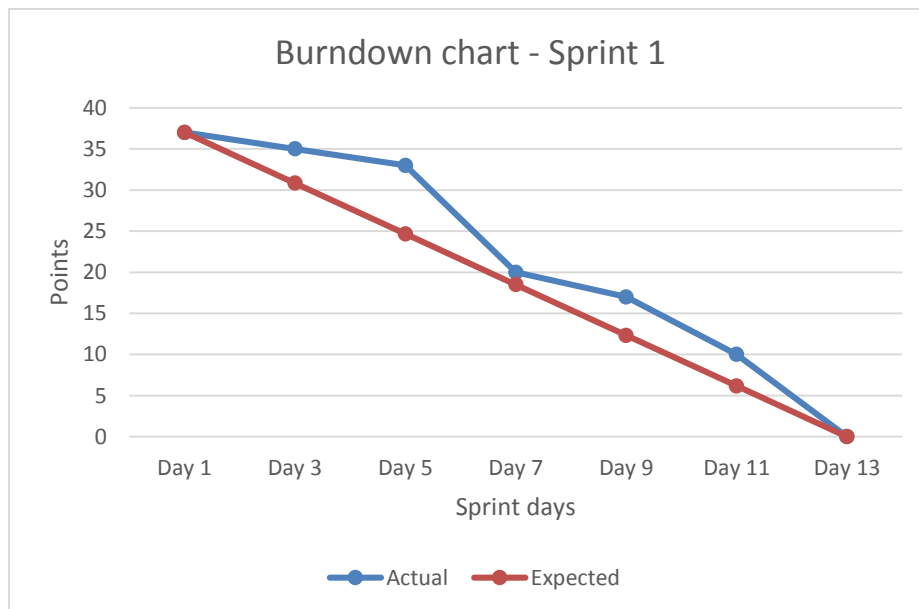


Figura A-1 - Burndown chart do 1º sprint

Sprint 2

Este *sprint* teve início no dia 5 de março de 2015 e terminou no dia 27 de março de 2015, totalizando-se em 17 dias de trabalho e 52 pontos. As 10 *user stories* (Tabela A-2) concretizadas incidiram-se no desenvolvimento do desenho do sistema e implementação da estrutura inicial do *marketplace* à plataforma existente. No final do *sprint* já se podia inserir, listar e ver detalhes dos artigos no *marketplace*. Como se pode observar pela Figura A-2 o *sprint* teve um bom início, pois não envolveu tarefas de implementação de código. Depois

houve um ligeiro atraso, pois iniciou-se a implementação das funcionalidades integrando-as ao que já existia no *frontOffice*, e como não havia documentação de código e desenho do sistema do que se encontrava feito, o arranque levou mais tempo de que se esperava. Depois de se conhecer minimamente o que já estava implementado, então foi possível recuperar esse tempo e terminar o *sprint* cumprindo a implementação de todas as funcionalidades previstas.

<i>ID</i>	<i>Categoria</i>	<i>User Story</i>	<i>CrITÉrios de Aceitação</i>	<i>Pontos</i>
8	Marketplace	Como membro da C'B Eu quero um diagrama de estados de compra/venda de um artigo do <i>marketplace</i> Para que possa perceber de forma fácil, quais os estados de um artigo e as ações que os despertam	a) O diagrama tem de ser aprovado pelo eng. Pedro Martins.	2
9	Marketplace	Como membro da C'B Eu quero um diagrama de estados de uma licitação a um artigo do <i>markeplace</i> Para que possa perceber de forma fácil, quais os estados de uma licitação e as ações que as despertam	a) O diagrama tem de ser aprovado pelo eng. Pedro Martins.	2
10	Marketplace	Como membro da C'B Eu quero um diagrama de modelos com as tabelas necessárias para a construção do <i>marketplace</i> Para que possa perceber como o Marketplace irá se integrar com o que já existe	a) O diagrama tem de ser aprovado pelo eng. Pedro Martins; b) Depois do diagrama estar aprovado, deverão ser implementados os modelos	5
11	Marketplace	Como membro da C'B Eu quero um protótipo em papel do <i>marketplace</i> Para que possa perceber previamente como vai ser o resultado final	a) Deverão ser construídos ou alterados <i>mockups</i> necessários para representar o <i>marketplace</i> ; b) Os <i>mockups</i> deverão ser aprovados pelo eng. Pedro Martins;	3
12	Marketplace	Como utilizador Eu quero uma opção de acesso ao Marketplace no menu lateral Para que possa ter acesso ao <i>Marketplace</i> em qualquer parte da plataforma	a) A presença de um link para o <i>marketplace</i> com o respetivo ícone.	2
13	Marketplace	Como comprador Eu quero ver uma lista de artigos de que licitei ou comprei na página inicial do <i>Marketplace</i> , em que mostre o nome e o estado da compra ou licitação Para que possa ter uma visão geral sobre o estado das minhas compras ou licitações	a) Para cada elemento da lista deverão ser mostrados o nome e estado da compra;	5

14	Marketplace	<p>Como utilizador autenticado</p> <p>Eu quero poder inserir artigos para venda</p> <p>Para que possam ser incorporados no Marketplace</p>	<p>a) A inserção de artigos deverá ser realizada em dois passos. No primeiro passo deverá ser pedido o nome do artigo, descrição e uma imagem de capa. No segundo passo deverá ser pedido possível alterar as informações do primeiro passo e deverá ser pedido o preço do artigo, tipo de moeda aceite no preço, país onde se encontra o artigo, categoria, <i>email</i> do PayPal (para receber o pagamento), e opcionalmente o custo de envio para diversas zonas.</p> <p>b) As categorias, tipos de moeda, países e zonas de envio deverão estar já previamente criadas no sistema;</p> <p>c) Ao preencher o formulário corretamente, os dados deverão ser salvos na base de dados;</p> <p>d) A opção para adicionar um novo artigo ao <i>Marketplace</i> deverá estar na página principal do <i>Marketplace</i>;</p> <p>e) Os dados deverão ser validados no lado do cliente e no lado do servidor;</p> <p>f) Ao salvar o artigo, este não deverá ser publicado</p>	12
15	Marketplace	<p>Como vendedor</p> <p>Eu quero uma lista de artigos de que estou a vender na página inicial do Marketplace, em que mostre o nome e o estado da venda</p> <p>Para que possa ter uma visão geral sobre o estado das minhas vendas</p>	<p>a) Para cada elemento da lista deverão ser mostrados o nome e estado da venda;</p>	5
16	Marketplace	<p>Como utilizador</p> <p>Eu quero ver uma lista de artigos que estão à venda</p> <p>Para que possa saber e encontrar mais rapidamente artigos que me interessem</p>	<p>a) Para cada elemento da lista deverão ser mostrados uma fotografia, nome e preço;</p>	8
17	Marketplace	<p>Como utilizador</p> <p>Eu quero ver detalhes de um artigo para venda</p> <p>Para que possa saber mais informações sobre o mesmo</p>	<p>a) Os detalhes deverão ser mostrados numa nova página;</p> <p>b) Os detalhes a serem mostrados são: imagem do artigo, nome, descrição, <i>rating</i> do vendedor, localização do artigo (país), preço e uma galeria fotográfica do artigo;</p>	8

Tabela A-2 - *User Stories* do 2º *sprint*

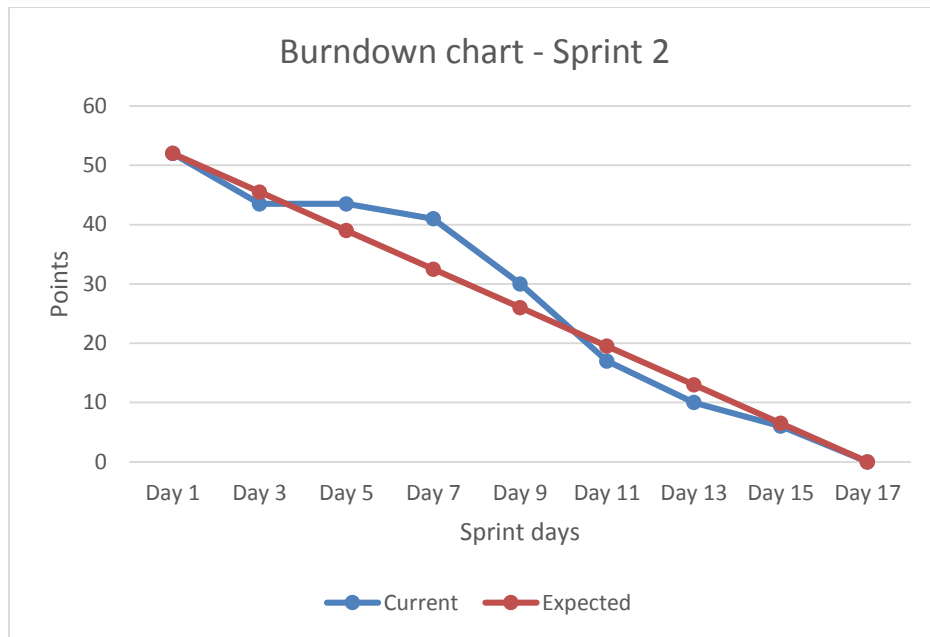


Figura A-2 - Burndown chart do 2º sprint

Sprint 3

Este *sprint* teve início no dia 30 de março de 2015 e terminou no dia 22 de abril de 2015, totalizando-se em 17 dias de trabalho e 50 pontos. As 15 *user stories* (Tabela A-3) concretizadas incidiram-se no desenvolvimento diversas funções no *marketplace*. No final do *sprint* podia-se licitar artigos, rejeitar licitações, editar detalhes de artigos, cancelar vendas, fazer perguntas e dar respostas, ou seja, quase todas as ações do diagrama de estados de um artigo, até ao ato de pagar. Como se pode observar pela Figura A-3 o *sprint* não teve grandes desvios ao plano estimado e foi concluído como previsto.

ID	Categoria	User Story	Crítérios de Aceitação	Pontos
18	Marketplace	Como vendedor Eu quero poder inserir várias imagens no meu artigo a ser colado à venda Para que o mesmo fique melhor ilustrado	a) A adição de imagens só deverá ser possível quando o artigo estiver no estado de não publicado; b) Deverá ser utilizado o sistema de carregamento de imagens já existente na plataforma;	2
19	Marketplace	Como vendedor Eu quero poder fazer <i>crop</i> às imagens da galeria do meu artigo Para que eu possa mostrar os pormenores que pretendo	a) O <i>crop</i> das imagens só deverá ser possível quando o artigo estiver no estado de não publicado; b) Deverá ser utilizado o sistema de <i>crop</i> de imagens já existente na plataforma;	3

20	Marketplace	<p>Como vendedor</p> <p>Eu quero poder fazer <i>crop</i> à imagem de capa do meu artigo</p> <p>Para que eu possa mostrar pormenores</p>	<p>a) O <i>crop</i> da imagem de capa só deverá ser possível quando o artigo estiver no estado de não publicado;</p> <p>b) Deverá ser utilizado o sistema de <i>crop</i> de imagens já existente na plataforma;</p>	3
21	Marketplace	<p>Como vendedor</p> <p>Eu quero poder selecionar uma imagem da galeria do meu artigo como imagem de capa</p> <p>Para que eu não tenha de fazer novamente um <i>upload</i></p>	<p>a) A alteração da imagem de capa só deverá ser possível quando o artigo estiver no estado de não publicado;</p>	2
22	Marketplace	<p>Como vendedor</p> <p>Eu quero poder fazer <i>upload</i> de uma nova imagem de capa do meu artigo</p> <p>Para que possa a alterar caso pretenda</p>	<p>a) A alteração da imagem de capa só deverá ser possível quando o artigo estiver no estado de não publicado;</p> <p>b) Deverá ser utilizado o sistema de carregamento de imagens já existente na plataforma;</p>	3
23	Marketplace	<p>Como vendedor</p> <p>Eu quero poder apagar imagens da galeria do meu artigo</p> <p>Para que elas não sejam mostradas posteriormente ao público</p>	<p>a) A eliminação das imagens da galeria só deverão ser possíveis quando o artigo estiver no estado de não publicado;</p>	2
24	Marketplace	<p>Como vendedor</p> <p>Eu quero poder inserir ou editar a descrição das imagens da galeria do meu artigo</p> <p>Para que os interessados no produto possam ter mais informações sobre as mesmas</p>	<p>a) A inserção ou edição da descrição das imagens da galeria só deverá ser possíveis quando o artigo estiver no estado de não publicado;</p>	3
25	Marketplace	<p>Como vendedor</p> <p>Eu quero poder publicar um artigo no <i>marketplace</i></p> <p>Para que ele possa ser visível pelos utilizadores</p>	<p>a) O artigo só poderá ser publicado se o seu estado for não publicado e tenha um nome, imagem de capa, categoria, país, preço e tipo de moeda aceite devidamente preenchidos;</p>	3
26	Marketplace	<p>Como vendedor</p> <p>Eu quero poder editar o artigo que estou a vender, desde que não haja licitações</p> <p>Para que possa alterar as informações do mesmo</p>	<p>a) O artigo só poderá ser publicado se o seu estado for à espera de licitações</p>	2
27	Marketplace	<p>Como utilizador registado</p> <p>Eu quero poder fazer perguntas ao comprador de um determinado artigo à venda</p> <p>Para que possa saber mais informações sobre o mesmo</p>	<p>a) O vendedor do artigo não poderá colocar perguntas no seu próprio artigo;</p> <p>b) As perguntas colocadas devem ser visíveis apenas para o autor da pergunta e o vendedor do artigo;</p> <p>d) A identificação do autor da pergunta deverá ser com a primeira e última letra do nome do mesmo;</p>	8
28	Marketplace	<p>Como vendedor</p> <p>Eu quero poder responder às perguntas efetuadas pelos utilizadores registados sobre o meu artigo à venda</p> <p>Para que possam ficar mais esclarecidos sobre o mesmo</p>	<p>a) Quando responder a uma pergunta, a mesma deverá ficar visível ao público juntamente com a resposta;</p> <p>b) Só deverá ser possível responder às perguntas quando o artigo se encontra num estado visível ao público;</p>	5

29	Marketplace	Como vendedor Eu quero poder cancelar o artigo que estou a vender, desde que não haja um pagamento já realizado Para que não fique mais visível no <i>marketplace</i>	a) O artigo só poderá ser cancelado se não tiver nenhum pagamento a decorrer ou concluído sobre o mesmo; b) Se o artigo tiver licitações ativas, estas devem ser recusadas automaticamente;	3
30	Marketplace	Como utilizador registado Eu quero poder licitar um artigo à venda no <i>marketplace</i> Para que possa ser comprado posteriormente	a) Ao licitar um artigo, o utilizador registado poderá aceitar ou propor um novo preço, escolher o país e região onde pretende receber o artigo e escolher o número de dias em que a sua licitação é válida;	5
31	Marketplace	Como vendedor Eu quero ver todas as licitações à espera de serem aceites ao artigo que estou a vender Para que possa aceitar ou recusar posteriormente	a) As licitações deverão aparecer numa tabela na página do artigo; b) Para cada licitação deverá aparecer a data de expiração, a cidade e país de destino, o preço pretendido e a primeira e ultima letra do autor da mesma;	3
32	Marketplace	Como vendedor Eu quero poder rejeitar uma licitação sobre o meu artigo Para que para que o utilizador autor da mesma fique a saber que a sua proposta não me agrada	a) O vendedor só poderá rejeitar uma licitação se o estado do artigo for “com licitações”;	3

Tabela A-3 - User Stories do 3º sprint

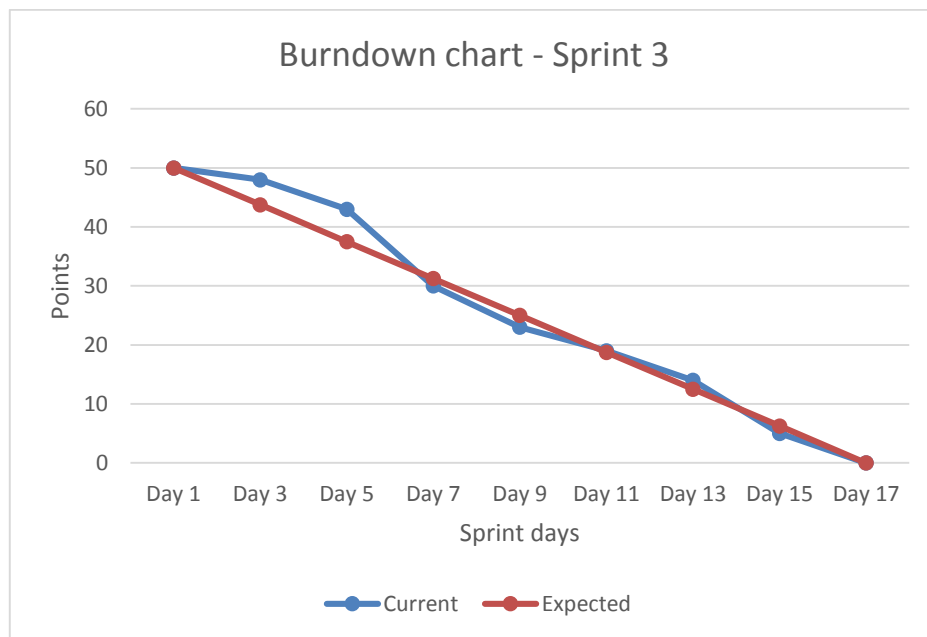


Figura A-3- Burndown chart do 3º sprint

Sprint 4

Este *sprint* teve início no dia 27 de abril de 2015 e terminou no dia 15 de maio de 2015, totalizando-se em 15 dias de trabalho e 53 pontos. As 13 *user stories* (Tabela A-4)

concretizadas incidiram-se no desenvolvimento várias funções ao *marketplace*, como por exemplo, pagamentos, históricos dos estados, classificação de utilizadores e *feedbacks*. No final do *sprint* já se podia fazer compras e vendas no *marketplace*. Como se pode observar pela Figura A-4 o *sprint* não teve grandes desvios ao plano estimado e foi concluído como previsto.

<i>ID</i>	<i>Categoria</i>	<i>User Story</i>	<i>Critérios de Aceitação</i>	<i>Pontos</i>
33	Marketplace	Como membro da C'B Eu quero um documento que indique quais as formas possíveis no PayPal de obter uma comissão para a C'B nas transações entre comprador e vendedor Para que possa ajudar-me a decidir qual a forma a usar	a) O documento tem de ser aprovado pelo eng. Pedro Martins;	3
34	Marketplace	Como utilizador Eu quero poder ver o <i>feedback</i> dos vendedores de cada artigo à venda Para que possa saber a opinião dos outros utilizadores	a) O <i>feedback</i> deverá aparecer numa <i>popup</i> , ao clicar sobre a identificação do vendedor na página do artigo;	3
35	Marketplace	Como vendedor Eu quero poder ver o <i>rating</i> e <i>feedback</i> dos utilizadores que possuem licitações sobre meu artigo Para que possa saber a opinião dos outros vendedores sobre os mesmos	a) O <i>feedback</i> deverá aparecer numa <i>popup</i> , ao clicar sobre a identificação dos utilizadores na lista de licitações;	3
36	Marketplace	Como vendedor Eu quero poder aceitar uma licitação sobre o meu artigo Para que o comprador possa pagar ou recusar	a) O vendedor só poderá aceitar uma licitação se o estado do artigo for com licitações; b) Antes de aceitar o vendedor deverá dizer o valor do custo de envio e poderá alterar o <i>email</i> do PayPal; c) Ao ser aceite o sistema deverá criar um pedido de pagamento no PayPal, para que possa ser paga posteriormente pelo comprador;	8
37	Marketplace	Como comprador Eu quero poder rejeitar uma licitação aceite pelo vendedor Para que o vendedor fique a saber que a proposta não me agrada	a) Ao rejeitar a licitação o comprador deverá ficar apto para licitar novamente o artigo;	2

38	Marketplace	<p>Como comprador Eu quero poder pagar uma licitação aceite pelo vendedor Para que o mesmo me possa enviar o artigo</p>	<p>a) O <i>checkout</i> do artigo deverá ser feito no PayPal; b) Deverá ser pedido os dados de envio no PayPal para que a transação fique protegida contra possíveis fraudes; c) Se o PayPal responder como pagamento falhado, o sistema deverá fazer outro pedido de requisição de pagamento ao PayPal e renovar a chave de pagamento; d) Se o PayPal responder que o pagamento está concluído, o sistema deverá rejeitar automaticamente todas as licitações pendentes sobre o artigo;</p>	13
39	Marketplace	<p>Como vendedor Eu quero poder dizer no sistema que o artigo já foi enviado Para que o comprador fique a saber</p>	<p>a) O vendedor só poderá dizer que o artigo foi enviado quando o estado do artigo é de pagamento concluído;</p>	2
40	Marketplace	<p>Como comprador Eu quero poder confirmar a receção no sistema e avaliar o vendedor Para que o vendedor fique a saber que o artigo já chegou ao destino</p>	<p>a) O comprador só poderá confirmar a receção do artigo, caso o estado do mesmo seja pago ou enviado; b) Ao confirmar a receção do artigo, o comprador deverá dizer se a compra correu bem ou mal, podendo dar <i>feedback</i> de forma escrita;</p>	3
41	Marketplace	<p>Como comprador Eu quero poder avaliar o vendedor caso a compra falhe Para que os próximos compradores fiquem com <i>feedback</i> do vendedor</p>	<p>a) Na avaliação da compra o comprador deverá dizer se a mesma correu bem ou mal, podendo dar <i>feedback</i> de forma escrita. B) Esta avaliação só poderá ser efetuada caso o estado do artigo seja compra falhada;</p>	3
42	Marketplace	<p>Como vendedor Eu quero poder avaliar o comprador no fim de uma venda bem-sucedida ou falhada Para que os próximos vendedores fiquem com <i>feedback</i> do comprador</p>	<p>a) Na avaliação da venda o vendedor deverá dizer se a mesma correu bem ou mal, podendo dar <i>feedback</i> de forma escrita. B) Esta avaliação só poderá ser efetuada caso o estado do artigo seja compra falhada ou artigo recebido;</p>	3
43	Marketplace	<p>Como vendedor Eu quero ver o histórico das licitações e mudanças de estado efetuadas sobre o meu artigo Para que possa saber as datas das ações efetuadas sobre o mesmo no passado</p>	<p>a) O histórico deverá aparecer ao vendedor numa tabela na página do artigo; b) Deverão ser registado todas as mudanças de estado de um artigo, assim como as licitações sobre o mesmo;</p>	5
44	Marketplace	<p>Como comprador Eu quero ver histórico das licitações que efetuei a um artigo Para que saber as datas que efetuei ações sobre o mesmo</p>	<p>a) O histórico deverá aparecer ao comprador numa tabela na página do artigo; b) Deverão ser registado todas as mudanças de estado das licitações do comprador sobre um artigo;</p>	5

