



Dissertação/Estágio  
Mestrado em Engenharia Informática  
Sistemas de Informação

## Aplicação Web para Gestão de Associados

João Carlos Sousa Ribeiro

Orientador  
Prof. Álvaro Rocha

Departamento de Engenharia Informática  
Faculdade de Ciências e Tecnologia  
Universidade de Coimbra

2014/2015





Dissertação/Estágio  
Mestrado em Engenharia Informática  
Sistemas de Informação

## Aplicação Web para Gestão de Associados

João Carlos Sousa Ribeiro

Orientador  
Prof. Álvaro Rocha

*Júri Arguente*  
Prof. Carlos Nuno Laranjeiro

*Júri Vogal*  
Prof. Edmundo Monteiro

Departamento de Engenharia Informática  
Faculdade de Ciências e Tecnologia  
Universidade de Coimbra

2014/2015



Este documento não foi redigido ao abrigo do novo acordo ortográfico.



# Resumo

Actualmente a sociedade dá mais importância às aplicações desenvolvidas para a web do que a aplicações desktop, havendo dentro desta categoria um variado leque de opções disponíveis, desde redes sociais abertas a toda a comunidade até serviços de *streaming* de música ou filmes.

O objectivo deste trabalho passa pelo desenvolvimento de uma aplicação web de gestão de associados para a Associação Ibérica de Sistemas e Tecnologias de Informação, de forma a que esta passe a ser utilizada de forma recorrente pelos seus associados e gestores. O projecto passou por várias fases começando pelo estudo do estado da arte, onde foram analisadas várias aplicações web de gestão de associados, consideradas concorrentes da aplicação que se desenvolveu. Passada esta fase foi iniciada a fase de levantamento de requisitos, onde foram efectuadas reuniões com vista a definir todos os requisitos do sistema. Com os requisitos já levantados e descritos chegou a altura de iniciar a definição da arquitectura, desenhando o seu modelo e ainda o diagrama de entidades e relacionamentos que descreve a organização da base de dados. De seguida iniciou-se o desenvolvimento da aplicação web, dividido por módulos que foram posteriormente integrados de maneira a formar o sistema completo. Com a aplicação web desenvolvida passou-se ao *deployment* da mesma, num servidor temporário e aos testes unitários e funcionais sobre o que se encontrava já funcional, na fase de testes estudou-se ainda o melhor método para efectuar os testes de usabilidade. Com esta aplicação web é possível a novos associados efectuarem o seu registo e o pagamento das anuidades, para que posteriormente o administrador do sistema possa consultar todos os seus associados. É ainda possível que os associados se registem em eventos criados na aplicação web.

Ao longo do projecto surgiram alguns desafios que foram superados pelo autor, sendo esta uma solução que se encontra já online e pronta a funcionar.

**Keywords:** Gestão de associados, Aplicação web, Associações, Associação Ibérica de Sistemas e Tecnologias de Informação, Engenharia de Software, Desenvolvimento Web





# Abstract

Nowadays, society pays more attention to web applications than to desktop application, and there is a huge amount of variety: from open social networks to streaming services of video and/or music.

This project's main goal was to develop a web app to manage the members of the IAIST association, making an app useful for both the associates and the association managers.

The project went through different phases, starting by the state of the art study, where various competitors web applications, for membership management, were analysed. After this phase the requirements definition started. These were gathered thanks to some meetings with the goal of describing all the important requirements for the system.

With all the requirements gathered and described, the time came to start defining the system's architecture, defining its model and the entity-relationship diagram with the goal of depicting the way the database is organised. Next, development started, divided in modules and then integrated with each other. After this phase the deployment of the web application to the remote server was initiated, at the same time the application was being tested through unit and functional tests. During the testing phase the usability evaluation tests methodology was also studied and prepared.

This web application makes it possible for new associates to register and pay their annual fees, so that the administrator can then manage all the members of the IAIST. It's also possible for users to register in events created through the web application.

Throughout this project, some challenges have occurred that were overcome by the author resulting in a final web app that is online and ready to be used by the association.

**Keywords:** Membership management, Web Application, Associations, Iberian Association for Information Systems and Technologies, Software Engineering, Web Development



# Agradecimentos

Gostaria de começar por agradecer à minha família, em especial aos meus pais, Carlos e Graça. Sem eles não seria possível ter desenvolvido todo o trabalho realizado até agora. Por todo o apoio dado ao longo do meu percurso académico, o meu enorme Obrigado.

Queria também agradecer ao Professor Álvaro Rocha que me deu a oportunidade de ser orientado por ele no decorrer de toda a dissertação de mestrado.