45	Marketplace	<p>Como membro da C'B</p> <p>Eu quero um documento que indique quais os valores das comissões cobradas pelos concorrentes</p> <p>Para que possa ajudar-me a decidir qual a percentagem a ser cobrada pela Collectors Bridge</p>	<p>a) Deverão ser analisados as seguintes plataformas: delcampe, Collectors Conner, todocoleccion, gemr e eBay;</p> <p>b) Por cada plataforma deverão ser indicados os valores das comissões numa compra de 50€ e 500€</p>	2
----	-------------	--	--	---

Tabela A-4 - User Stories do 4º sprint

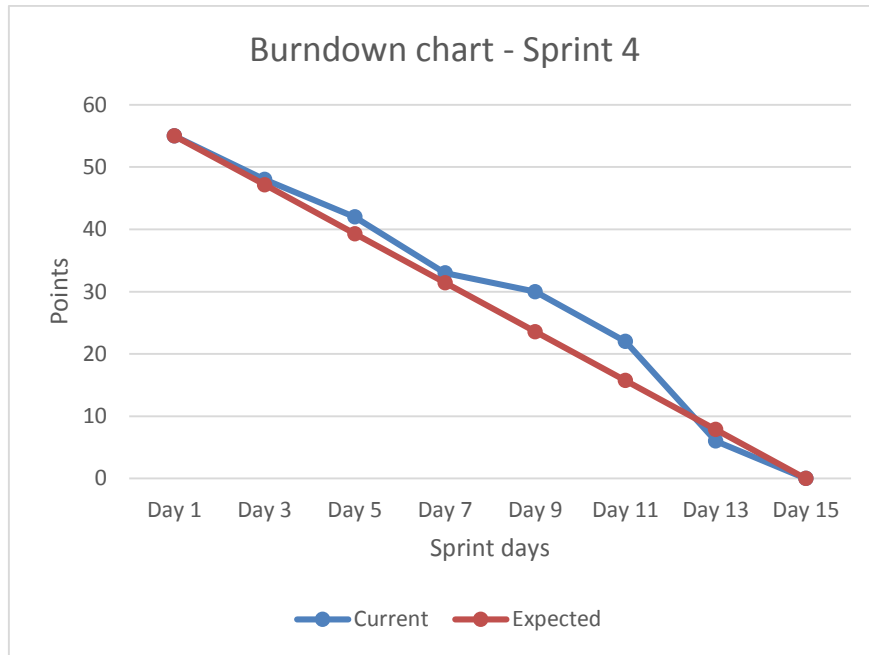


Figura A-4 - Burndown chart do 4º sprint

Sprint 5

Este *sprint* teve início no dia 18 de abril de 2015 e terminou no dia 5 de junho de 2015, totalizando-se em 15 dias de trabalho e 52 pontos. As 16 *user stories* (Tabela A-5) concretizadas incidiram-se nas funcionalidades de notificações, expirações de licitações/compras, *watchlist*, partilhas nas redes sociais e colocação de artigos no *marketplace* através de itens de coleções. Como se pode observar pela Figura A-5 o *sprint* não teve grandes desvios ao plano estimado e foi concluído como previsto.

ID	Categoria	User Story	Crítérios de Aceitação	Pontos
46	Marketplace	<p>Como comprador</p> <p>Eu quero ser notificado por <i>email</i> sempre que ocorram mudanças de estado na minha compra/licitação, que não foram encadeadas por mim</p> <p>Para que possa estar atualizado, mesmo não estando dentro da plataforma</p>	a) Os <i>emails</i> deverão ser enviados de forma assíncrona;	2

47	Marketplace	<p>Como vendedor</p> <p>Eu quero ser notificado por <i>email</i> sempre que surgem questões ou ocorrerem mudanças de estado na minha venda, que não foram encadeadas por mim</p> <p>Para que possa estar atualizado, mesmo não estando dentro da plataforma</p>	<p>a) Os <i>emails</i> deverão ser enviados de forma assíncrona;</p>	2
48	Marketplace	<p>Como utilizador registado</p> <p>Eu quero ser notificado por <i>email</i> quando o vendedor responder à minha questão sobre o seu artigo</p> <p>Para que possa estar saber que esta foi respondida</p>	<p>a) Os <i>emails</i> deverão ser enviados de forma assíncrona;</p>	2
49	Marketplace	<p>Como comprador</p> <p>Eu quero ser notificado na plataforma sempre que ocorram mudanças de estado nas minhas compras/licitações, que não foram encadeadas por mim</p> <p>Para que possa identificar o que ocorreu mais facilmente</p>	<p>a) No menu geral da plataforma deverá ser incrementado o número de notificações;</p> <p>b) Deverá ser incrementa o número de notificações na compra/licitação respetiva do painel de comprador;</p> <p>c) Deverá ser identificado na página do artigo por um símbolo o que foi atualizado;</p> <p>d) As notificações do comprado sobre um artigo deverão desaparecer depois da página ser aberta pelo mesmo;</p>	5
50	Marketplace	<p>Como vendedor</p> <p>Eu quero ser notificado na plataforma sempre que ocorram novas perguntas sobre os meus artigos à venda ou mudanças de estado sobre os mesmos, que não foram encadeadas por mim</p> <p>Para que possa identificar as atualizações mais facilmente</p>	<p>a) No menu geral da plataforma deverá ser incrementado o número de notificações;</p> <p>b) Deverá ser incrementa o número de notificações na venda respetiva do painel de vendedor;</p> <p>c) Deverá ser identificado na página do artigo por um símbolo o que foi atualizado;</p> <p>d) As notificações do vendedor de um artigo deverão desaparecer depois da página ser aberta pelo mesmo;</p>	5
51	Marketplace	<p>Como utilizador registado</p> <p>Eu quero poder copiar um artigo de uma coleção minha para o <i>marketplace</i></p> <p>Para que não tenha de carregar tudo de novo para a plataforma</p>	<p>a) Deverá ser copiado o nome, descrição, imagem de capa e galeria de imagens do item da coleção;</p> <p>b) Ao carregar em copiar o sistema deverá ir para o passo 2 de inserção de artigos no <i>marketplace</i>;</p>	3
52	Marketplace	<p>Como comprador</p> <p>Eu quero que a minha licitação seja expirada pelo sistema quando chegar o dia que eu defini previamente</p> <p>Para que a mesma não fique eternamente no sistema</p>	<p>a) A verificação das expirações deverá ser feita utilizando a gem <i>clock_work</i>;</p>	3
53	Marketplace	<p>Como vendedor</p> <p>Eu quero que a licitação que aceitei seja expirada pelo sistema se o comprador não responder num tempo limite</p> <p>Para que a mesma não fique eternamente no sistema</p>	<p>a) A verificação das expirações deverá ser feita utilizando a gem <i>clock_work</i>;</p> <p>b) O limite de dias até ocorrer a expiração é 2;</p>	3

54	Marketplace	<p>Como vendedor</p> <p>Eu quero que o artigo que eu enviei para o comprador seja expirado, caso ele não confirme no sistema a receção do mesmo,</p> <p>Para que possa finalizar a venda</p>	<p>a) A verificação das expirações deverá ser feita utilizando a gem <i>clock_work</i>;</p> <p>b) O limite de dias até ocorrer a expiração é 60;</p>	3
55	Marketplace	<p>Como comprador ou vendedor</p> <p>Eu quero que a minha compra/venda seja finalizada de forma automática, caso não seja dado feedback num máximo de dias;</p> <p>Para que o processo seja concluído</p>	<p>a) A verificação das expirações deverá ser feita em utilizando a gem <i>clock_work</i>;</p> <p>b) O limite de dias até ocorrer a expiração é 10;</p>	3
56	Marketplace	<p>Como utilizador</p> <p>Eu quero poder pesquisar artigos no <i>marketplace</i> introduzindo texto</p> <p>Para que eu encontre mais rapidamente o que pretendo</p>	<p>a) Os atributos dos artigos no qual o sistema deve pesquisar é o nome e descrição;</p> <p>b) Os resultados da pesquisa devem aparecer na página inicial do <i>marketplace</i>;</p>	3
57	Marketplace	<p>Como utilizador</p> <p>Eu quero poder filtrar os artigos do <i>marketplace</i> por categorias</p> <p>Para que eu encontre mais rapidamente o que pretendo</p>	<p>b) Os resultados do filtro devem aparecer na página inicial do <i>marketplace</i>;</p>	3
58	Marketplace	<p>Como utilizador</p> <p>Eu quero que os artigos sejam apresentados num formato de páginas e possa escolher quantas linhas de artigos aparecem em cada página</p> <p>Para que eu possa dissidir se desejo ver muitos ou poucos artigos por página</p>	<p>a) Deverá ser utilizado a funcionalidade genérica de atribuição de número de linhas por página, presente na plataforma;</p> <p>b) Deverá ser utilizado o sistema de paginação já existente na plataforma;</p>	2
59	Marketplace	<p>Como utilizador registado</p> <p>Eu quero poder partilhar um artigo do <i>marketplace</i> nas redes sociais</p> <p>Para que o possa mostrar noutras comunidades</p>	<p>a) Deverá ser integrado com a funcionalidade de partilha nas redes sociais da plataforma;</p> <p>b) A opção de partilha deverá estar presente na página inicial do <i>marketplace</i> em cada artigo listado e dentro da página de um artigo;</p> <p>c) O botão de partilha deverá estar presente para utilizadores não registados, sendo que ao pressionar deverá surgir a <i>modalbox</i> de autenticação;</p>	5
60	Marketplace	<p>Como utilizador registado</p> <p>Eu quero poder adicionar ou remover artigos do <i>marketplace</i> à minha <i>watchlist</i></p> <p>Para que eu possa encontrá-los de uma forma mais fácil posteriormente</p>	<p>a) Um vendedor de um artigo não poderá adicionar o seu artigo à sua <i>watchlist</i>;</p> <p>b) O botão de adicionar/remover da <i>watchlist</i> deverá estar em cada artigo presente na página inicial do <i>marketplace</i> e na página de um artigo;</p> <p>c) O botão de adicionar/remover da <i>watchlist</i> deverá estar presente para utilizadores não registados, sendo que ao pressionar deverá surgir a <i>modalbox</i> de autenticação;</p>	5

61	Marketplace	<p>Como utilizador registado</p> <p>Eu quero poder ver todos os meus artigos que marquei como <i>watch</i></p> <p>Para que eu os possa encontrar de forma mais fácil</p>	<p>a) O filtro só poderá ser aplicado na página inicial do <i>marketplace</i>;</p>	3
----	-------------	---	--	---

Tabela A-5 - User Stories do 5º sprint

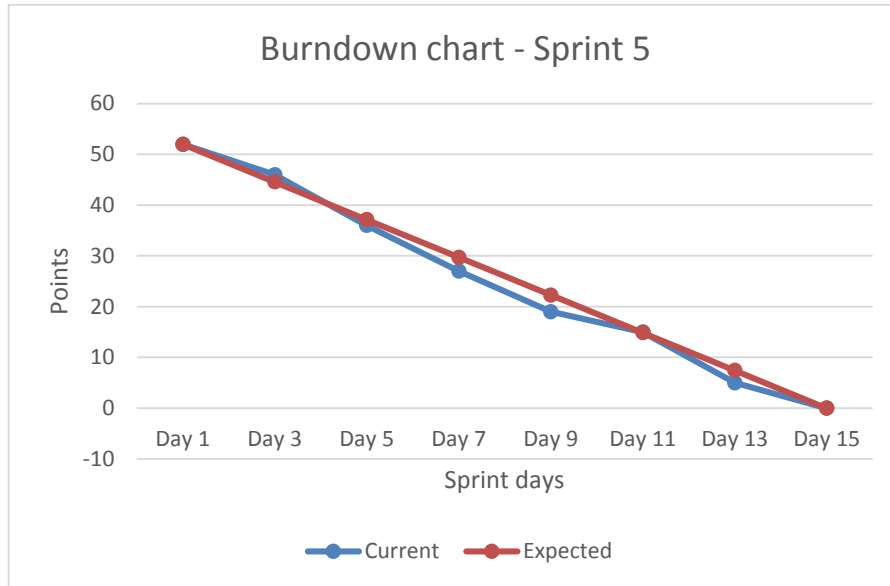


Figura A-5 - Burndown chart do 5º sprint

Anexo B - Análise de riscos

Neste anexo estão descritos e avaliados todos os riscos encontrados ao longo do projeto de estágio. Por cada risco está atribuído um identificador, um título, uma descrição dizendo em que consiste e um plano de mitigação. Como está descrito na secção 5 do capítulo 3 deste presente relatório, utilizou-se três parâmetros para a avaliação de cada risco, sendo eles: probabilidade (baixa, média, alta e muito alta), grau de impacto (baixo, médio, alto e muito alto) e magnitude (baixa, moderada, alta e crítica). A classificação da probabilidade e o grau de impacto foram atribuídos pela percepção do estagiário e orientador da empresa. A magnitude foi extraída por uma matriz de exposição de riscos que relaciona a probabilidade com o grau de impacto.

De seguida serão apresentados os riscos e depois é mostrada a avaliação dos mesmos ao longo do projeto.

1. Descrição dos riscos

R.1. Inexperiência em *Ruby on Rails*

- Descrição: A inexperiência do estagiário na tecnologia de *Ruby on Rails* poderá levar ao atraso das tarefas de implementação.
- Plano de mitigação: O estagiário deverá continuar praticando a tecnologia, estudando e fazendo tutoriais. No entanto há a possibilidade da empresa contratada pela Collectors Bridge, que está a desenvolver a plataforma responder a algumas dúvidas sobre a tecnologia, caso seja necessário.

R.2. Mal previsão do esforço necessário

- Descrição: A inexperiência do estagiário na previsão do tempo necessário para executar determinada tarefa poderá levar ao atraso da mesma.
- Plano de mitigação: Deverá ser utilizado a técnica para estimação de esforço descrita na secção 3.1.2 deste relatório. Durante o decorrer do *sprint* deverá ser utilizado um gráfico *Burndown chart*, com a finalidade de detetar possíveis falhas de previsão.

R.3. Definição da fronteira do trabalho desenvolvido

- Descrição: Sendo este projeto de estágio individual, a má definição da fronteira entre o trabalho desenvolvido e o já existente poderá prejudicar a avaliação do estagiário.
- Plano de mitigação: Todo o trabalho desenvolvido pelo estagiário deverá estar muito bem documentado no relatório final, explicando sempre o que já existia e o que fez de novo.

R.4. Integração no código do projeto

- Descrição: Este risco foi sugerido pelo júri na defesa intermédia do estágio no qual se concordou. Consiste na integração por parte do estagiário no desenvolvimento do código do projeto, ou seja, até defesa intermédia não foi disponibilizado qualquer acesso ao mesmo, podendo assim atrasar o projeto.
- Plano de mitigação: Deverá ser reservado uma semana antes de iniciar os *sprints* de desenvolvimento para configurações de ambientes e estudo inicial da estrutura do código do projeto existente.

R.5. Utilização de diferentes controladores de versões

- Descrição: O desenvolvimento das funcionalidades ou módulos irá ser realizado em simultâneo com a empresa que está também a desenvolver a plataforma. No entanto não irá ser possível ter acesso ao mesmo controlador de versões e repositório de código dessa empresa, no qual se criará outro para a Collectors Bridge. O eng. Pedro Martins irá fazer o *merge* do código mensalmente para repositório da Collectors Bridge e o *merge* final para entrar em pré-produção e produção será feito quando a empresa acabar a sua parte do desenvolvimento. Ter estes dois repositórios e controladores de código diferentes poderá dar conflitos ao fazer *merge* e provocar alterações inesperadas nas funcionalidades já existentes.
- Plano de mitigação: Deverá ser evitado alterar código já existente e sempre que possível criar código em ficheiros separados.

R.6. Fraca documentação do que está implementado

- Descrição: A nível de documentação de código e arquitetura, apenas existe um diagrama ER, o que poderá atrasar o desenvolvimento das funcionalidades ou módulos a implementar.
- Plano de mitigação: Sempre que existirem dúvidas de como está implementado uma certa funcionalidade, deverá ser pedido à empresa que está também a desenvolver o código para dar uma explicação sobre o mesmo.

R.7. API PayPal

- Descrição: A Collectors Bridge utiliza a API do PayPal para os pagamentos e uma alteração nos métodos utilizados por parte do PayPal poderá ser crítica para o correto funcionamento do sistema.
- Plano de mitigação: Não existe um plano eficaz para combater este risco. A plataforma deverá adaptar-se às alterações feitas na API.

2. Avaliação dos riscos

A Tabela B-1 mostra a avaliação completa dos riscos durante o projeto, estando avaliados para cada a probabilidade, grau de impacto e magnitude. A Figura B-1 mostra um gráfico que ilustra a evolução da magnitude de cada risco ao longo das diversas datas das avaliações. A primeira avaliação foi na primeira parte do estágio, onde se identificou apenas 3 riscos. No início do segundo semestre surgiu dois novos riscos relacionados com a integração do estagiário no desenvolvimento do código do projeto. Depois fez-se uma avaliação no início de cada *sprint* e uma última avaliação no final do projeto. Como se esperava, a magnitude dos 2 primeiros riscos foram diminuindo ao longo do projeto, pois como são riscos ligados à experiência, esta foi ganha ao longo do mesmo. O terceiro risco persistiu até metade do desenvolvimento, mas depois de se ter ganho um bom conhecimento do código este desapareceu. O quarto risco apenas persistiu durante o início do projeto, pois foi logo mitigado com uma semana reservada para combater o mesmo. O quinto risco foi identificado antes de iniciar o desenvolvimento e a sua magnitude foi diminuindo à medida que se fez os primeiros *merges* do código sem problemas. O sexto risco apareceu depois de se ter iniciada a implementação, pois expirava-se mais documentação, e desapareceu no decorrer do desenvolvimento, quando se ganhou um bom conhecimento do código. O último risco apareceu quando se integrou o PayPal à plataforma e este deve-se manter no futuro.

		Riscos							
		R.1	R.2	R.3	R.4	R.5	R.6	R.7	
Datas das avaliações dos riscos	09/12/2014	Probabilidade	Alta	Média	Baixa				
		Impacto	Muito alto	Alto	Alto				
		Magnitude	Alta	Moderada	Moderada				
	09/02/2015	Probabilidade	Alta	Média	Baixa	Alta	Alta		
		Impacto	Muito alto	Alto	Alto	Alto	Alto		
		Magnitude	Alta	Moderada	Moderada	Alta	Alta		
	16/02/2015	Probabilidade	Alta	Média	Baixa		Alta	Alta	
		Impacto	Alto	Alto	Alto		Alto	Alto	
		Magnitude	Alta	Moderada	Moderada		Alta	Alta	
	05/03/2015	Probabilidade	Média	Média	Baixa		Alta	Alta	
		Impacto	Médio	Médio	Alto		Alto	Médio	
		Magnitude	Moderada	Moderada	Moderada		Alta	Moderada	
	30/03/2015	Probabilidade	Baixa	Baixa	Baixa		Média	Média	
		Impacto	Médio	Médio	Alto		Moderada	Médio	
		Magnitude	Baixa	Baixa	Moderada		Alta	Moderada	
	23/04/2015	Probabilidade			Baixa		Média		
		Impacto			Média		Médio		
		Magnitude			Baixa		Moderada		
	18/05/2015	Probabilidade					Baixa		Baixa
		Impacto					Baixo		Muito Alto
		Magnitude					Baixa		Moderada
	08/06/2015	Probabilidade							Baixa
		Impacto							Muito Alto
		Magnitude							Moderada

Tabela B-1 – Avaliação dos riscos durante o projeto

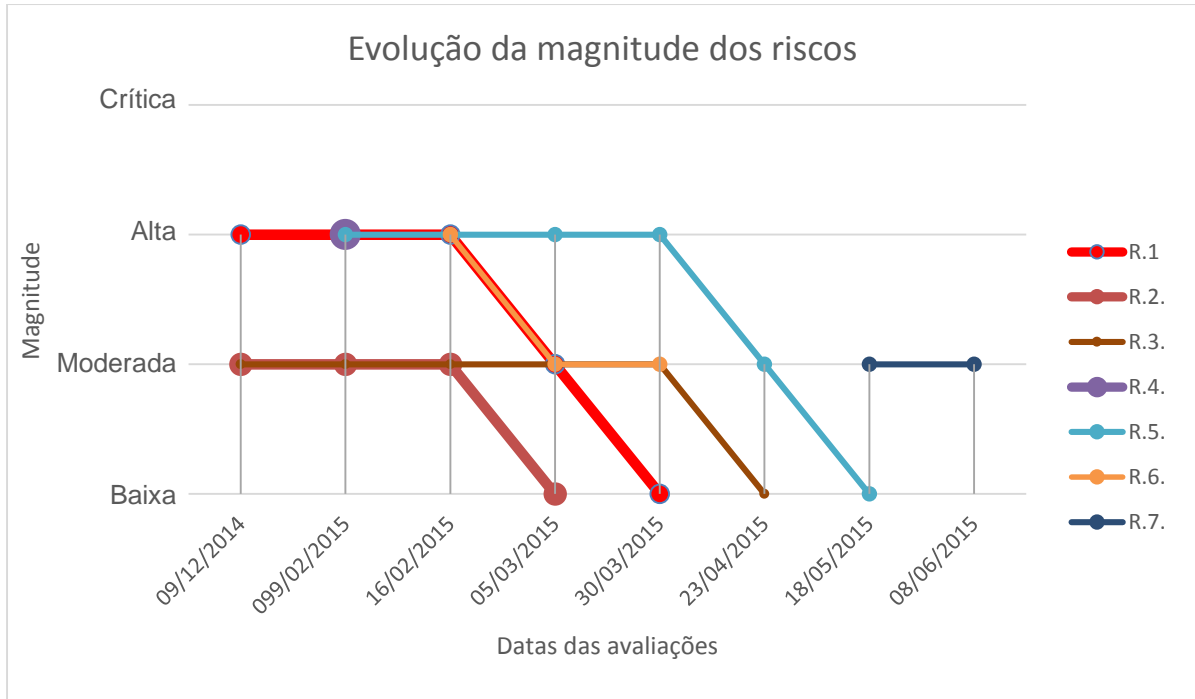


Figura B-1 - Evolução da magnitude dos riscos durante o projeto

Anexo C - Descrição dos modelos

Neste anexo são descritas todos os modelos, desenvolvidas durante o estágio. Para cada modelo é feita uma breve descrição do que armazena, seguindo-se da listagem dos seus atributos, indicando o tipo, se não é nulo e uma pequena descrição. Os últimos dois modelos apresentados já existiam na plataforma, no entanto é descrito os atributos que se adicionou.

1. *BackOffice*

Nome: PortalShout

Descrição: Este modelo representa uma mensagem da shoutbox de champions

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
user_id	false	integer	Identificador do utilizador autor da mensagem
portal_id	false	integer	Identificador do portal da mensagem
message	false	text	Mensagem
keep	false	boolean	Indicação se a mensagem deve ser guardado (true)
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

2. *Marketplace*

Nome: MarketArticle

Descrição: Este modelo representa a um artigo no marketplace

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
name	false	string	Nome do artigo
description		text	Descrição do artigo
thumbnail	false	string	URL da thumbnail do artigo
country_id		integer	Identificador do país onde se encontra localizado o artigo
price		float	Preço do artigo
seller_id		integer	Identificador do vendedor do artigo
category_id		integer	Identificador da categoria do artigo
status	false	integer	Estado do artigo
currency_id		integer	Identificador do tipo de moeda aceite no preço do artigo
crop_coors		string	Coordenadas do crop da thumbnail
rate		float	Taxa da comissão da C'B
total_notif		integer	Total de notificações do o vendedor sobre este artigo
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketBids

Descrição: Este modelo representa uma licitação no marketplace

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
buyer_id	false	integer	Identificador do comprador
article_id	false	integer	Identificador do artigo licitado
region_id	false	integer	Identificador da região onde o comprador pretende receber o artigo
expiration_date		datetime	Data de expiração
price		float	Preço da licitação

status	false	integer	Estado da licitação
shipping_coust		float	Valor dos portes de envio
total_notif	false	integer	Numero total de notificações do comprador sobre esta licitação
seller_notif		boolean	Indicação se esta licitação deve ser notificada no vendedor
buyer_notif		boolean	Indicação se esta licitação deve ser notificada no comprador
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketHistoryArticle

Descrição: Este modelo representa um dado histórico sobre um artigo

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
article_id	false	integer	Identificador do artigo em que pertence o dado histórico
status	false	integer	Estado em que o artigo se encontrava
seller_notif	false	boolean	Indicação se este dado histórico deve ser notificado no vendedor
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketHistoryBid

Descrição: Este modelo representa um dado histórico sobre uma licitação

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
bid_id	false	integer	Identificador da licitação em que pertence o dado histórico
status	false	integer	Estado em que a licitação se encontrava
buyer_notif	false	boolean	Indicação se este dado histórico deve ser notificado no comprador
seller_notif	false	boolean	Indicação se este dado histórico deve ser notificado no vendedor
seller_visible	false	boolean	Indicação se este dado histórico é visível ao vendedor
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketArticleImage

Descrição: Este modelo representa uma imagem da galeria de imagens de um artigo

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
article_id	false	integer	Identificador do artigo em que pertence a imagem
file	false	integer	URL da imagem
crop_coors		boolean	Coordenadas do crop da imagem
position		boolean	Posição da imagem na galeria
description		boolean	Descrição da imagem
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketArticleImage

Descrição: Este modelo representa uma imagem da galeria de imagens de um artigo

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
article_id	false	integer	Identificador do artigo em que pertence a imagem
file	false	string	URL da imagem
crop_coors		string	Coordenadas do crop da imagem
position		integer	Posição da imagem na galeria
description		text	Descrição da imagem
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketZone

Descrição: Este modelo representa uma zona do planeta

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
name	false	string	Nome da zona
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketShipping

Descrição: Este modelo representa um custo de envio de um artigo para uma zona

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
article_id	false	integer	Identificador do artigo à qual o custo de envio pertence
zone_id	false	integer	Identificador da zona à qual o custo de envio pertence
price	false	float	Preço do custo de envio
details		text	Detalhes do custo de envio
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketQuestion

Descrição: Este modelo representa uma questão e a respetiva resposta a um artigo do marketplace

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
user_question_id	false	integer	Identificador do utilizado que fez a pergunta
question	false	text	Questão ao artigo
answer		text	Resposta do vendedor à questão
article_id	false	integer	Identificador do artigo da questão
seller_notif		boolean	Indicação se deve ser notificado no vendedor
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketCurrency

Descrição: Este modelo representa um tipo de moeda

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
currency	false	integer	Nome do tipo de moeda
code	false	string	ISSO 4217 do tipo de moeda
visible	false	boolean	Indicação se o tipo de moeda é visível no marketplace
decimal	false	integer	Indicação do número de casas decimais aceites pelo PayPal
notes		string	Notas do PayPal sobre este tipo de moeda
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketPurchase

Descrição: Este modelo representa uma compra/venda no marketplace

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
seller_rating		boolean	Classificação do vendedor, avaliado pelo comprador (true – correu bem; false – correu mal; null – não foi avaliado)
seller_feedback		text	Feedback do vendedor, dado pelo comprador

buyer_rating		boolean	Classificação do comprador, avaliado pelo vendedor (true – correu bem; false – correu mal; null – não foi avaliado)
buyer_feedback		text	Feedback do comprador, dado pelo vendedor
receiver		string	Nome do recetor
city		string	Cidade do destino
status		boolean	Estado em que se encontra a compra/venda – true: finalizada; false - cancelada
article_id	false	integer	Identificador do artigo
bid_id	false	integer	Identificador da licitação
seller_id		integer	Identificador do vendedor
buyer_id		integer	Identificador do comprador
paykey		string	Código da compra/venda do PayPal
transaction_id		string	Identificador da transação entre o comprador e vendedor no PayPal
transaction_id_cb		string	Identificador da transação entre o vendedor e C'B no PayPal
country_id		integer	Identificador do país de receção do artigo
type		string	Tipo de transação efetuada no PayPal
postal_code		string	Código postal do destino
address_11		string	Endereço de destino – linha 1
address_12		string	Endereço de destino – linha 2
region		string	Região de destino
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: MarketWatchlist

Descrição: Este modelo representa um artigo que se encontra na watchlist de um utilizador

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
user_id	false	integer	Identificador do utilizador à qual pertence o artigo em watch
article_id	false	integer	Identificador do artigo em watch
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: Country

Descrição: Este modelo representa um país

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
name	false	string	Nome do país
alt_name		string	Nomes alternativos do país
iso2	false	string	Código ISO 3166-1 alfa-2 do país
iso3	false	string	Código ISO 3166-1 alfa-3 do país
un_region		string	Continente que o país pertence
un_sub_region		string	Parte do continente que o país pertence
lat		string	Latitude do país
lon		string	Longitude do país
created_at		datetime	Data da criação
updated_at		datetime	Data da última atualização

Nome: Region

Descrição: Este modelo representa uma subdivisão de um país

Atributo	Null?	Tipo	Descrição
id	false	integer	Identificador
name	false	string	Nome da região
country_id	false	integer	Identificador do país à qual a região pertence
code	false	string	ISO 3166-2 da subdivisão

rtype	string	Tipo da subdivisão (estado, região, distrito, etc)
alt_name	string	Nomes alternativos da subdivisão
created_at	datetime	Data da criação
updated_at	datetime	Data da última atualização

Nome: User

Descrição: Este modelo representa um utilizador da plataforma

Atributo	Null?	Tipo	Descrição
...
paypal_email		string	Email da conta PayPal
total_market_notif	false	integer	Numero total de notificações no marketplace

Nome: Category

Descrição: Este modelo representa uma categoria da plataforma

Atributo	Null?	Tipo	Descrição
...

Anexo D - Testes automatizados

Neste anexo são listados todos os testes automatizados implementados durante o estágio. As gems utilizadas para a implementação foram RSpec⁴¹, Capybara⁴² e FactoryGirl⁴³. É utilizado uma base de dados SQLite para o ambiente de testes, sendo possível configurar para outros tipos como por exemplo MySQL. Antes de correr os testes deve-se correr o seguinte comando para criar ou atualizar a base de dados de testes: `rake db:migrate RAILS_ENV=test`. Caso seja a primeira vez a utilizar a base de dados do ambiente de testes também se deverá popular com o seguinte comando: `rake db:seed RAILS_ENV=test`. Os testes encontram-se localizados na pasta “spec” da raiz do projeto e para executá-los todos de uma vez deverão ser executado o seguinte comando: `rspec spec/`. Se pretender executar apenas testes unitários ou de regreção ou mesmo um específico, deve-se colocar o diretório à frente do comando `rspec`. Se pretender exportar o resultado dos testes para um ficheiro HTML basta executar o seguinte comando: `rspec --colour -f html -o specs.html spec/`

1. Testes unitários

BackOffice

```

1 PortalShout
2   On creation
3     Should create a new shout given valid attributes
4     Should not create a new shout given invalid attributes
5   Scopes
6     Should return all my shouts
7     Should not return any shout
    
```

Marketplace

```

1 MarketArticle
2   Creation
3     Should create a new article given valid attributes
4     Should not create a new article given invalid attributes
5
6   Publish
7     Should publish the article given valid attributes
8     Should not publish the article given invalid attributes
    
```

⁴¹ <https://github.com/rspec/rspec-rails>

⁴² <https://github.com/jnicklas/capybara>

⁴³ https://github.com/thoughtbot/factory_girl

9	
10	Editing
11	Should edit an article given valid attributes
12	Should edit an article, adding 1 shipping cost given valid attributes
13	Should edit an article, deleting 1 shipping cost given valid attributes
14	Should not edit the article given invalid attributes
15	
16	Put in edit mode
17	Should put in edit mode an article given valid attributes
18	Should not put in edit mode an article given invalid attributes
19	
20	Cancel
21	Should cancel the article given valid attributes
22	Should not cancel the article given invalid attributes
23	
24	Reject Bid
25	Should reject the bid given valid attributes
26	
27	Dispatch Bid
28	Should dispatch the bid given valid attributes
29	Should not dispatch the bid given invalid attributes
30	
31	Evaluate buyer
32	Should evaluate the buyer given valid attributes
33	Should not evaluate the buyer given invalid attributes
34	
35	Increase notifications
36	Should increase notifications
37	
38	Remove notifications
39	Should remove notifications
40	
41	MarketShipping
42	Creation
43	Should create a new shipping given valid attributes
44	Should not create a new shipping given invalid attributes
45	
46	On destroying
47	Should destroy the shipping
48	
49	MarketWatchlist
50	Adding
51	Should add an article to the watchlist given valid attributes
52	Should not add an article to the watchlist given invalid attributes
53	
54	Removing
55	Should remove an article to the watchlist given valid attributes
56	Should not remove an article to the watchlist given invalid
57	attributes
58	
59	MarketBid
60	Creation
61	Should create a new bid given valid attributes (with negotiation)
62	Should create a new bid given valid attributes (without negotiation)
63	Should not create a new bid given invalid attributes
64	
65	Reception
66	Should confirm the reception given valid attributes
67	Should not confirm the reception given invalid attributes

68
69 Seller evaluation
70 Should evaluate the seller given valid attributes
71 Should not evaluate the seller given invalid attributes
72
73 Refusing
74 Should refuse the negotiation given valid attributes
75 Should not refuse the negotiation given invalid attributes
76
77 Expiration of a bid
78 Should expire the bid given valid attributes
79 Should not expire the bid given invalid attributes
80
81 Expiration (accepted bid)
82 Should expire accepted bid given valid attributes
83 Should not expire accepted bid given invalid attributes
84
85 Expiration (sent)
86 Should expire the article sent given valid attributes (status
87 completed)
88 Should expire the article sent given valid attributes (status
89 failed)
90 Should not expire article sent given invalid attributes
91
92 Expiration (purchase finished)
93 Should expire the purchase finished given valid attributes (status
94 received)
95 Should expire the purchase finished given valid attributes (status
96 failed)
97 Should not expire the purchase finished given invalid attributes
98
99 Increase notifications
100 Should increase notifications
101
102 Remove notifications
103 Should remove notifications
104
105 MarketPurchase
106 Creation
107 Should create a new purchase given valid attributes
108 Should not create a new purchase given invalid attributes
109
110 Put payment pending
111 Should put the payment pending given valid attributes
112 Should not put the payment pending given invalid attributes
113
114 Put purchase failed
115 Should put the purchase failed
116
117 Put payment concluded
118 Should put payment concluded
119

2. Testes de regressão

BackOffice

1	BackOffice navigation
2	Should ignore the submission of a search form when pressing the key Enter
3	
4	Series management
5	Should have a Series option in the menu
6	Should have a page for Series management
7	
8	Shoutbox
9	Check if the shoutbox is visible
10	Send a normal message on shoutbox 1 and check if was sent
11	Send a normal message on shoutbox 2 and check if was sent
12	Send a keep message and check if was sent
13	Check is the message send by other user appears on shoutbox 1

Marketplace

1	Public navigation
2	Check the marketplace option in the menu
3	Check if I cannot see details of a private article
4	Check if I can see details of a public article
5	View homepage and check if has the buyer and seller panels
6	Search an article on marketplace
7	Filter an article by category
8	Check if I cannot share on social networks an article
9	Check if I cannot bid an article
10	Check if I cannot add an article to the watchlist
11	Check if shows the option to filter my watchlist
12	
13	Marketplace navigation
14	Check the marketplace option in the menu
15	Check if I cannot see details of a public article
16	Check if I can see details of a public article
17	View homepage and check if has the buyer and seller panels
18	Search an article on marketplace
19	Filter articles by category
20	Choose how many lines I want to see in the items list
21	Navigate between pages
22	See seller rating of an article
23	See seller feedback of an offer
24	Check the seller anonymity of an article
25	
26	Insert articles into marketplace
27	Insert article by homepage
28	Copy one article of my collection to marketplace
29	
30	Edit and publish into marketplace
31	Insert details
32	Insert cover image by photos gallery
33	Delete an image gallery
34	Publish article
35	Check article dashboard
36	Put article in edit mode
37	Try to put an article in edit mode with bids

38
39 Cancel article
40 Cancel an article in the unpublished state
41 Cancel an article in the waiting for bids state
42 Cancel an article in the with offers state
43 Cancel an article in the accepted state
44 Should not be possible cancel an article paid
45
46 Questions and answers
47 Insert question in a public article
48 Insert question in a not public article
49 Check if the question without answer is not public
50 Insert answer to the question
51 Check if the question with answer is public
52
53 Watchlist
54 Add an article to the watchlist on the homepage
55 Try to add my article to the watchlist
56 Remove an article of the watchlist on the homepage
57 Add an article to the watchlist on the articles page
58 Remove an article to the watchlist on the article page
59 Show all articles of my watchlist
60
61 Bids
62 Seller actions
63 See the history of my article
64 See all offers of my article
65 See buyer rating and feedback of an offer
66 Check the anonymity of bidders
67 Reject an offer
68 Accept an offer
69 Cannot bid my article
70 Buyer actions
71 Bid an article with fixed price
72 Cannot bid an article with an active bid made by me
73 Bid an article with negotiable price
74 See the history of my bids about an article
75 Reject a accepted bid
76
77 Purchases, evaluation and feedback
78 Seller actions
79 Put an article as dispatched
80 Evaluate and give feedback of a purchase that was successful
81 Evaluate and give feedback of a failed purchase (without buyer
82 evaluation)
83 Evaluate and give feedback of a failed purchase (with buyer
84 evaluation)
85
86 Buyer actions
87 Put an article as received and evaluate, giving feedback
88 Evaluate and give feedback of a failed purchase (with seller
89 evaluation)
90 Evaluate and give feedback of a failed purchase (without seller
91 evaluation)

Anexo E - Análise dos valores das comissões dos concorrentes

Para ajudar a empresa decidir o valor da comissão a cobrar por transação no *marketplace*, selecionou-se cinco concorrentes que possuem *marketplace* e fez-se uma análise ao valor das comissões cobradas. As plataformas escolhidas foram: delcampe, Collectors Conner, todocoleccion, gemr e eBay. Esta ultima plataforma, apesar de não ser um concorrente direto, tem bastante importância nesta análise, pois é utilizada na implementação de muitos concorrentes diretos e é a plataforma mais popular a nível mundial de leilões e compra/venda entre utilizadores.

A tabela seguinte mostra o resultado desta análise. Os parâmetros de avaliação foram todos aqueles que de alguma forma, retiram uma percentagem ou um valor fixo sobre cada venda efetuada, sendo eles:






- Listagem – refere-se o preço por listagem na plataforma por artigo;
- Tempo de duração da listagem – indica quanto tempo o artigo estará visível ao público;
- Taxa sobre a venda – indica a percentagem retirada sobre cada venda efetuada;
- Métodos de pagamento – indica quais os métodos de pagamento aceites;
- Taxas no método de pagamento – indica se o valor das comissões cobradas por estes serviços, estão incluídos na taxa sobre a venda do artigo

Os valores colocados na tabela são todos dos serviços mais básicos de cada plataforma. Existem alguns serviços que não foram considerados nesta análise, como por exemplo se pagar uma subscrição mensal a taxa sobre a venda desce, entre outros. Para ter uma melhor visão sobre as diferenças, comparou-se uma venda de um artigo nos valores de \$50 e \$500, pressupõe que é a primeira venda de um utilizador em cada plataforma.

	eBay	delcampe	todocoleccion	Collectors Corner	gemr
Listagem	\$0 se for as primeiras 20 listagens do mês, senão \$0.30	-	-	-	\$0.30
Tempo de duração da listagem	1 mês	-	-	-	-
Taxa sobre a venda	10% com um máximo de \$750	Ver Figura E-1	< = 300€ - 6.5% >300€ - 2%	\$0.99 - \$50.00 - 5% \$50.01 - \$1000 - 4% > 1,000 → 2%	5%
Métodos de pagamento	PayPal, PayPal Credit, ProPay, Skrill, Credit card or debit card processed through the	PayPal, Skrill, Bank transfer, Domestic check, cash and credit card	PayPal	-	Stripe, PayPal

	seller's Internet merchant account, Payment upon pickup				
Taxas no método de pagamento incluídas?	Não	Não	Não	Sim	Não
Total de taxas sobre uma venda de \$50	\$5	\$2.93	\$3.25	\$2.5	\$2.8
Total de taxas sobre uma venda de \$500	\$50	\$27.68	\$10	\$20	\$25.30

Tabela E-1 - Comparação das comissões entre plataformas

Monthly sales amount	Commission
No sale	No commission
Under 750.00 USD/month	5.50% of the monthly sales amount + 0.18 USD per sold item (Club+  : 0,12 USD)*
750.00 USD - 1499.99 USD/month	5.00% x (sales amount - 750 USD) + 41.25 USD (commission lower level) + 0.18 USD per sold item (Club+  : 0,12 USD)*
1500.00 USD - 7499.99 USD/month	4.50% x (sales amount - 1500 USD) + 78.75 USD (commission lower levels) + 0.18 USD per sold item (Club+  : 0,12 USD)*
7500.00 USD - 14999.99 USD/month	3.00% x (sales amount - 7400 USD) + 348.75 USD (commission lower levels) + 0.18 USD per sold item (Club+  : 0,12 USD)*
15000.00 USD and beyond/month	2.00% x (sales amount - 15000 USD) + 573.75 USD (commission lower levels) + 0.18 USD per sold item (Club+  : 0,12 USD)*

Those rates are valid as of September 1, 2012.

*Until 01/01/2013, items put on sale before Sep, 1st 2012 won't be subject to the new rate of 0.18/0.12 USD.