Queria agradecer ainda à Ana, à Bia, à Sara e ao António pelas correcções, pelos reparos e por todas as dicas de escrita para esta dissertação de mestrado.

A estas quatro pessoas gostaria de juntar o Daniel, o Marques, o Shark, o Pedro Correia, a Mariana, a Catarina Parente, a Alexandra, a Joana e o José Ribeiro. Por toda a amizade e apoio que demonstraram durante não só este último ano, mas durante todo o meu percurso académico.

A estes e a todos os que, apesar de não serem aqui mencionados, me marcaram e apoiaram ao longo destes seis anos e meio, obrigado!



# Índice

<b>Capítulo 1: Introdução</b> . . . . .	<b>1</b>
1.1 A AISTI . . . . .	1
1.2 Contexto . . . . .	2
1.3 Motivação . . . . .	2
1.4 Objectivos . . . . .	2
1.5 Estrutura . . . . .	2
<b>Capítulo 2: Estado da Arte</b> . . . . .	<b>5</b>
2.1 Associações . . . . .	5
2.2 Aplicações Web . . . . .	7
2.3 Análise de Concorrência . . . . .	10
2.3.1 WildApricot . . . . .	11
2.3.2 AssociaPro . . . . .	12
2.3.3 ClubMaster . . . . .	14
2.3.4 Silkstart . . . . .	16
2.3.5 BigTent . . . . .	17
2.3.6 YourMembership.com . . . . .	19
2.3.7 SoftManagement . . . . .	21
2.3.8 Discussão e Conclusões . . . . .	22
<b>Capítulo 3: Descrição do Sistema</b> . . . . .	<b>25</b>
3.1 Análise de Requisitos . . . . .	25
3.1.1 Requisitos Funcionais . . . . .	25
3.1.2 Requisitos Não Funcionais . . . . .	29
3.1.3 Restrições . . . . .	31
3.2 Casos de Uso . . . . .	32
3.2.1 Descrição dos Casos de Uso . . . . .	32
3.2.2 Diagrama UML . . . . .	54
3.3 Arquitectura do Sistema . . . . .	57
3.3.1 Diagrama de entidade e relacionamentos . . . . .	57
3.3.2 Modelo de arquitectura . . . . .	58
3.3.3 Diagrama de componentes . . . . .	60
3.4 Escolhas Tecnológicas . . . . .	61
3.4.1 Desenvolvimento . . . . .	61
3.4.2 Apoio ao desenvolvimento Web . . . . .	62
3.4.3 Base de dados . . . . .	63
<b>Capítulo 4: Planeamento</b> . . . . .	<b>65</b>
4.1 Tarefas Realizadas . . . . .	65
4.1.1 Primeiro Semestre . . . . .	66
4.1.2 Segundo Semestre . . . . .	69

4.2	Ferramentas utilizadas . . . . .	75
4.2.1	Git . . . . .	75
4.2.2	Trello . . . . .	75
4.2.3	Google Docs . . . . .	75
4.2.4	Visual Paradigm . . . . .	76
4.2.5	Ngrok . . . . .	77
4.2.6	Digital Ocean . . . . .	77
<b>Capítulo 5: Desenvolvimento . . . . .</b>		<b>79</b>
5.1	Módulo de gestão de utilizadores . . . . .	79
5.2	Módulo de gestão e monitorização de facturas e anuidades . . . . .	84
5.3	Módulo de Gestão de Paypal . . . . .	86
5.4	<i>Deployment</i> . . . . .	87
<b>Capítulo 6: Testes . . . . .</b>		<b>89</b>
6.1	Testes à base de dados . . . . .	89
6.1.1	Estratégia . . . . .	89
6.2	Testes Funcionais . . . . .	90
6.2.1	Estratégia . . . . .	90
6.3	Testes de segurança . . . . .	91
6.3.1	Estratégia . . . . .	91
6.4	Testes de performance . . . . .	92
6.5	Testes de usabilidade . . . . .	93
<b>Capítulo 7: Conclusão . . . . .</b>		<b>95</b>
7.1	Trabalho futuro . . . . .	95
7.1.1	Testes de usabilidade . . . . .	95
7.1.2	Deployment . . . . .	95
7.1.3	Manutenção . . . . .	95
7.2	Considerações finais . . . . .	96
<b>Apêndice A: Questionário de usabilidade para utilizadores . . . . .</b>		<b>97</b>
A.1	Informação Pessoal . . . . .	97
A.2	Tarefas . . . . .	97
A.3	Tabela de questionário . . . . .	97
<b>Apêndice B: Questionário de usabilidade para administradores . . . . .</b>		<b>99</b>
B.1	Informação Pessoal . . . . .	99
B.2	Tarefas . . . . .	99
B.3	Tabela de questionário . . . . .	99
<b>Apêndice C: Questionário de Feedback . . . . .</b>		<b>101</b>
C.1	Informação Pessoal . . . . .	101
C.2	Tarefas . . . . .	101
C.3	Tabela de questionário . . . . .	101
<b>Apêndice D: Relatório de GTMetrix . . . . .</b>		<b>103</b>
<b>Bibliografia . . . . .</b>		<b>109</b>