Example: Last month, you sold 20 items for a total of 800.00 USD. The commission due at the beginning of the following month will be $0.05 \times (800-750) + 41.25 = 43.75 \text{ USD} + (0.18 \text{ USD} \times 20) = 47.35 \text{ USD}$ (46.15 USD for Club+ )

Figura E-1 - Valor das comissões na plataforma Collectors Corner

Referências:

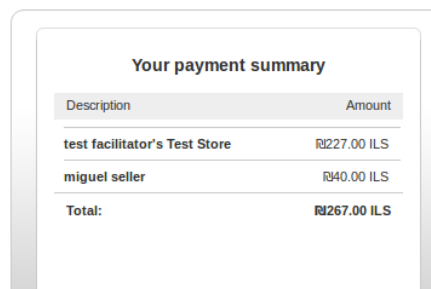
- Gemr - <https://blog.gemr.com/faq/>
- Collectors Conner - <http://www.collectorscorner.com/Help/>
- Todocoleccion - <http://en.todocoleccion.net/cuentasclaras.cfm>
- Delcampe - <http://www.delcampe.net/page/help/type,prices,language,E.html>
- Ebay - <http://pages.ebay.com/help/sell/fees.html>

Anexo F - Análise de soluções do PayPal

Para tentar perceber e ajudar a empresa a decidir qual o melhor método no PayPal, que permite à Collectors Bridge obter uma comissão nas compras e vendas dentro do seu *marketplace* e ao mesmo tempo utilizar a proteção do vendedor e comprador, fez-se uma análise à documentação do PayPal. Observou-se que a única API que permite fazer transações para múltiplos destinatários é a API Adaptive Payments. No entanto dentro desta API existem diversos modos para obter esse fim, sendo eles:

1. **Pagamentos paralelos** – O comprador paga à C'B e ao vendedor em simultâneo, no qual a API repartirá a percentagem a receber por recetor. Como se pode visualizar no exemplo de pagamento, presente na Figura F-1, o comprador irá ver os dois destinatários do pagamento e a percentagem de cada um deles. O vendedor quando abrir a sua conta PayPal verá apenas a transação que recebeu. Neste modo dá para utilizar a proteção PayPal do vendedor/comprador, configurando a obrigatoriedade do comprador colocar a morada de receção do artigo e que a venda é um objeto físico. Em relação às comissões por parte do PayPal, existe duas opções. Quem paga é o comprador ou é repartido entre o vendedor e a C'B (Figura F-2).

test facilitator's Test Store



The image shows a screenshot of a PayPal payment summary window. The window title is "test facilitator's Test Store". The main heading is "Your payment summary". Below this is a table with two columns: "Description" and "Amount". The table lists two items: "test facilitator's Test Store" for R\$227.00 ILS and "miguel seller" for R\$40.00 ILS. The total amount is R\$267.00 ILS.

Description	Amount
test facilitator's Test Store	R\$227.00 ILS
miguel seller	R\$40.00 ILS
Total:	R\$267.00 ILS

Figura F-1 – Janela do PayPal vista pelo comprador num ato de pagamento

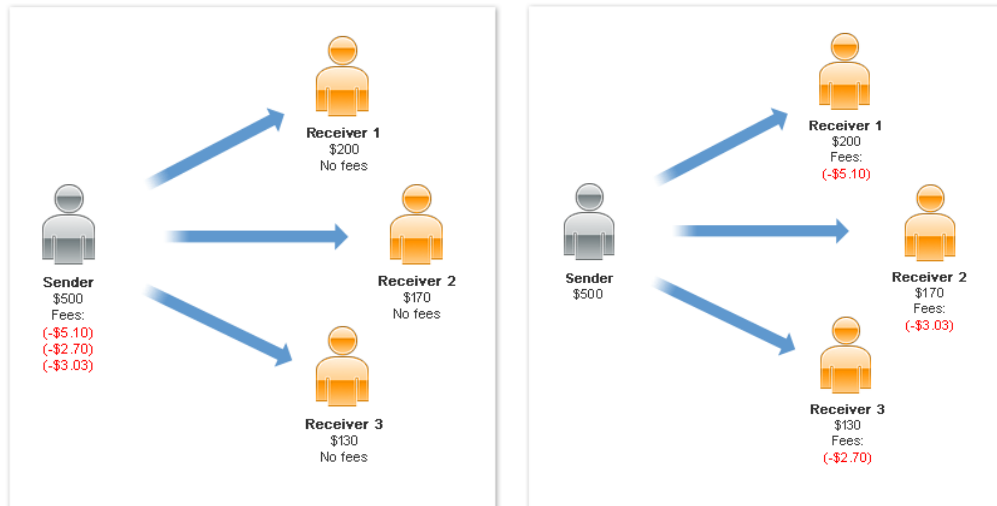


Figura F-2 – Opções de pagamentos de taxas PayPal

2. **Pagamentos acorrentados** – O pagamento acorrentado funciona da seguinte forma. O comprador paga a um recetor 1 e o recetor 1 paga aos outros recetores restantes. Esta segunda transação pode ser instantânea ou atrasada por um determinado tempo. Na situação da C'B o que se adequa melhor neste momento é a segunda transação ser instantânea e o comprador ser o primeiro recetor e a C'B ser o segundo recetor. A
3. Figura F-3 ilustra um exemplo de uma transação neste modo de funcionamento. É possível ver que o comprador apenas faz uma transação sendo e o vendedor quando recebe o dinheiro, este é automaticamente enviado para o recetor 2 (C'B). Neste modo dá para utilizar a proteção PayPal do vendedor/comprador, configurando a obrigatoriedade do comprador colocar a morada de receção do artigo e que a venda é um objeto físico. A situação da C'B ser o primeiro recetor não se adequa pois, esta tinha que ser responsável pela compra/venda ao utilizar a proteção do comprador/vendedor. Em relação às comissões por parte do PayPal, existe três opções. Quem paga é o vendedor, ou é o comprador ou é repartido por ambos.

Date	Type	Name/Email	Payment status	Details	Order status/Actions	Gross
Apr 27, 2015	Payment To	miguel seller	Completed	Details		-R\$40.00 ILS

Comprador:

Date	Type	Name/Email	Payment status	Details	Order status/Actions	Gross
27 Apr 2015	Payment To	test facilitator's Test Store	Completed	Details		-R\$10,00 ILS
27 Apr 2015	Payment From	test buyer	Completed	Details	Issue refund	R\$40,00 ILS

Vendedor:

Date	Type	Name/Email	Payment status	Details	Order status/Actions	Gross
Apr 27, 2015	Payment From	miguel seller	Completed	Details	Issue refund	R\$10.00 ILS

C'B:

Figura F-3 - Exemplo de transação no modo de pagamento acorrentado, sendo o vendedor o primeiro recetor e a C'B o segundo recetor.

Como foi demonstrado existem apenas duas opções que podem ser integradas na *marketplace* da Collectors Bridge, sendo a opção pagamento acorrentado com o vendedor a pagar as taxas do PayPal recomendado, pois estas taxas variam sendo mais fácil para a empresa gerir a percentagem da comissão a cobrar sem as mesmas. Os pagamentos paralelos não são recomendados, pois o comprador vê duas transações e não é possível o vendedor pagar todas as taxas do PayPal.

Referências:

- <https://developer.paypal.com/docs/classic/adaptive-payments/integration-guide/APIIntro/>
- <https://www.safaribooksonline.com/library/view/paypal-apis-up/9781449321666/ch04.html>

Anexo G - Manual de configuração do ambiente de trabalho

Este anexo contém um pequeno manual criado pelo estagiário, antes de começar o desenvolvimento do código. Faz parte dos documentos da empresa e tem como objetivo ajudar os próximos desenvolvedores da Collectors Bridge a configurar e instalar mais facilmente o ambiente de desenvolvimento.

Ruby and Rails Installation on Linux

Authors and Contributors:			
Name	Contact	Description	Date
Miguel Barcelos	mbarcelos@collectorsbridge.com	Author	11-02-2015
Pedro Martins	pmlmartins@collectorsbridge.com	Reviewer	12-02-2015

Revision History:			
Version	Date	Description	Author
01	11-02-2015	Document creation	Miguel Barcelos
02	12-02-2015	Document review	Pedro Martins
03	25-05-2015	Document update with the new features of the system	Miguel Barcelos

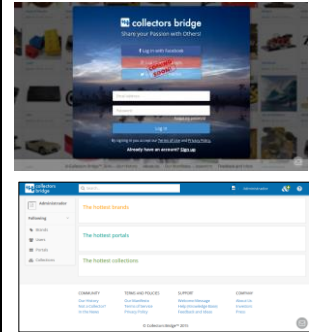
Objective
The purpose of this procedure is install Ruby and Rails on Linux (Ubuntu) and run the Collectors Bridge code in your local machine to develop.

Overview
<p>This document consists of four parts. The first is a list of steps to prepare your system. The second part explains how to install the Ruby using an RVM. The third part explains how to install MySQL and ImageMagic. The fourth part explains how to install Rails, as well as all the necessary gems, and finally, it is explained how to test C'B code on the local server.</p> <p>The versions used in the construction of tutorial were:</p> <ul style="list-style-type: none"> • Xubuntu 14.01 • Ruby: 2.1.2 (x64)

Task Description	Output
<p>Part 1 – Prepare Your System</p> <ol style="list-style-type: none"> 1. Update your package manager: \$ sudo apt-get update 2. Install Curl: \$ sudo apt-get install curl 3. Add this address to the hosts file: 127.0.0.1 local.collectorsbridge.com \$ sudo vi /etc/hosts 4. Install libqt4-dev: \$ sudo apt-get install libqt4-dev 	<p>Package manager updated</p> <p>Curl installed</p> <p>Address added</p> <p>libqt4-dev installed</p>
<p>Part 2 – Install Ruby Using RVM (Ruby Version Manager)</p> <ol style="list-style-type: none"> 1. Install the RVM and Ruby 2.1.2 (mandatory version) \$ \curl -L https://get.rvm.io bash -s stable --ruby-2.1.2 	<p>RVM and Ruby 2.1.2 installed</p>
<p>Part 3 – Install MySQL and ImageMagick</p> <ol style="list-style-type: none"> 1. Install MySQL \$ sudo apt-get install mysql-server mysql-client \$ sudo apt-get install libmysqlclient-dev 2. Install ImageMagick \$ sudo apt-get install imagemagick 	<p>MySQL installed</p> <p>ImageMagick installed</p>
<p>Part 4 – Install all gem’s (Rails included), download C’B code and create the database</p> <ol style="list-style-type: none"> 1. Go to the project root of Collectors Bridge (You can get the code here: https://bitbucket.org/collectorsbridge/collectorsbridge/downloads) 2. \$ chmod u+x app/clockwork.rb 3. \$ chmod u+x bin/delayed_job.rb 4. \$ bundle install 5. \$ rake db:migrate 6. \$ rake db:seed 	<p>Permissions granted</p> <p>All gem’s installed Database created and populated</p>
<p>Part 5 – Start the server and test C’B code</p> <ol style="list-style-type: none"> 1. \$ RAILS_ENV=development bin/delayed_job start 2. \$ RAILS_ENV=development rake websocket_rails:start_server 3. \$ server s 	<p>Delayed job started</p> <p>Websockets started</p> <p>Server started</p>

4. Open a new command line, go to the project root
5. `$ RAILS_ENV=development bundle exec clockworkd -c bin/clockwork.rb start`
6. Open one web browser <http://localhost:3000/> and authenticate with:
Email - admin@admin.com
Pass - admin

Clockwork started



Anexo H - Protótipos não funcionais



Protótipos Não Funcionais

Autor: Miguel Barcelos

Versão: 1.0

12/01/2015

1. Introdução

Este documento foi o resultado da terceira e última fase (*prototyping*) da metodologia *Design Thinking*, aplicada à plataforma Collectors Bridge no contexto do projeto de estágio. A fase *prototyping* consiste em implementar ideias recolhidas na fase *brainstorming*, de forma iterativa num modelo preliminar, possibilitando a avaliação de uma determinada abordagem a partir da qual poderão sugerir novas ideias consequentes (Figura H-1).

A ferramenta utilizada para a elaboração dos *mockups* foi a Balsamiq Mockups e a partilha e feedback dos mesmos foi feita apenas aos elementos da empresa Collectors Bridge.

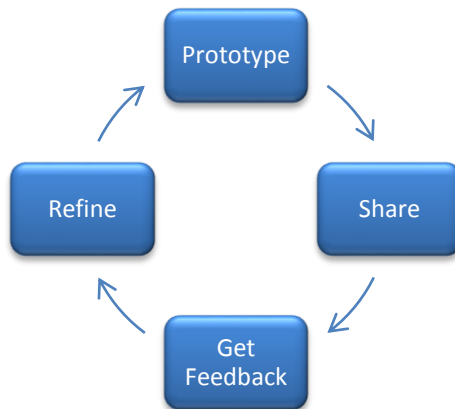


Figura H-1 - Processo iterativo do *prototyping*

2. Módulo de Publicidade v2

Contexto e problema: Atualmente a Collectors Bridge possui apenas uma forma de inserir publicidade, sendo esta manual e de inserção exclusiva da empresa (pelo *back-office*). É possível configurar a publicidade para que apareça em diversos lugares da plataforma como por exemplo a página inicial, página das *Brands*, *Portals*, Showcases entre outros. A disputa de publicidade num determinado lugar é feita aleatoriamente.

O principal problema da solução atual é a escalabilidade, pois todo o processo precisa passar pela empresa.

Objetivos: Pretende-se desenvolver um módulo que permita automatizar todo processo mencionado anteriormente, sendo os principais objetivos:

- Possibilidade de uma marca ou utilizador inserir publicidade de forma automática na plataforma;
- Possibilidade de monitorização pelos autores de cada publicidade;
- Implementar um algoritmo de disputa de publicidade.

Restrições: As preferências do utilizador não devem influenciar a mostragem da publicidade nos portais e *brands*.

Solução proposta:

Será possível fazer publicidade desde que se tenha conta na plataforma (*user* ou *brand*) e a opção de fazer publicidade irá estar sempre visível no menu do utilizador (Figura H-2).

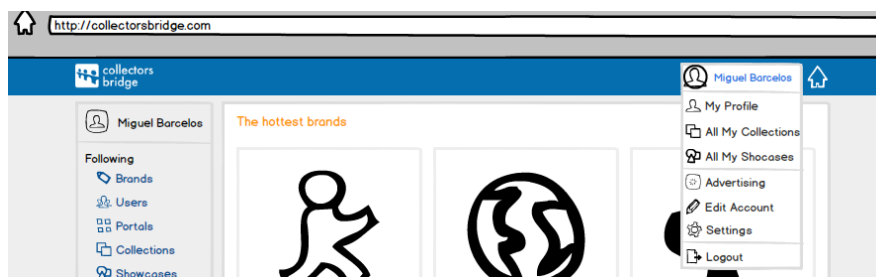


Figura H-2 - Link de Advertising no menu do utilizador

Após selecionar a opção “Advertising” será possível criar uma nova publicidade ou monitorizar uma já existente (Figura H-3).

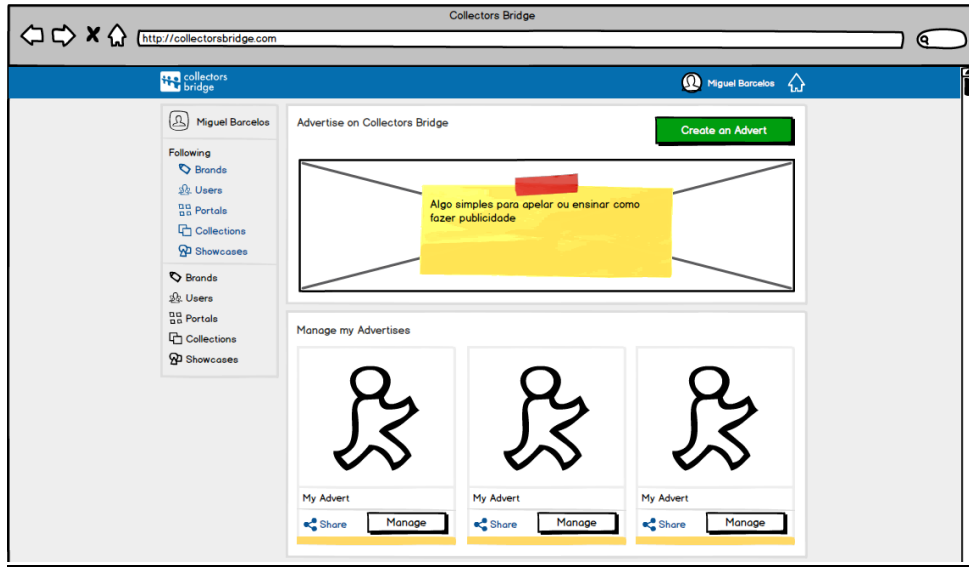


Figura H-3 - Página de Advertising

Ao selecionar a opção de criar um novo anúncio aparecerá uma janela onde permitirá criar um novo anuncio em 2 passos:

- 1) O utilizador preenche um formulário, no qual irá ver um *preview* do anuncio, escolherá a forma (por impressão ou clique), o *budget* a dar (por dia ou por intervalo de tempo) e o/os locais a aparecer o anuncio. Se selecionar 2 imagens resultará de 2 anúncios diferentes onde irá ser partilhado o *budget* (Figura H-4).
- 2) O utilizador efetua a compra e poderá escolher diferentes métodos de pagamento (Figura H-5).

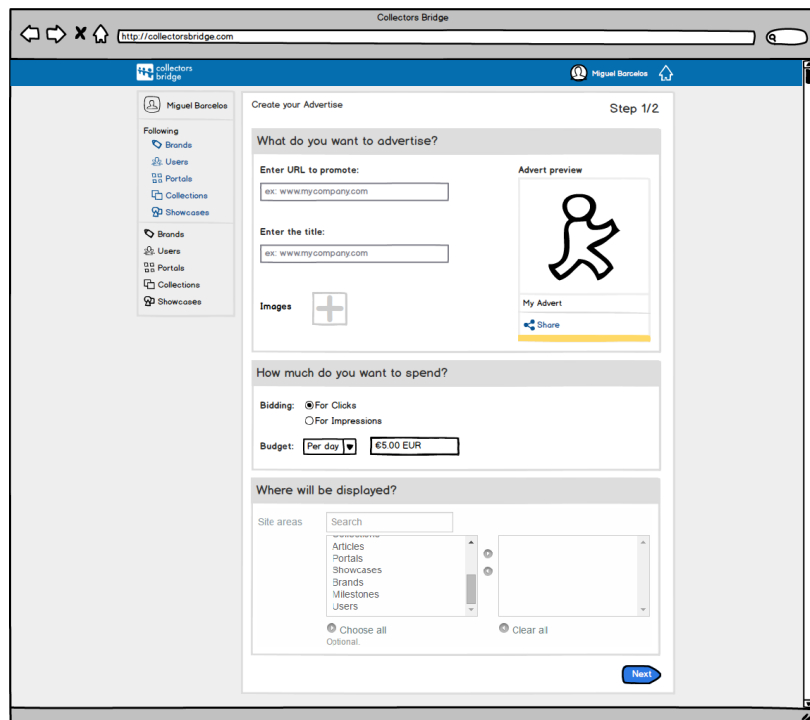


Figura H-4 - Passo 1 para adicionar um novo advertise

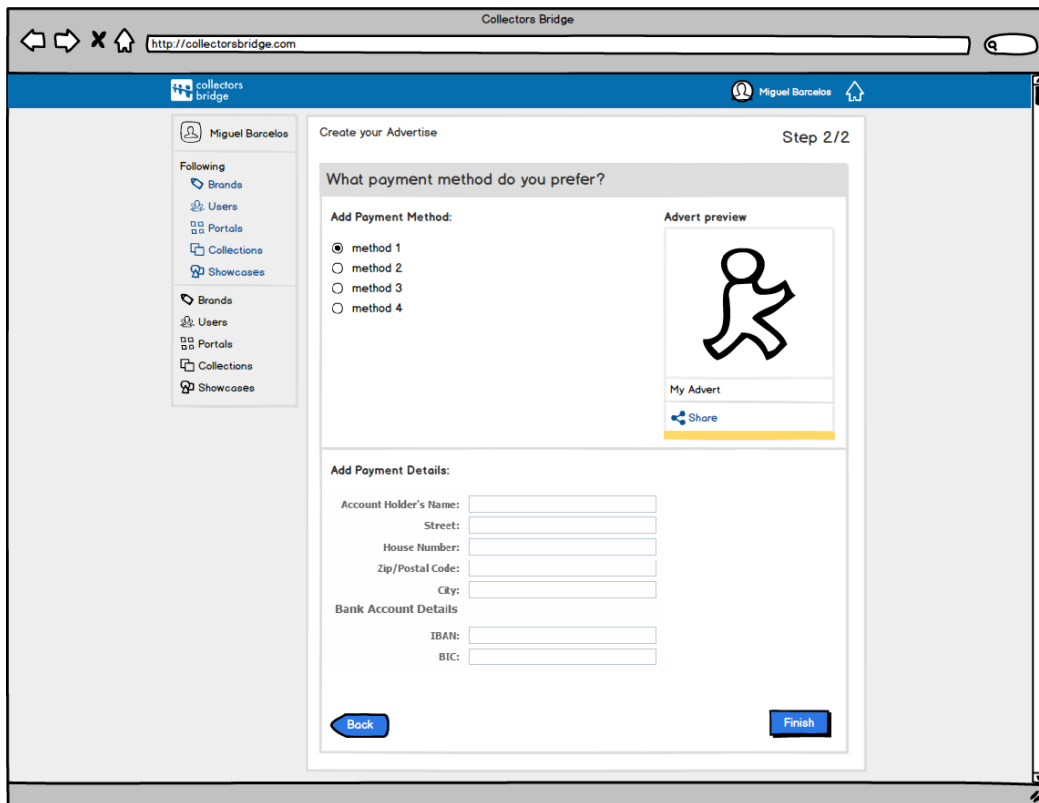


Figura H-5 - Passo 2 para adicionar um novo *advertise*

Ao ter um anúncio criado o utilizador poderá ver a estatística dele (por exemplo numero de *views*, cliques, etc.):

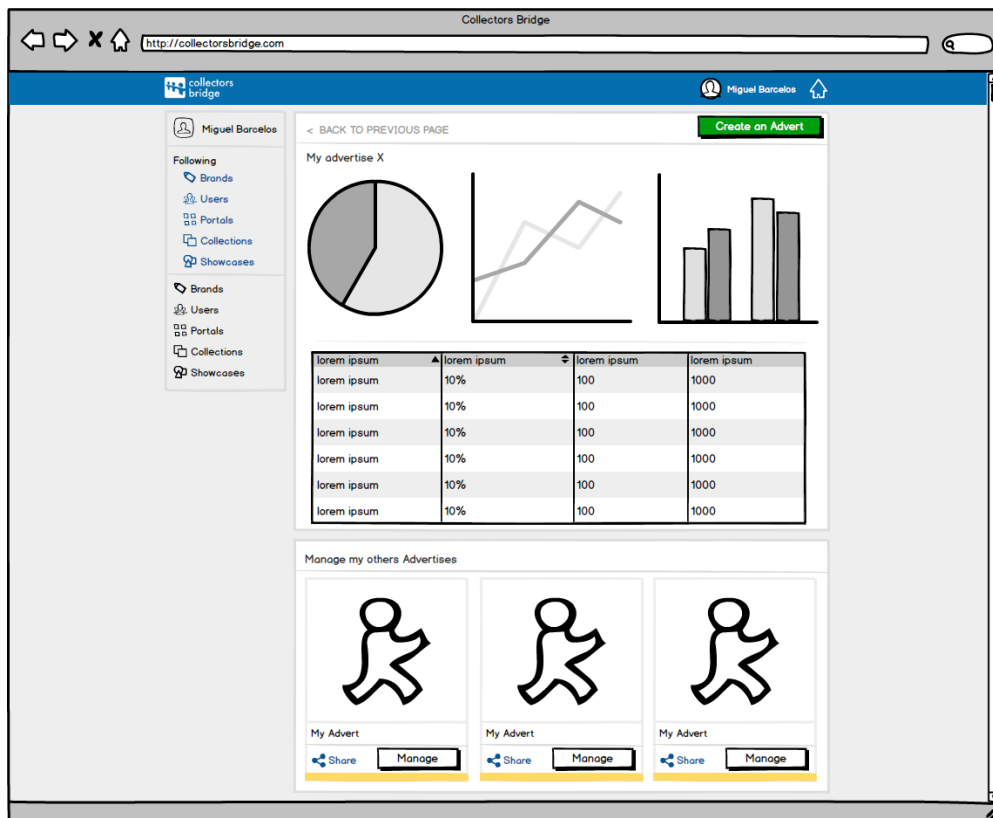


Figura H-6 - Painel de monitoramento de um *advertisse*

3. Módulo Marketplace v1

Contexto e problema: Um dos fatores críticos de sucesso da plataforma da Collectors Bridge encontra-se num *marketplace* que possibilite os seus utilizadores transacionarem artigos entre si. Atualmente esta funcionalidade é inexistente na plataforma.