# Lista de Figuras

2.1	Dispositivos móveis . . . . .	8
2.2	Design responsivo . . . . .	9
2.3	Ecrã da aplicação web WildApricot . . . . .	12
2.4	Ecrã da aplicação web AssociaPro . . . . .	14
2.5	Ecrã da aplicação web ClubMaster . . . . .	16
2.6	Ecrã da aplicação web SilkStart . . . . .	17
2.7	Ecrã da aplicação web BigTent . . . . .	19
2.8	Ecrã da aplicação web YourMembership.com . . . . .	21
2.9	Ecrã da aplicação web SoftManagement . . . . .	22
3.1	Níveis de casos de uso - Imagem retirada de [1] . . . . .	33
3.2	Diagrama UML de casos de uso geral . . . . .	54
3.3	Diagrama UML de casos de uso de Visitantes . . . . .	55
3.4	Diagrama UML de casos de uso de Associados . . . . .	55
3.5	Diagrama UML de casos de uso de Administradores . . . . .	56
3.6	Diagrama ER da base de dados . . . . .	57
3.7	Modelo de Arquitectura MVC . . . . .	58
3.8	Modelo de arquitectura do projecto . . . . .	59
3.9	Diagrama de componentes . . . . .	60
4.1	Diagrama de GANTT do 1º Semestre . . . . .	67
4.2	Diagrama de GANTT do 2º Semestre . . . . .	70
4.3	Repositório Git . . . . .	75
4.4	Quadro de tarefas do Trello . . . . .	76
4.5	Pasta geral da dissertação de mestrado no Google Docs . . . . .	76
4.6	Área de trabalho do Visual Paradigm . . . . .	76
4.7	Processo de tunneling do Ngrok . . . . .	77
4.8	Droplet do Digital Ocean . . . . .	77
5.1	Erro de validação de e-mail . . . . .	80
5.2	E-mail de confirmação de registo . . . . .	80
5.3	Fotografia de perfil do utilizador . . . . .	81
5.4	Cartão de associado . . . . .	81
5.5	Indicador de novas mensagens com novas mensagens . . . . .	81
5.6	Indicador de novas mensagens a zero . . . . .	82
5.7	Lista de membros registados na aplicação web . . . . .	82
5.8	Menu de alteração de estado . . . . .	83
5.9	Formulário de tipo de registo em evento . . . . .	84
5.10	Campo para upload de comprovativo . . . . .	85
5.11	Confirmação de pagamento da factura . . . . .	86
5.12	Página de pagamento do Paypal . . . . .	86
5.13	Estatísticas do servidor Digital Ocean . . . . .	87

5.14	Erro da aplicação web . . . . .	88
6.1	Dados de transacções da aplicação web . . . . .	92
6.2	Dados de servidor . . . . .	92
6.3	Rating da aplicação web na ferramenta GTMetrix . . . . .	93
6.4	Rating da aplicação web na ferramenta PageSpeed . . . . .	93