Objetivos: Pretende-se contruir uma primeira versão de um módulo designado de Marketplace que tem como objetivos:

- Possibilidade do utilizador criar artigos para venda;
- Possibilidade do utilizador passar um artigo de uma coleção para venda;
- Possibilidade de um utilizador comprar artigos;
- Anonimato do comprador e vendedor;
- Perguntas e respostas sobre um determinado artigo;
- O preço dos itens é fixo pelos vendedores, podendo ser negociável pelos compradores;
- Pagamentos com PayPal;
- Pesquisa de artigos.

Solução proposta:

Será adicionado uma nova opção no menu lateral esquerdo, designado por “Marketplace” que incorporará o número de notificações referentes aos itens aos artigos em processo de compra e à venda por parte do utilizador. Na página inicial do *marketplace*, o utilizador poderá ter uma visão geral do estado dos seus artigos à venda e das suas compras. Poderá também adicionar novos artigos ao *marketplace* e visualizar artigos à venda, sendo possível a pesquisa e aplicação de filtros sobre os mesmos (Figura H-7). Para adicionar um novo artigo para venda, surgirá um *popup* com um formulário, em que o utilizador deverá preencher (Figura H-8). Se o utilizador quiser transferir um artigo para venda, pertencente a uma coleção, deverá carregar numa nova opção de “colocar artigo para venda”, na vista de edição de artigos. Irá surgir o mesmo *popup* anterior, mas desta vez estará parcialmente preenchido.

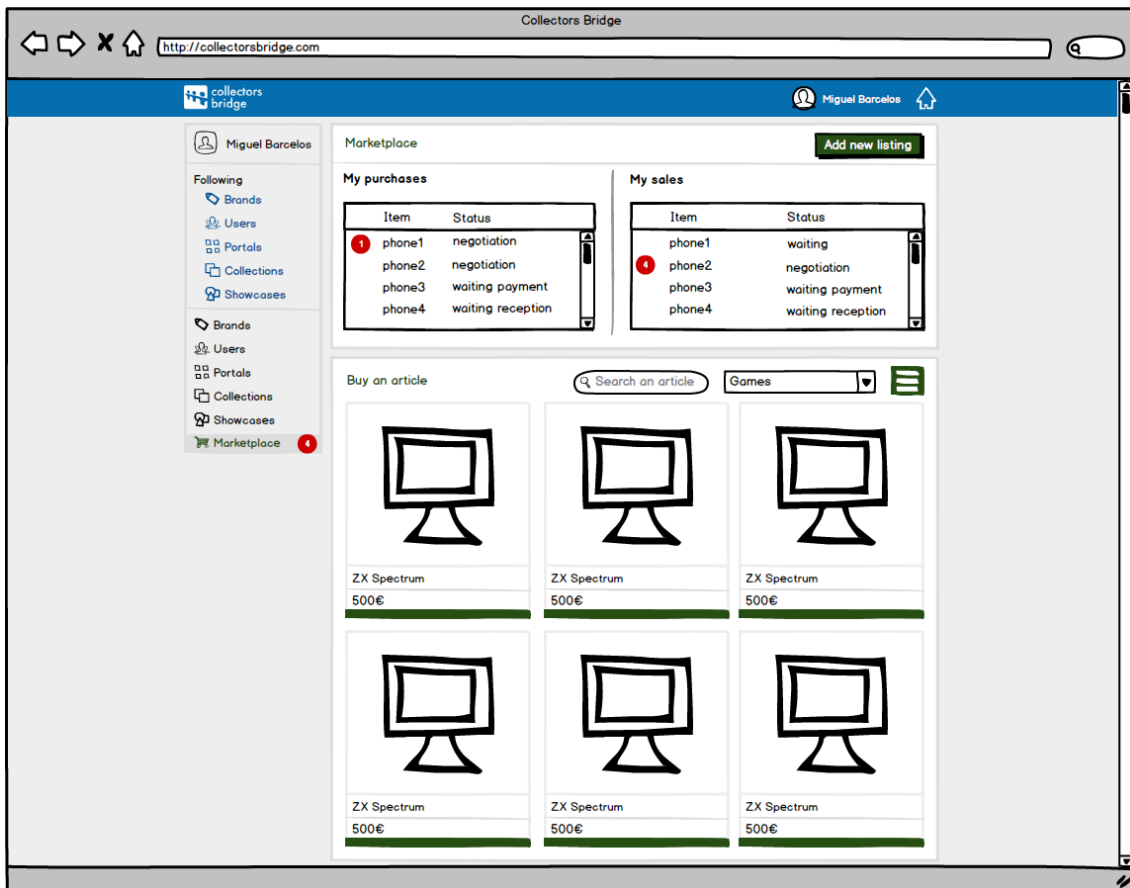


Figura H-7 - Página inicial do *Marketplace*

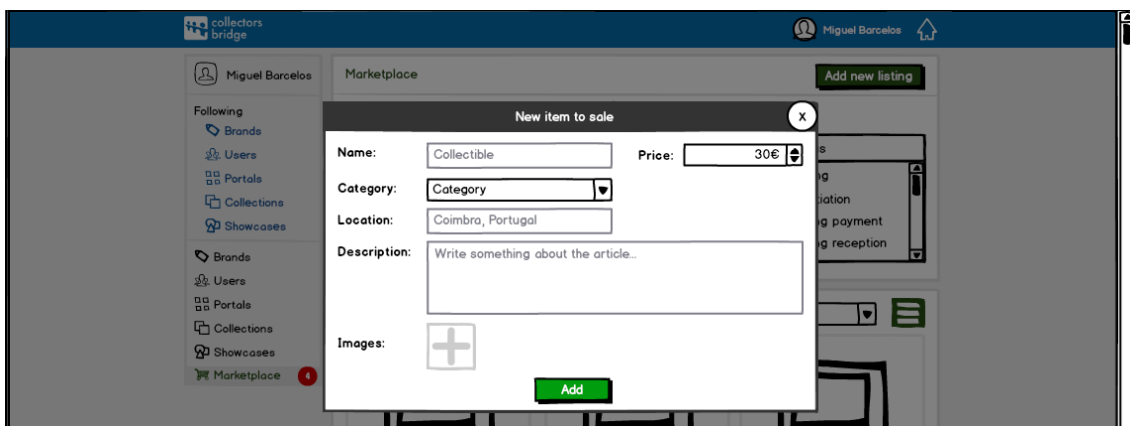


Figura H-8 - *Popup* de adição de um artigo ao *Marketplace*

Ao selecionar um artigo do *marketplace* irá ser mostrado detalhes, como por exemplo o nome, uma descrição, fotografias, preço, localização e o *rating* do utilizador que o está a vender. Também é possível aos utilizadores fazerem perguntas de forma pública ao vendedor sobre o artigo à venda (**Erro! A origem da referência não foi encontrada.**). Ao carregar em comprar, surgirá um *popup*, no qual o utilizador poderá aceitar o preço do artigo ou sugerir um novo valor, e colocará o país e cidade em que pretende receber, para que o vendedor possa dizer o valor dos portes de envio (Figura H-10 e Figura H-11).

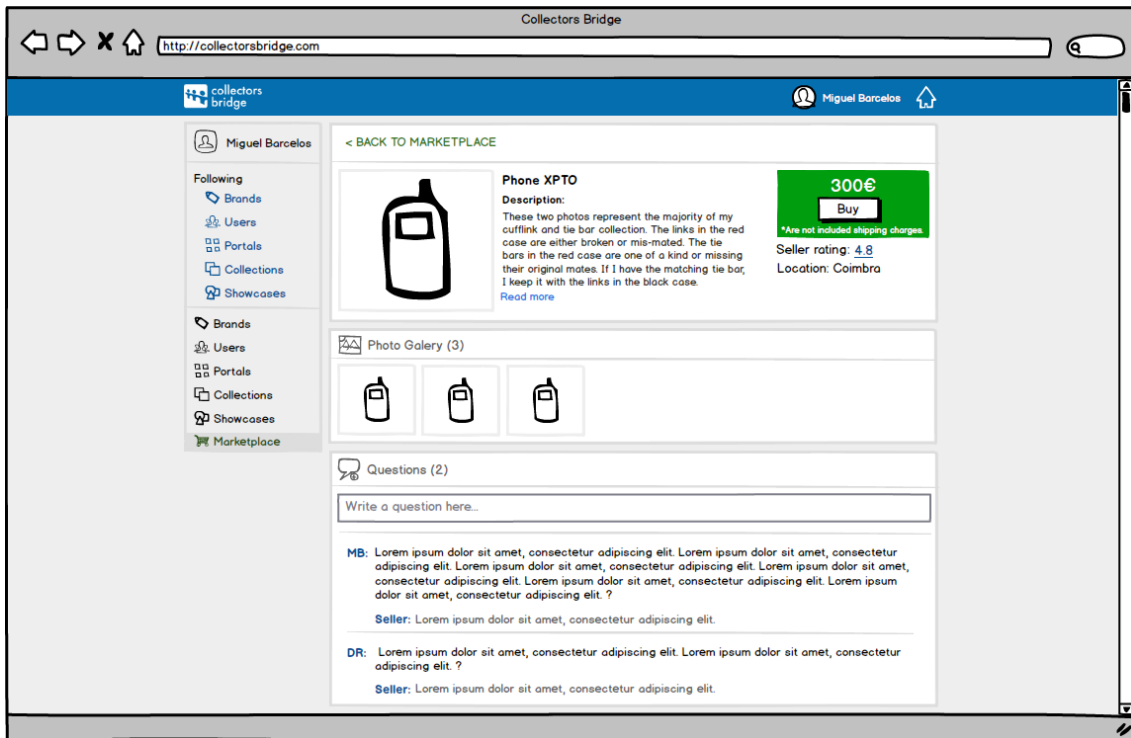


Figura H 9 - Página de um artigo à venda



Figura H-10 - Primeiro passo da compra por parte do comprador

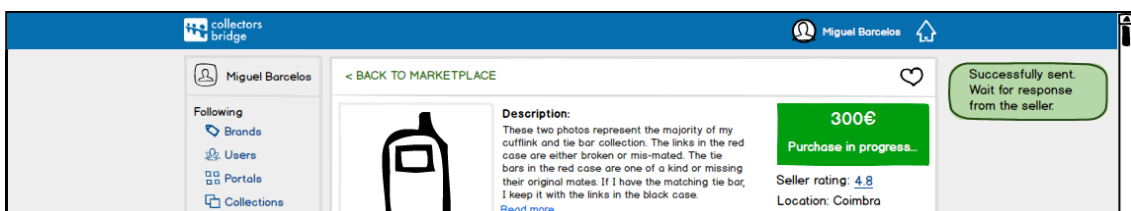


Figura H-11 - Confirmação do primeiro passo da compra por parte do comprador

Sempre que houver uma novas notificações sobre um artigo à venda pelo utilizador, estas podem ser consultadas na página do artigo (Figura H-12). Nesta página o utilizador poderá responder às perguntas sobre o artigo, ver o histórico das propostas rejeitadas, apagar o artigo (caso este não se encontre pago pelo comprador), e ver o estado atual da negociação. Na Figura H-12 o artigo está no primeiro passo da negociação, ou seja, a visualização da proposta, no qual deverá rejeitar ou atribuir um valor de portos de envio e aceitar. Poderá haver várias propostas, mas o utilizador só conseguirá ver exclusivamente uma (a mais antiga). Para ver

outras, ele deverá rejeitar a proposta presente. Na Figura H-13 é possível ver uma proposta já aceite e paga pelo comprador. O utilizador deverá enviar o artigo para o endereço apresentado no ecrã. Por fim na Figura H-14, é indicado que o comprador já recebeu o artigo, e o vendedor deverá classificar o comprador para que o processo seja encerrado.

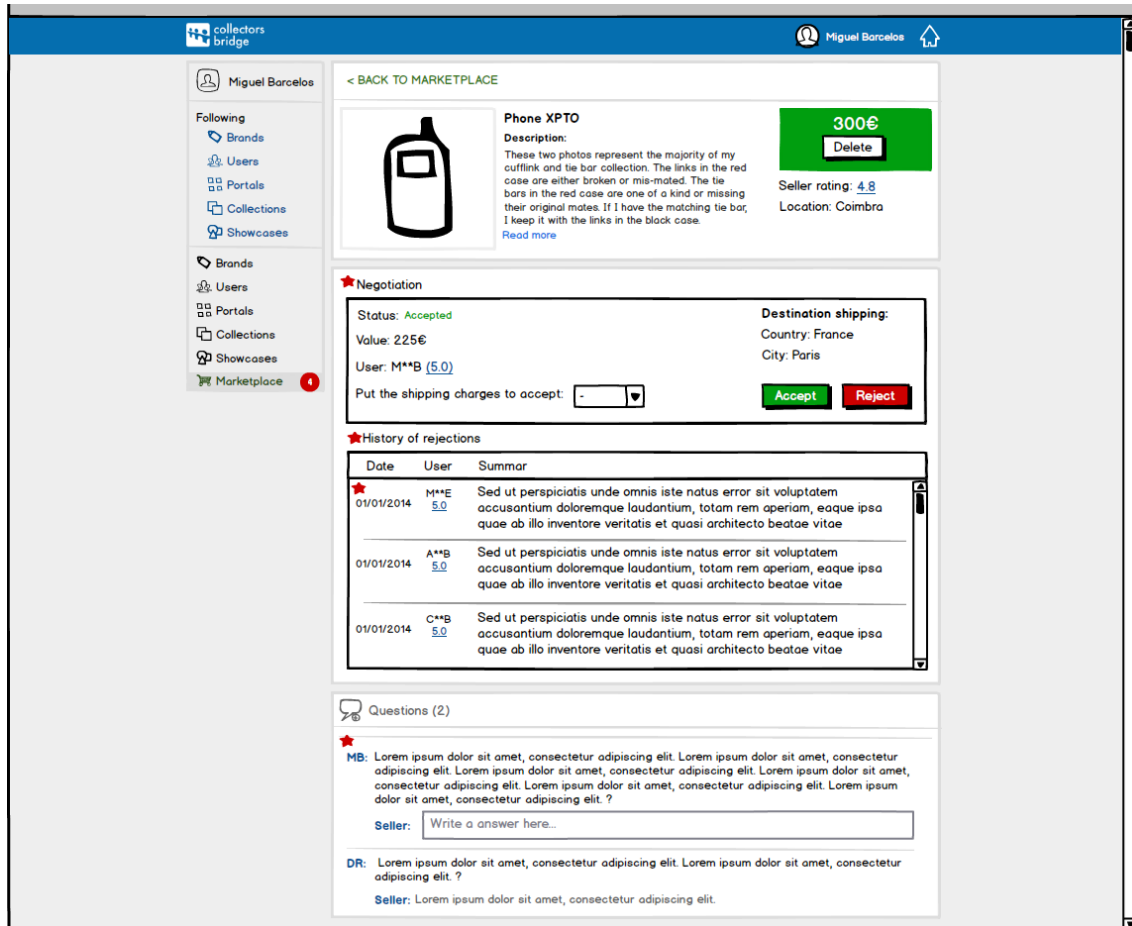


Figura H-12 - Página de um produto colocado à venda, vista pelo seu vendedor

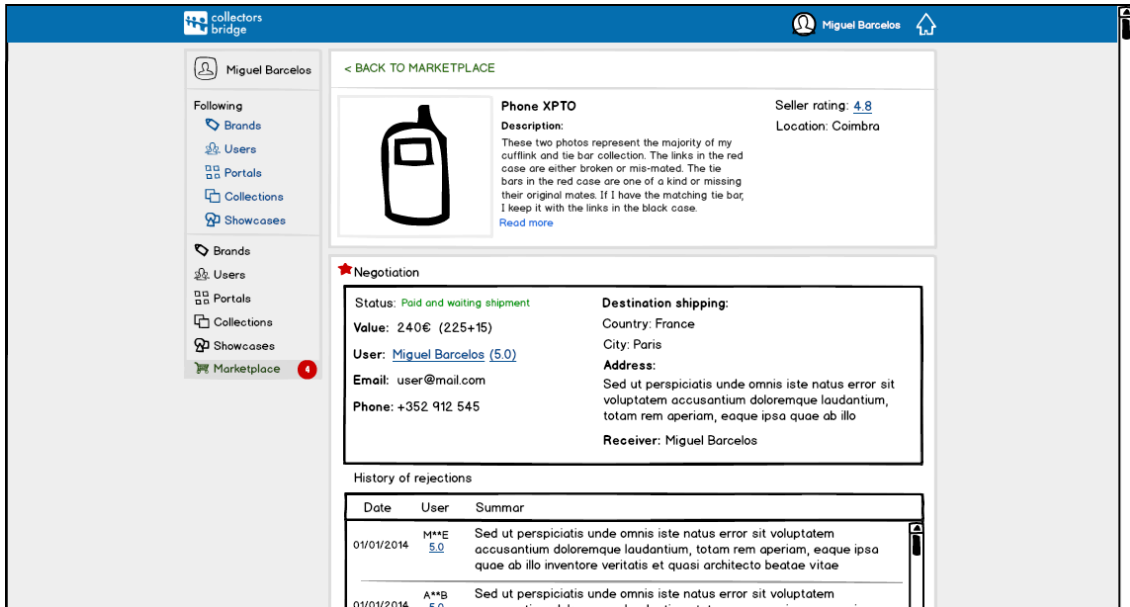


Figura H-13 - Artigo pago pelo comprador, à espera do envio por parte do vendedor (vista do vendedor)

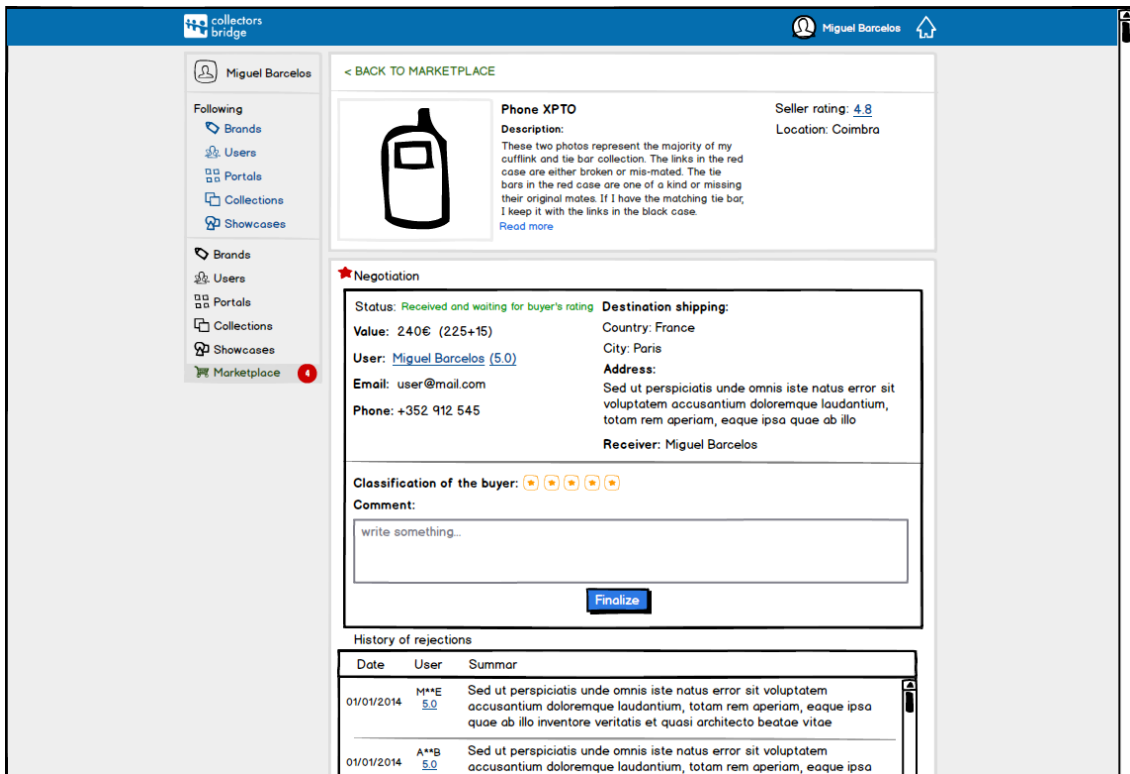


Figura H-14 - Último passo do processo de compra/venda de um artigo (classificação dos dois utilizadores)

Ao ser aceite a proposta pelo vendedor, o comprador poderá rejeitar ou fazer o checkout (Figura H-15). Se fizer checkout, surgirá um *popup* (Figura H-16) para preencher os dados da entrega e fazer o pagamento. Por fim, ao receber o artigo, o comprador deverá dizer que já recebeu e classificar o vendedor para que o processo seja concluído (Figura H-14).

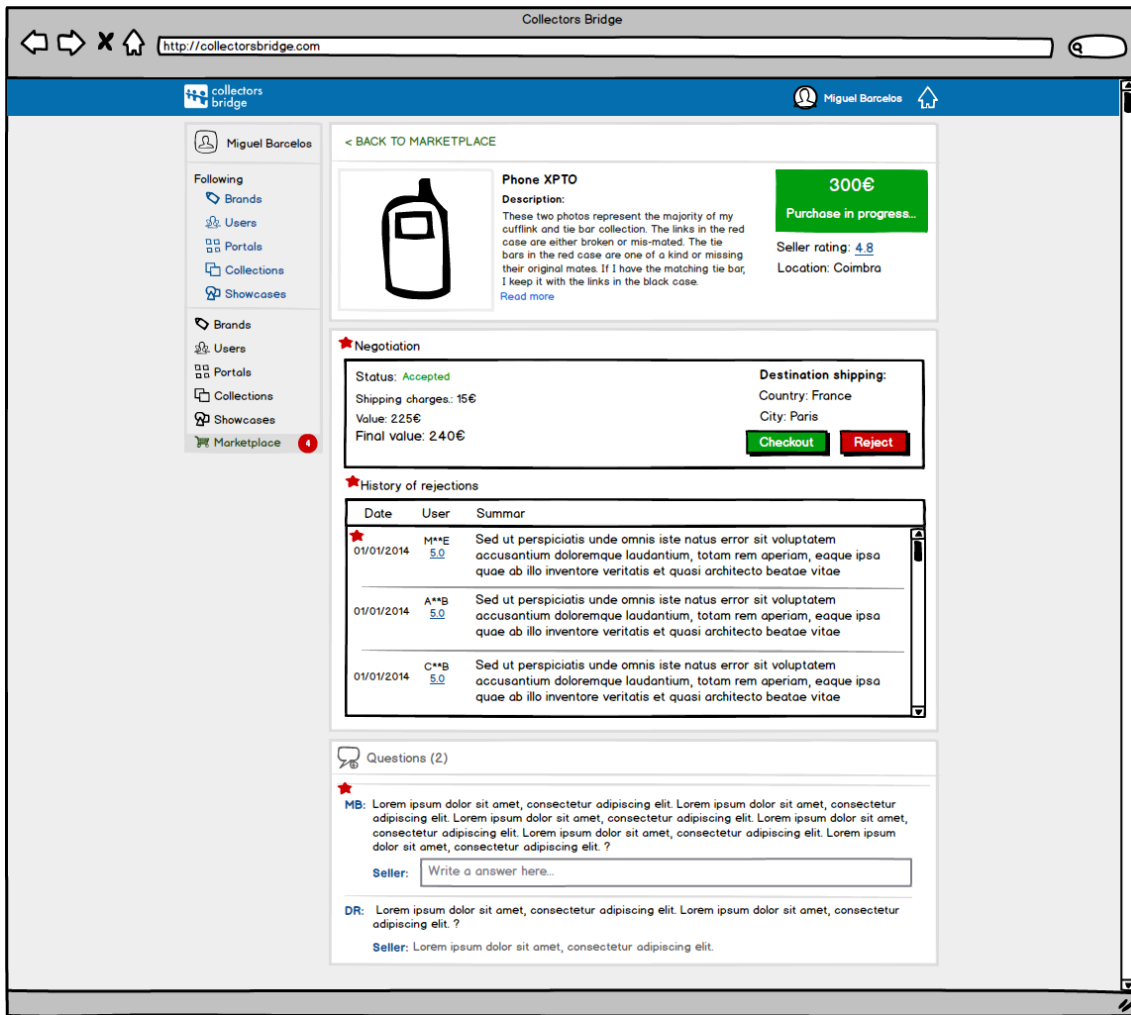


Figura H-15 - Preço aceito pelo vendedor (visto pelo comprador)

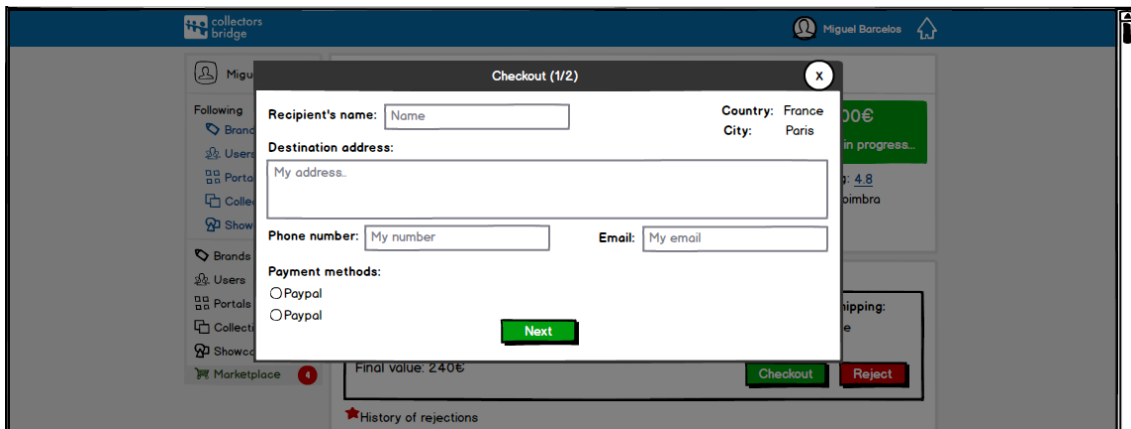


Figura H-16 - Popup de checkout de artigo

4. Módulo *Mystery* v1

Contexto e problema: Sendo a parte educativa um fator crítico de sucesso para plataforma Collectors Bridge, é necessário fomentá-la.

Atualmente muitos colecionadores e não colecionadores possuem objetos no qual gostariam saber mais informações sobre os mesmos. Este problema pode ser observado em alguns grupos dedicados ao colecionismo no Facebook, no qual os membros questionem a comunidade sobre informações de um determinado colecionável.

Objetivos: Pretende-se desenvolver um módulo que permita a qualquer utilizador da plataforma inserir um objeto, sabendo ou não a que portal pertence, questionando a comunidade com objetivo de saber informações sobre o mesmo. Ao ter uma resposta conclusiva sobre o objeto, o utilizador proprietário ou *Champion* de um portal deverá dar como resolvido a questão colocada.

Solução proposta:

Será acrescentado uma opção no menu lateral designada de “Mystery”, sendo um *link* para página inicial deste tema. Na página *Mystery* (Figura H-17) será possível ver todos os colecionáveis, havendo uma divisão de objetos já identificados e em identificação. Será possível pesquisar, adicionar um filtro ou mudar a ordenação da listagem (Figura H-18).

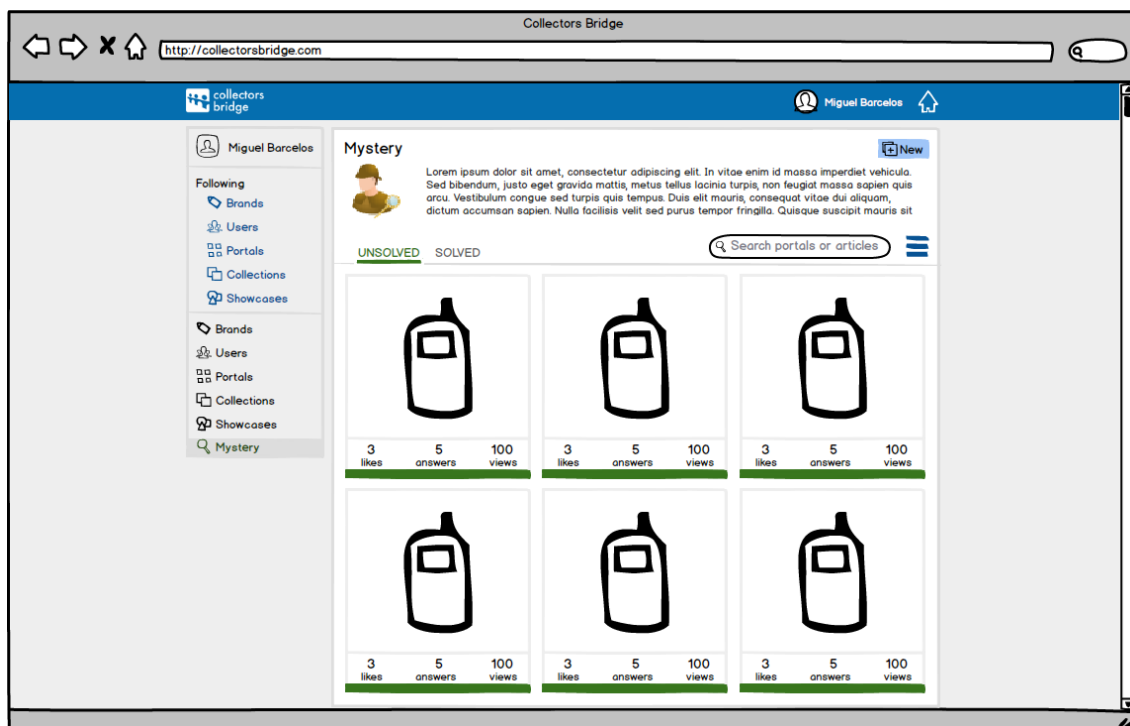


Figura H-17 - Página inicial do *Mystery*

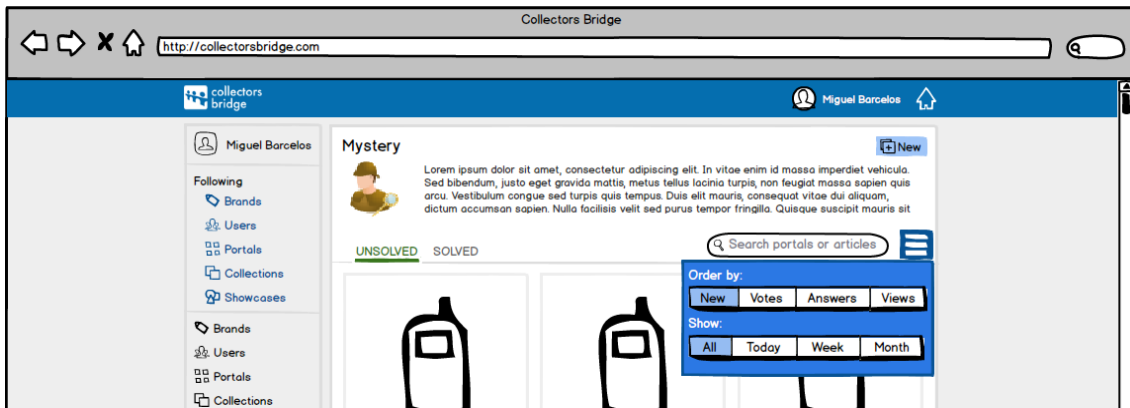


Figura H-18 - Exemplo do filtro a aplicar à listagem dos itens

Para adicionar um novo item basta clicar no botão “New” no qual aparecerá uma *popup* onde o utilizador irá inserir as informações relativamente ao colecionável pretendido (Figura H-19).

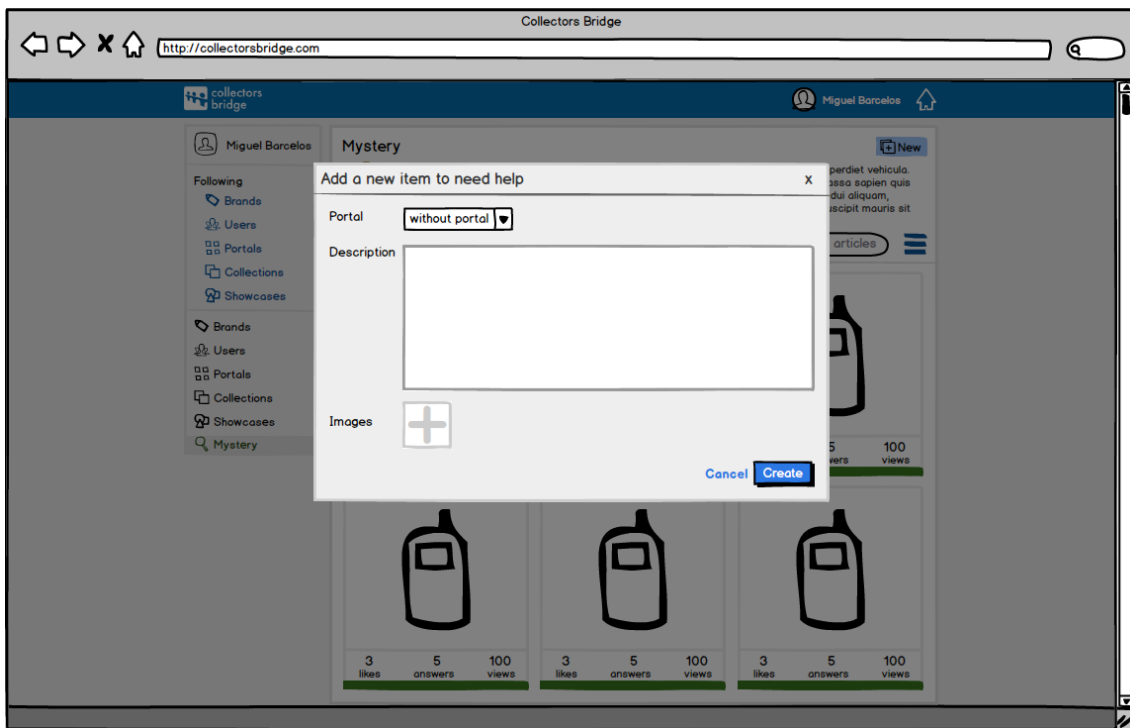


Figura H-19 - Inserção de um novo colecionável no *Mystery*

Na página relativa a um colecionável do *Mystery* será possível visualizar as fotos, a questão do autor, responder e votar nas respostas que acha mais corretas (Figura H-20). Ao estar no estado resolvido irá aparecer a resposta com mais *likes* no topo destacada por um ícone. Poderá também haver uma resposta destacada, selecionada pelo autor do artigo ou *champion* (Figura H-21).

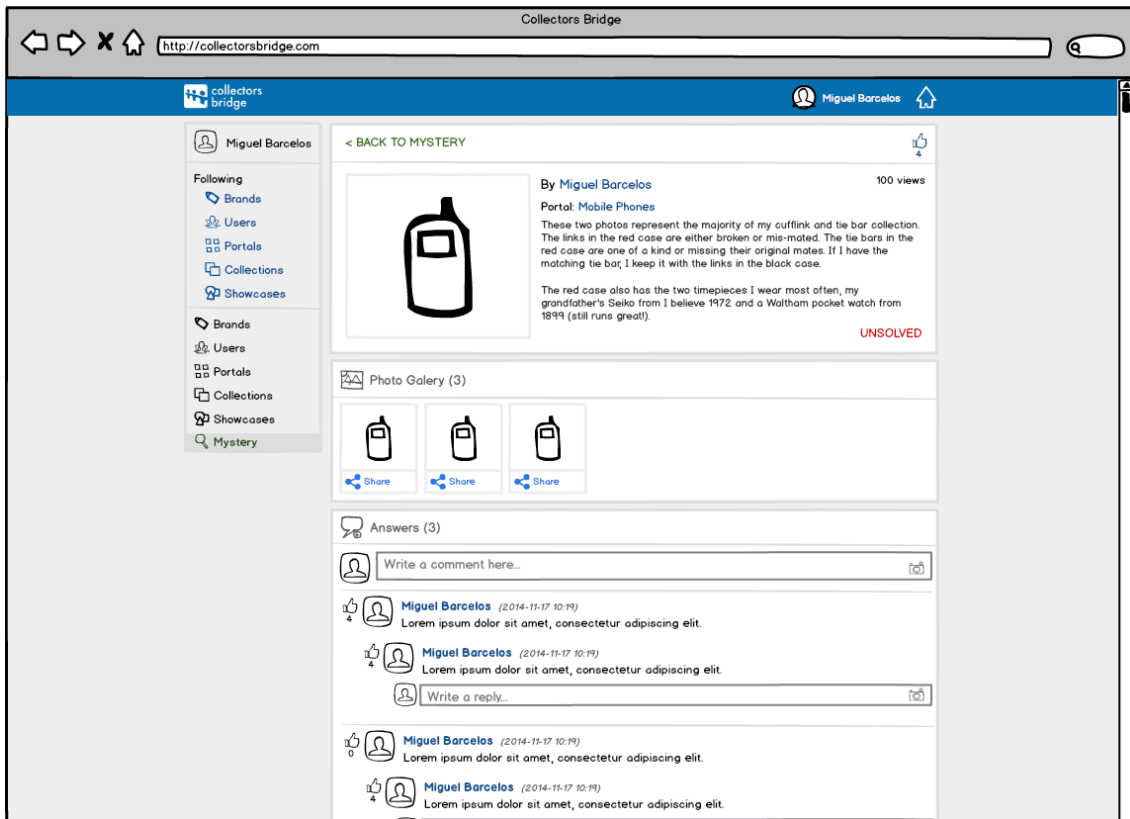


Figura H-20 - Página relativa a um colecionável do *Mystery* ainda não resolvido (vista por um utilizador)

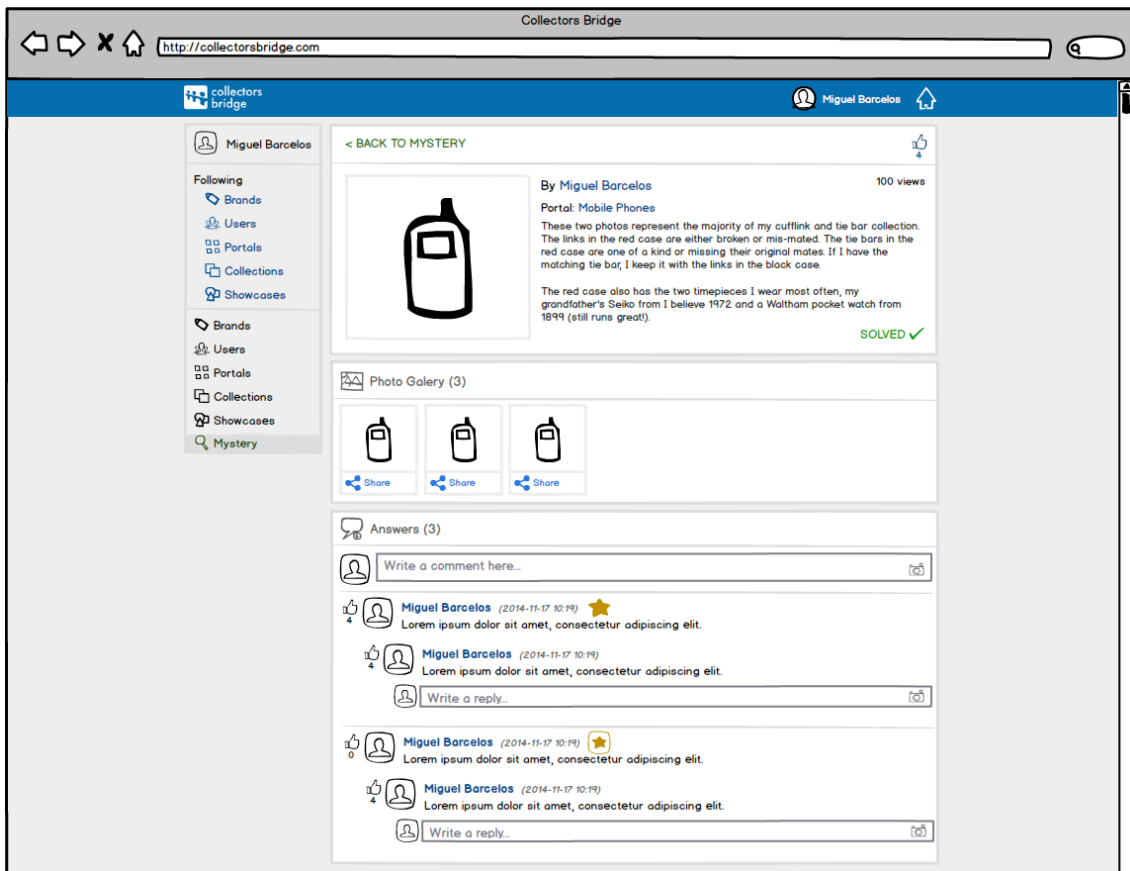


Figura H-21 - Página relativa a um colecionável do *Mystery* já resolvido (vista por um utilizador)

Um *champion* ou autor do colecionável poderá atribuir um portal ao item, responder, alterar o estado para resolvido e selecionar a resposta que acha mais correta (Figura H-22).

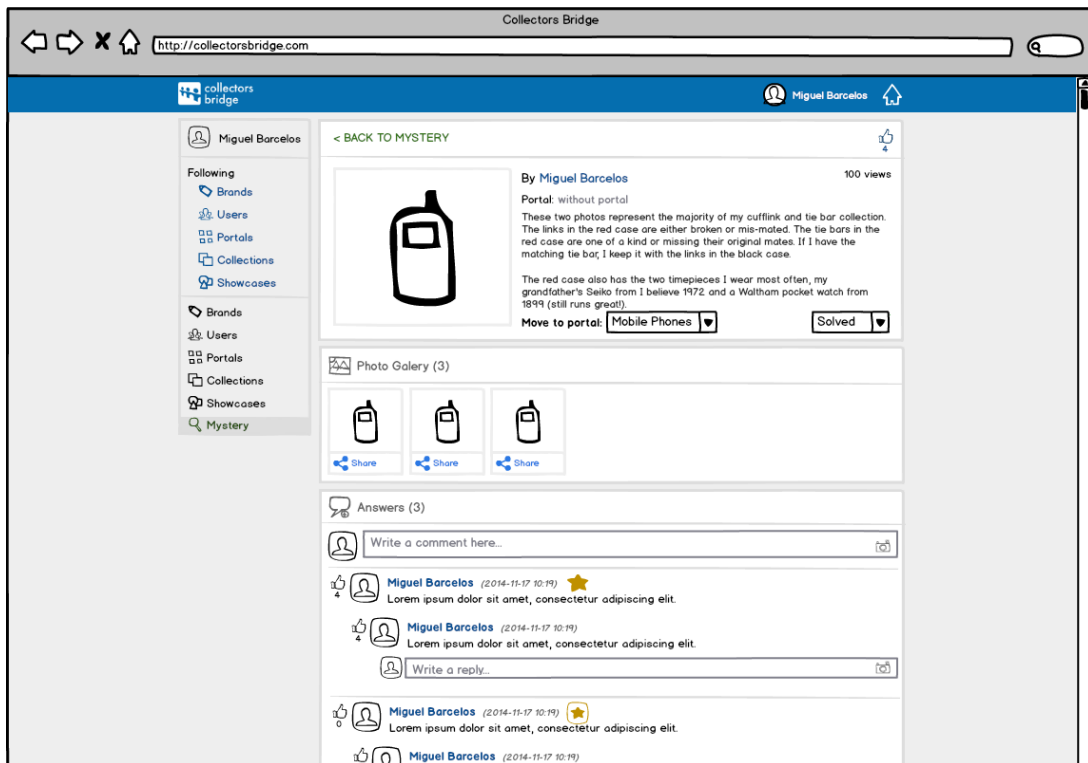


Figura H-22 - Página relativa a um colecionável do *Mystery* (vista por um *champion* ou autor)

Irá também ser acrescentado uma vista a cada portal designada por “Mystery”. Nela irá conter todos os itens resolvidos ou por resolver, pertencentes ao portal selecionado (Figura H-23).

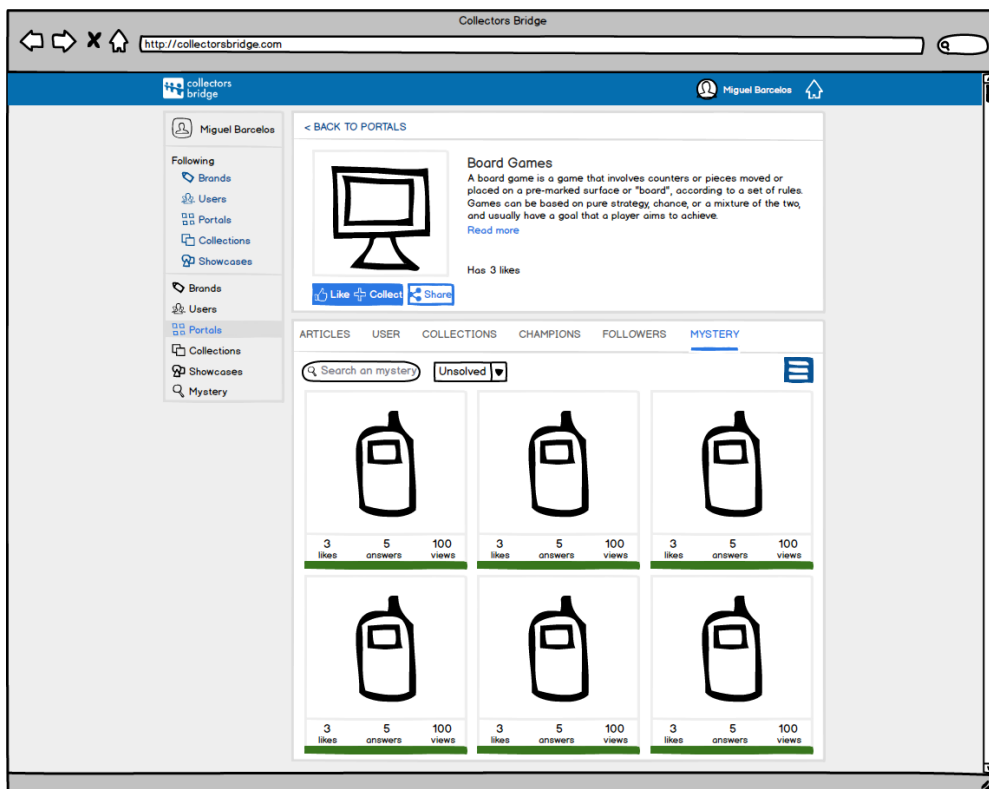


Figura H-23 - Separador *Mystery* dentro de um portal

4. Módulo *Freemium* v1

Contexto: *Freemium* é o nome que se dá a um produto (tipicamente digital) onde se oferece alguns recursos, sendo cobrado recursos adicionais (*premium*). Este modelo de negócio será uma das fontes de receita da Collectors Bridge no futuro.

Objetivos: Pretende-se desenvolver um módulo que implemente o conceito de *freemium*, no qual é necessário arranjar algumas funcionalidades para criar um tipo de utilizador *premium*.

Solução proposta:

Será acrescentado uma opção no menu de utilizador designada de “Upgrade” (Figura H-24).

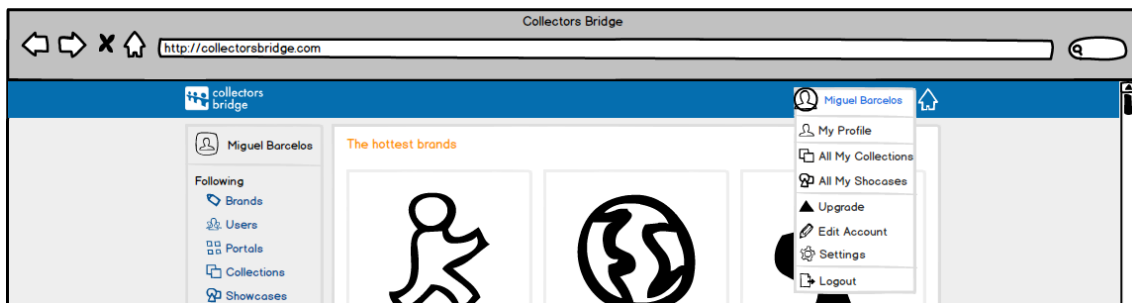


Figura H-24 - Link para o upgrade no menu de utilizador

Na página de *upgrade* aparecerá duas opções no mínimo. Uma free, contendo as principais funcionalidade grátis da plataforma, e pelo menos uma com funcionalidades extra (Figura H-25). As funcionalidades sugeridas para um utilizador *premium* são:

- Preferência nos resultados das pesquisas;
- Crachá *premium* na imagem de perfil do utilizador;
- Remoção da publicidade para o utilizador em toda a plataforma;
- Remoção da publicidade na página do utilizador para qualquer utilizador da plataforma;
- Saber quem viu o seu perfil;
- Imprimir as suas coleções.

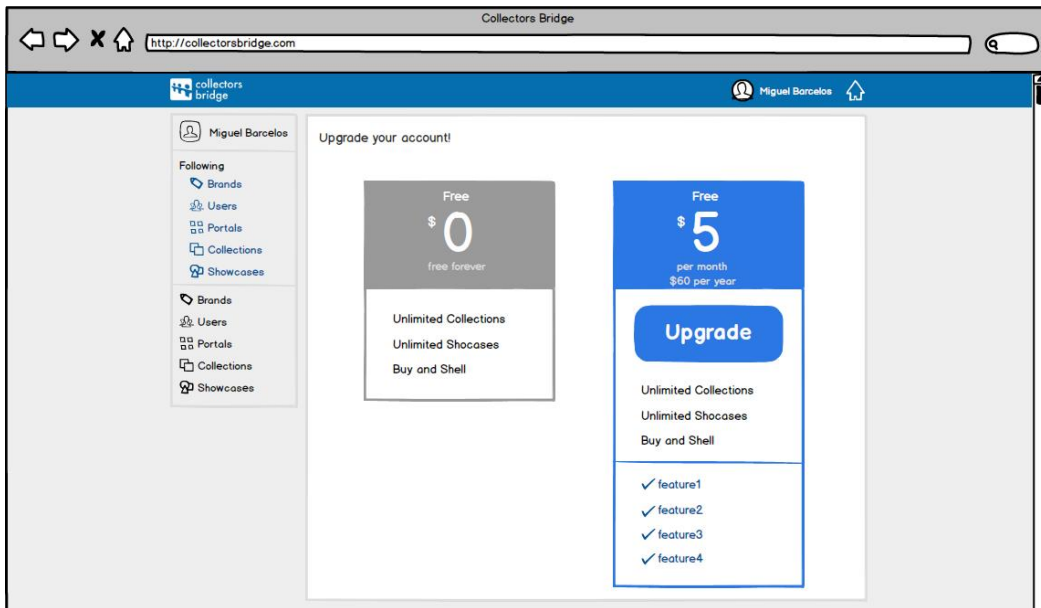


Figura H-25 - Exemplo da página de *upgrade*

Estando o utilizador como *premium* será possível visualizar um crachá na sua foto de perfil e uma opção no menu de utilizador para gerir as funcionalidades *premium* (Figura H-26).

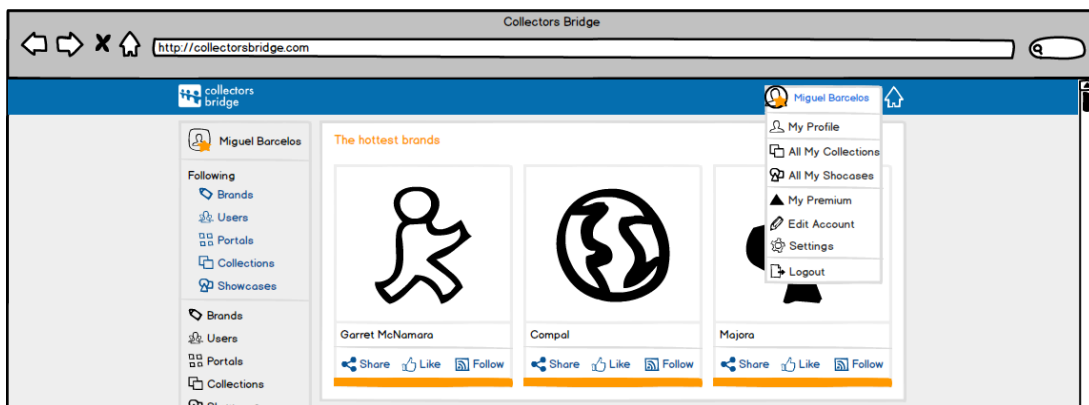


Figura H-26 - Link para gerir as funcionalidades *premium* e foto de perfil com crachá

Na página de gestão das funcionalidades *premium* (*My Premium*) será possível de gerir e usufruir das funcionalidades descritas anteriormente (Figura H-27).

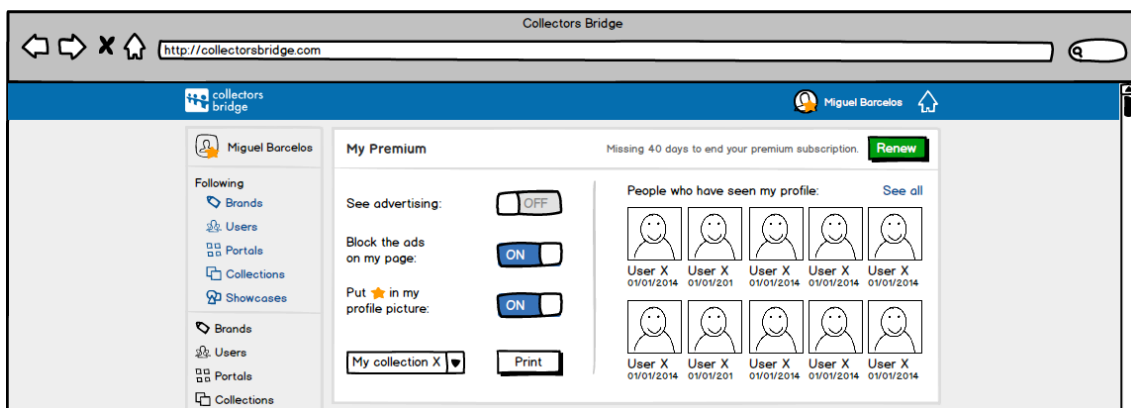


Figura H-27 - Página My Premium

5. Módulo *Message* v1

Contexto e problema: Além do colecionismo, a Collectors Bridge pretende ser uma plataforma que fomenta a comunicação entre os colecionares. Atualmente a comunicação dentro da plataforma só é possível através do sistema de comentários, sendo este público.

Objetivos: Pretende-se construir um sistema de envio de mensagens privadas entre utilizadores.

Solução proposta:

Irá ser adicionado um ícone na barra de topo, dedicado ao sistema de troca de mensagens. No ícone também será indicado o número de mensagens que o utilizador possui não lidas. Ao carregar no ícone, surgirá um balão contendo as ultimas mensagens recebidas, com a opção de ver todas as mensagens (Figura H-28).

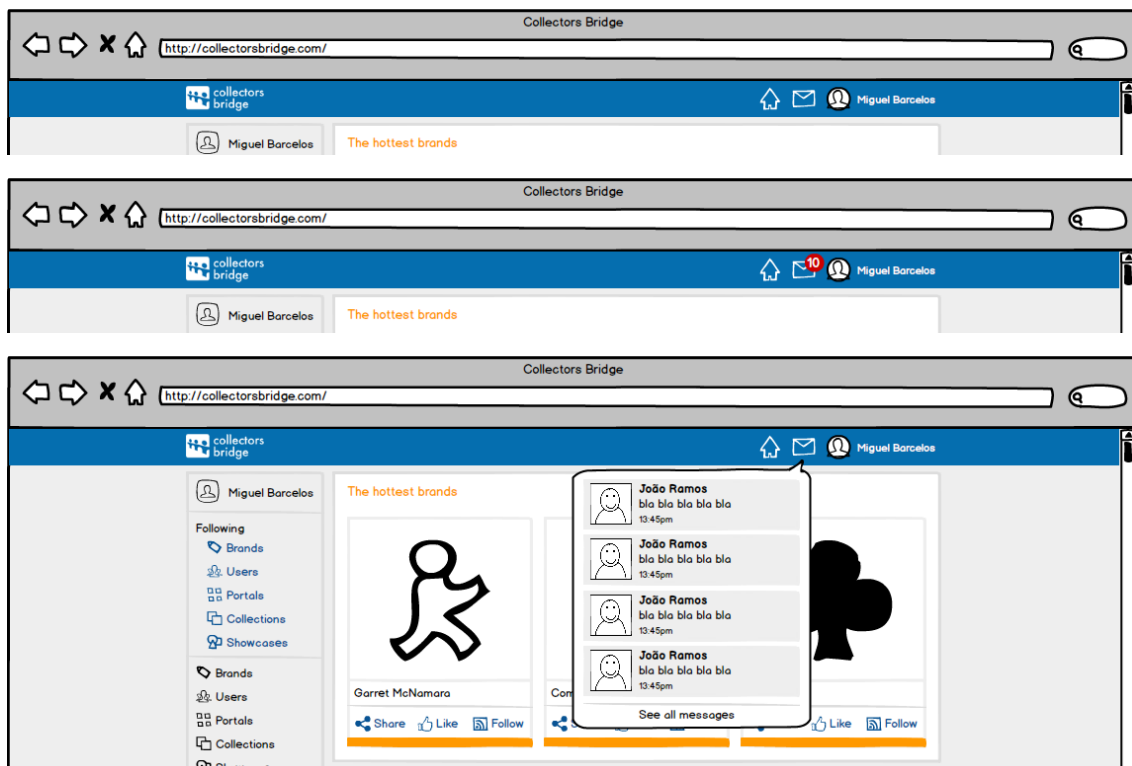


Figura H-28 - Botão de mensagens com o seu comportamento

Ao clicar numa das mensagens do balão, o utilizador será direcionado para a página das mensagens. Nela estão contidas todas as mensagens, sendo possível a pesquisa, leitura e escrita de mensagens (Figura H-29).

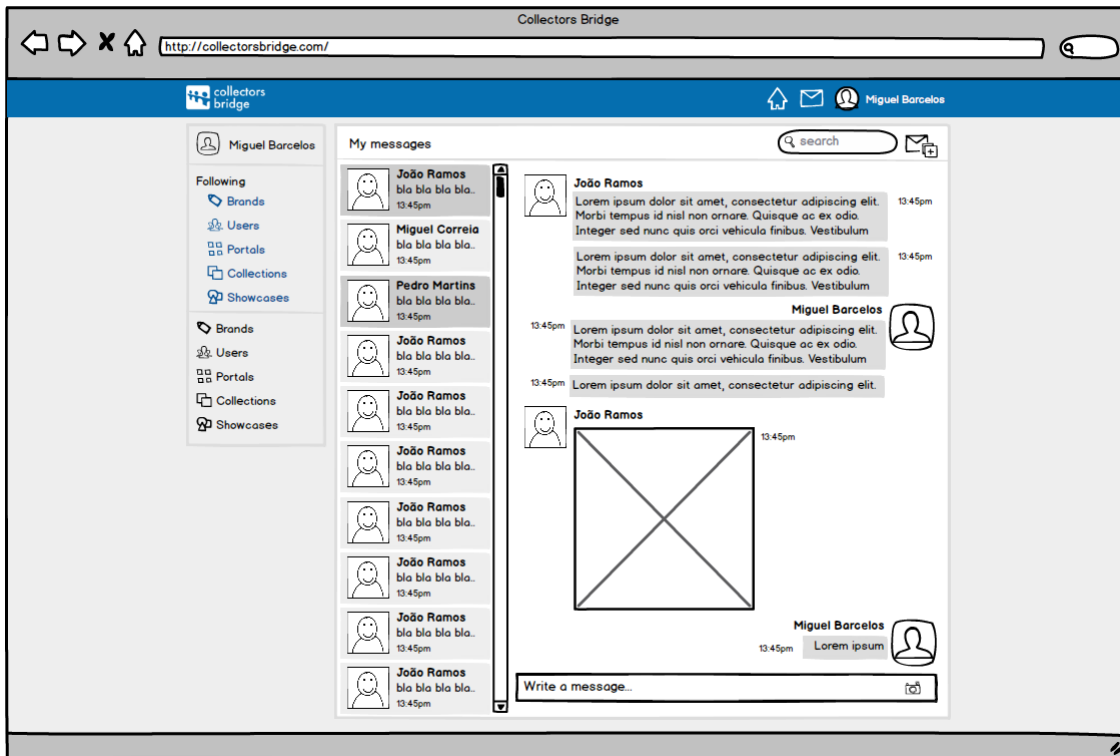


Figura H-29 - Mensagens entre dois utilizadores

Para criar uma nova mensagem o utilizador terá de clicar no ícone referente a uma nova mensagem, presente na página de perfil de cada utilizador (Figura H-30 e Figura H-31) ou na própria página de mensagens (Figura H-29).



Figura H-30 - Página de utilizador com botão de envio de mensagem

Ao estar na página de criação de nova mensagem, o utilizador selecionará o recetor e enviará a mensagem (Figura H-31).

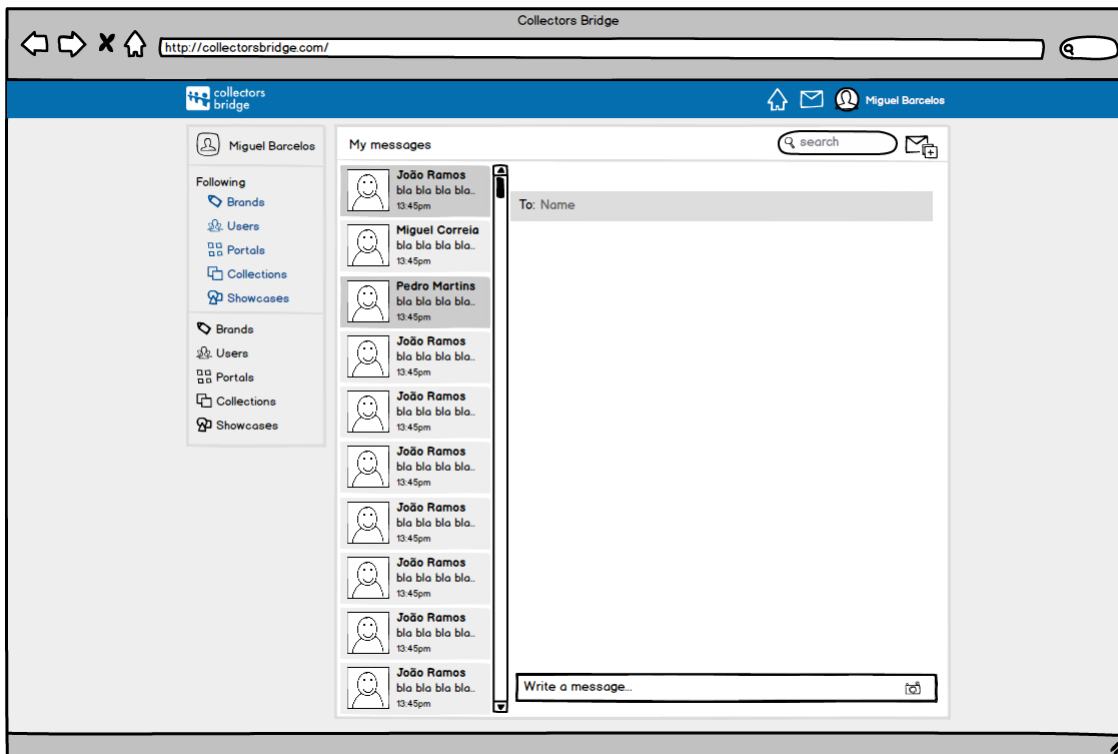


Figura H-31 - Nova mensagem

6. Módulo de Comentários v2

Contexto e objetivos: Atualmente a plataforma permite comentários a artigos, *portals*, *collections*, *shocases*, *brands* e *users*. Pretende-se evoluir este módulo para que possa ser possível a adição de imagens nos comentários e *preview* da informação nos *URLs* inseridos.

Solução proposta:

Irá ser permitido a adição de imagens nos comentários através de um ícone inserido na caixa de texto dos comentários. O *preview* dos *URLs* surgirá automaticamente ao inserir um *URL*, no qual será possível ver uma imagem da página, título e descrição (Figura H-32).

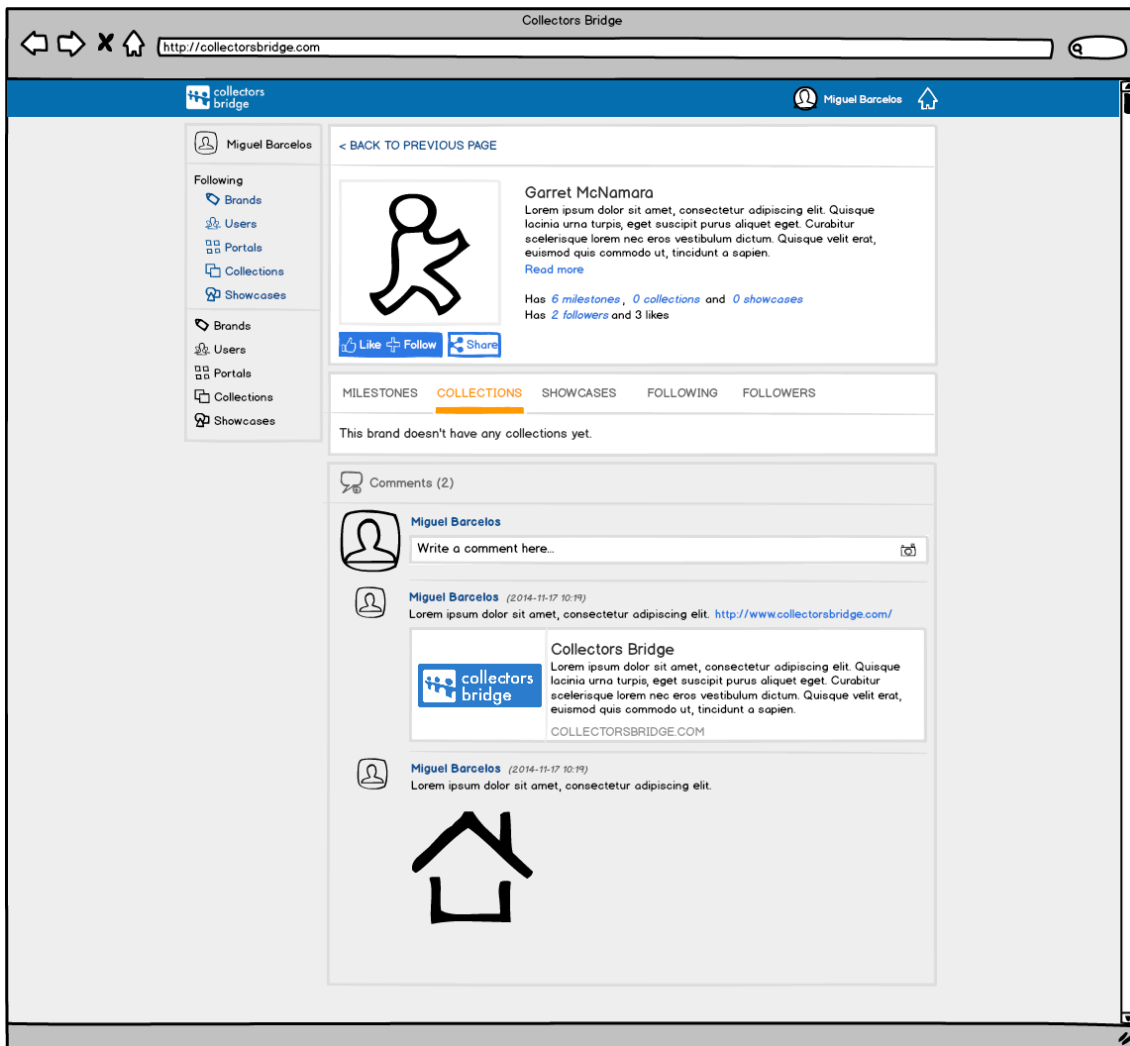


Figura H-32 - Exemplo de comentários

7. Módulo External Content v1

Contexto e problema: Um dos grandes problemas iniciais da Collectors Bridge é a inserção de conteúdos na plataforma, pois ainda tem poucos *champions*. Como a inserção de conteúdos envolve conhecimento do assunto, a Collectors Bridge pretende contornar o problema, permitindo retirar dados de outras fontes, sem ter de violar os direitos de autor das mesmas, de forma a inserir de forma rápida uma grande quantidade de dados nesta fase inicial.

Objetivos: Pretende-se construir um incorporador de informação de outros sítios, dentro de um item de um portal.

Solução proposta:

Será adicionado um ou vários separadores na página de um artigo, no qual a informação da página externa será incorporada dentro de uma área de dimensão fixa. No Figura H-33 é exemplificado essa incorporação com um artigo da *wikipedia*.

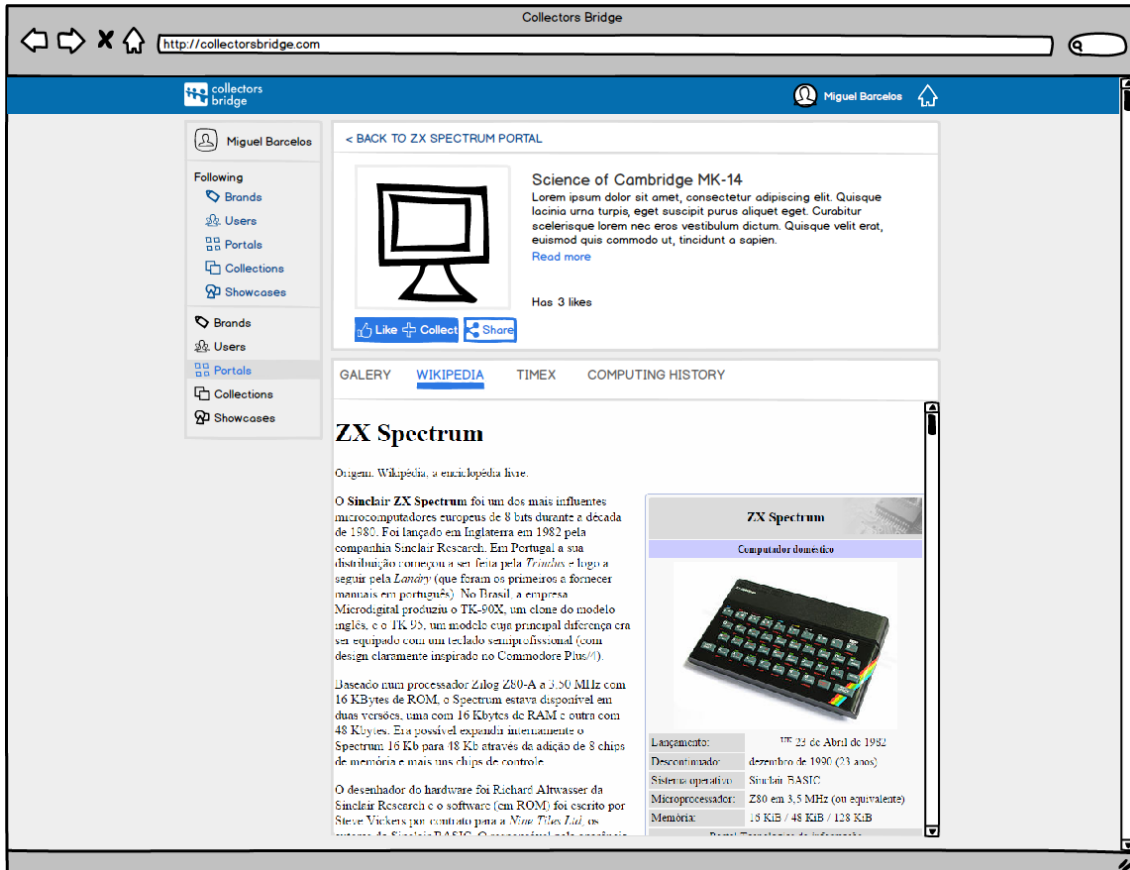


Figura H-33 - Exemplo de uma possível incorporação da Wikipedia num separador de um artigo

8. Módulo de Vídeo v1

Contexto e problema: Atualmente a única forma de descrever um artigo na plataforma é com texto e fotografias. Outra forma de ilustrar um artigo é com recurso ao vídeo.

Objetivos: Pretende-se possibilitar a adição de vídeos à galeria multimédia dos artigos, sem que o formato quadrangular de cada item seja alterado.

Solução proposta:

Os vídeos serão dispostos na mesma forma das fotográficas, estando identificados por uma *frame* aleatória do vídeo, com um botão de *play* sobreposto (Figura H-34).

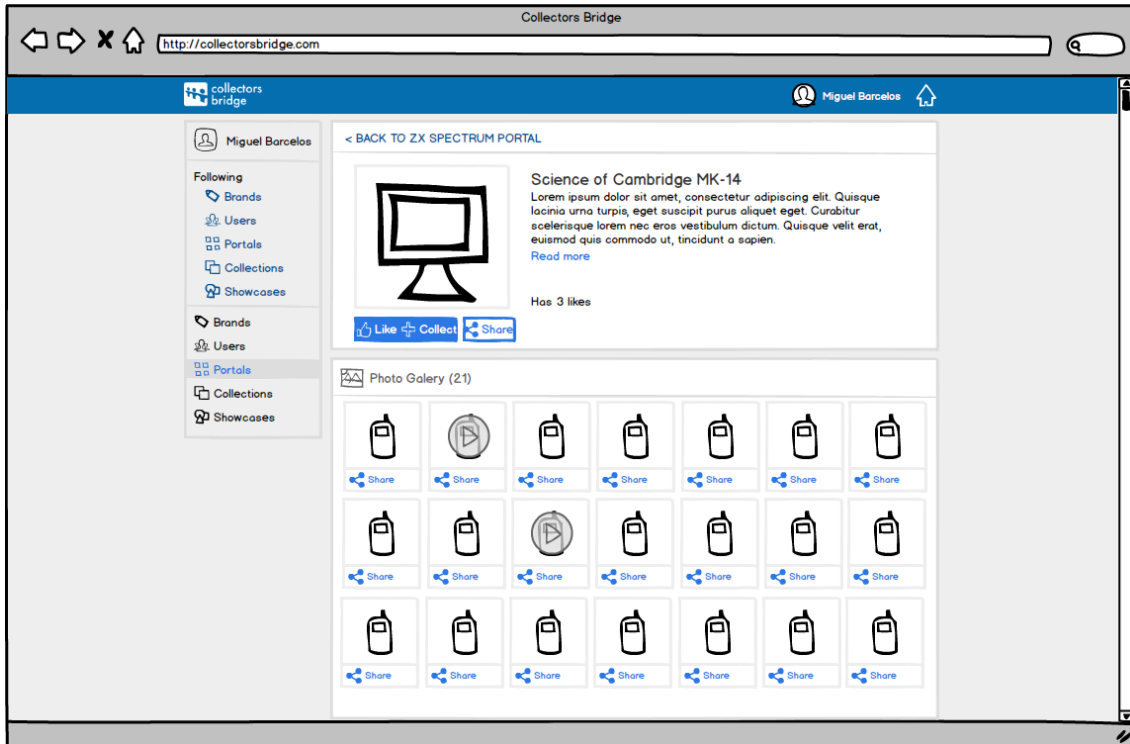


Figura H-34 - Galeria de um artigo com 2 vídeos

Ao carregar num vídeo surgirá uma *popup*, no qual o vídeo começará automaticamente a ser reproduzido (Figura H- 35).

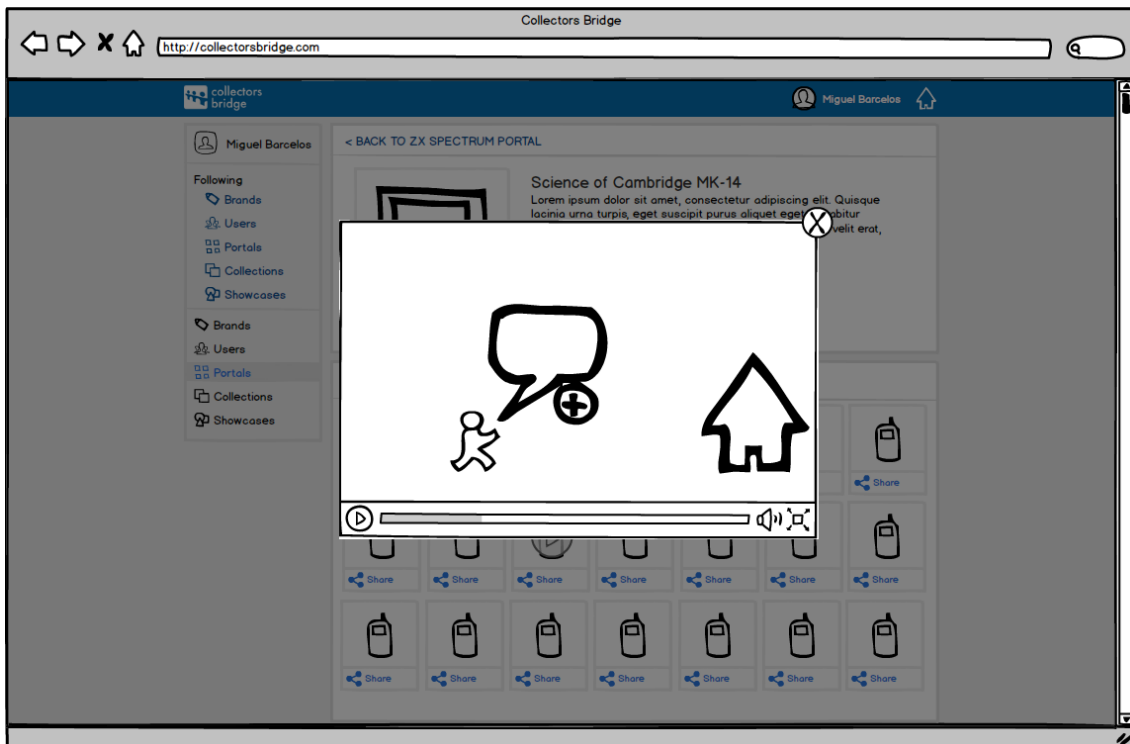


Figura H- 35 - Vídeo em execução

Anexo I - Testes funcionais manuais

Neste anexo são listados todos os testes funcionais, que não foram automatizados. À semelhança dos testes de regressão automatizados, estes seguem a forma de cenários com resultados esperados, como por exemplo:

Dado <contexto>
 Quando <evento/ação>
 <resultados esperados>

De seguida são listados todos os testes realizados de forma manual no *marketplace*.

- | | |
|----|---|
| 1 | Dado um vendedor com um artigo no modo de edição
Quando fizer <i>crop</i> na imagem de capa e submeter
A imagem de capa deverá ser apresentada com o <i>crop</i> realizado |
| 2 | Dado um vendedor com um artigo no modo de edição
Quando fizer <i>crop</i> numa imagem da galeria e submeter
A imagem da galeria deverá ser apresentada com o <i>crop</i> realizado |
| 3 | Dado um comprador com um artigo por pagar
Quando ele fizer o checkout e pagar no PayPal
O comprador deverá ser notificado por <i>email</i> ;
O vendedor deverá ser notificado por <i>email</i> ; |
| 4 | Dado um vendedor com um artigo no estado de pagamento enviado
Quando ele fizer <i>refund</i> para o comprador no PayPal
O comprador deverá ser notificado por <i>email</i> ;
O vendedor deverá ser notificado por <i>email</i> ; |
| 5 | Dado um utilizador registado
Quando fizer licitar um artigo
O vendedor deverá receber uma notificação por <i>email</i> |
| 6 | Dado um comprador com uma licitação aceite
Quando rejeitar a proposta do vendedor
O vendedor deverá receber uma notificação por <i>email</i> |
| 7 | Dado um comprador com um artigo pago
Quando colocar na plataforma que recebeu o artigo
O vendedor deverá receber uma notificação por <i>email</i> |
| 8 | Dado um comprador com uma compra falhada
Quando fizer a avaliação da compra
O vendedor deverá receber uma notificação por <i>email</i> |
| 9 | Dado um vendedor com uma licitação pendente
Quando rejeitar a licitação
O comprador deverá receber uma notificação por <i>email</i> |
| 10 | Dado um vendedor com uma licitação pendente
Quando aceitar a licitação
O comprador deverá receber uma notificação por <i>email</i> |
| 11 | Dado um vendedor com um artigo num estado público com licitações pendentes e sem pagamentos a decorrer
Quando cancelar o artigo
Todos os licitadores deverão receber uma notificação por <i>email</i> |
| 12 | Dado um vendedor com um artigo no estado de recebido |

	Quando avaliar a venda O comprador deverá receber uma notificação por <i>email</i>
13	Dado um vendedor com uma venda falhada Quando fizer a avaliação da venda O comprador deverá receber uma notificação por <i>email</i>
14	Dado um vendedor com um artigo público e uma pergunta sobre ele Quando responder à pergunta O utilizador autor da pergunta deverá receber uma notificação por <i>email</i>
15	Dada uma licitação pendente Quando chegar ao dia da expiração O comprador deverá ser notificado por <i>email</i> O vendedor deverá ser notificado por <i>email</i>
16	Dada uma licitação aceite e à espera por uma resposta do comprador Quando chegar ao dia da expiração O comprador deverá ser notificado por <i>email</i> O vendedor deverá ser notificado por <i>email</i>
17	Dado um artigo no estado de pago ou enviado Quando chegar ao dia da expiração O comprador deverá ser notificado por <i>email</i> O vendedor deverá ser notificado por <i>email</i>
18	Dado um artigo no estado de recebido ou falhado Quando chegar ao dia da expiração O comprador deverá ser notificado por <i>email</i> O vendedor deverá ser notificado por <i>email</i>
19	Dado um vendedor com um artigo no estado de não publicado Quando abrir a opção de inserção de <i>multi-upload</i> , da galeria de imagens, seleccionar 2 imagens e carregar em adicionar As 2 imagens deverão ser inseridas no sistema
20	Dado um vendedor com um artigo no estado de não publicado e com uma imagem na galeria Quando editar a descrição da imagem da galeria A descrição da imagem deverá ser alterada
21	Dado um vendedor com um artigo no estado de não publicado e com uma imagem na galeria Quando editar a descrição da imagem da galeria A descrição da imagem deve ser alterada
22	Dado um vendedor com um artigo no estado de não publicado Quando fizer <i>upload</i> de uma nova foto de capa A foto de capa deverá ser alterada
23	Dado um utilizador e um artigo num estado público Quando carregar na opção de partilhar no Facebook O artigo deverá ser partilhado no Facebook
24	Dado um utilizador e um artigo num estado público Quando carregar na opção de partilhar no Twitter O artigo deverá ser partilhado no Twitter
25	Dado um utilizador e um artigo num estado público Quando carregar na opção de partilhar no Google+ O artigo deverá ser partilhado no Google+
26	Dado um utilizador e um artigo num estado público Quando carregar na opção de partilhar no Pinrest O artigo deverá ser partilhado no Pinrest