# Lista de Tabelas

2.1	Alteração dos membros da associação. Tabela retirada de [2]	6
2.2	Tabela comparativa de concorrência	24
3.1	Tabela de requisitos de associados.	27
3.2	Tabela de requisitos de geração de documentos.	28
3.3	Tabela de requisitos financeiros.	28
3.4	Tabela de requisitos extra.	29
3.5	Descrição de caso de uso RQ01	34
3.6	Tabela referente ao caso de uso RQ02	35
3.7	Descrição de caso de uso RQ03	36
3.8	Descrição de caso de uso RQ04	37
3.9	Descrição de caso de uso RQ05	38
3.10	Descrição de caso de uso RQ06	39
3.11	Descrição de caso de uso RQ07	40
3.12	Descrição de caso de uso RQ09	41
3.13	Descrição de caso de uso RQ10	42
3.14	Descrição de caso de uso RQ11	43
3.15	Descrição de caso de uso RQ12	44
3.16	Descrição de caso de uso RQ13	45
3.17	Descrição de caso de uso RQ14	45
3.18	Descrição de caso de uso RQ15	46
3.19	Descrição de caso de uso RQ16	47
3.20	Descrição de caso de uso RQ17	48
3.21	Descrição de caso de uso RQ18	49
3.22	Descrição de caso de uso RQ19	50
3.23	Descrição de caso de uso RQ20	51
3.24	Descrição de caso de uso RQ21	52
3.25	Descrição de caso de uso RQ22	53
4.1	Tarefas de estudo de soluções existentes no mercado.	66
4.2	Tarefas de levantamento e definição de requisitos.	68
4.3	Tarefas de arquitectura do sistema	68
4.4	Tarefas de desenvolvimento e teste dos módulos de software.	68
4.5	Tarefas de relatório e apresentação intermédia.	69
4.6	Tarefas de desenvolvimento de integração dos módulos de software - Datas planeadas.	71
4.7	Tarefas de deployment da aplicação web - Datas planeadas	72
4.8	Tarefas de testes à aplicação web - Datas planeadas.	72
4.9	Escrita do relatório final da dissertação de mestrado - Datas planeadas.	73
4.10	Tarefas de desenvolvimento de integração dos módulos de software - Datas reais.	73
4.11	Tarefas de deployment da aplicação web - Datas reais.	74

4.12	Tarefas de testes à aplicação web - Datas reais. . . . .	74
4.13	Escrita do relatório final da dissertação de mestrado - Datas reais. . . . .	74
6.1	Testes à Base de Dados . . . . .	90
6.2	Testes funcionais . . . . .	91
6.3	Testes de segurança . . . . .	91

# Capítulo 1

## Introdução

Hoje em dia há uma necessidade crescente das associações em angariar novos membros. Muitas destas associações mantêm-se ainda antiquadas nos métodos utilizados para levar a cabo essa angariação. Unindo assim esta necessidade à crescente utilização de aplicações web, surgiu a necessidade deste projecto.

Neste capítulo é feita uma introdução ao projecto, para que o leitor possa compreender melhor de que forma nasceu a proposta de dissertação. A introdução está dividida em várias secções, onde é apresentada a Associação Ibérica de Sistemas e Tecnologias de Informação, o Contexto, a Motivação, os Objectivos e a Estrutura desta dissertação.

### 1.1 A AISTI

A Associação Ibérica de Sistemas e Tecnologias de Informação (AISTI)<sup>1</sup> foi fundada em 2007, sendo desde então uma associação técnico-científica sem fins lucrativos. A associação tem como objectivo principal a promoção e divulgação do domínio dos Sistemas e Tecnologias de Informação na península Ibérica, para que desta forma a ligação entre a academia, a investigação, as empresas e a sociedade possa ser dinamizada.

Para que a dinamização seja conseguida, a AISTI oferece aos seus membros algumas regalias, como a Revista Ibérica de Sistemas e Tecnologias de Informação (RISTI), que é um periódico semestral de índole científica, focando a investigação no domínio dos sistemas e tecnologias de informação. Neste periódico são publicados artigos originais e inovadores, aceites por um processo de avaliação de no mínimo três membros do conselho científico. Cada número representa um tema específico, anunciado previamente na chamada de artigos. Apenas são aceites entre 5 a 8 artigos para publicação, demonstrando que a taxa média de aceitação é bastante reduzida.

Para além da publicação impressa a AISTI organiza ainda: a CISTI, Conferência Ibérica de Sistemas e Tecnologias de Informação, conferência anual que vai já para a sua 10<sup>a</sup> edição. A CISTI tem como objectivo o apresentar e o discutir de conhecimentos, novas perspectivas e inovações no domínio dos sistemas de tecnologias de informação; a WorldCIST que teve a sua terceira edição em Abril de 2015 nos Açores, e ainda a CIEm, uma conferência direccionada ao empreendedorismo que vai já para a sua 5<sup>a</sup> edição a decorrer em Oeiras, Portugal. Para além destas três conferências a AISTI atribui ainda o prémio para a melhor tese de doutoramento ibérico em STI, dá pareceres para a criação de novos cursos superiores, faz consultoria e participa em projectos de investigação, entre outras actividades.

---

<sup>1</sup>Para mais informações consultar [www.aisti.eu](http://www.aisti.eu)

## 1.2 Contexto

O projecto nasce da necessidade da associação em aumentar a sua massa associativa actual, objectivo que choca com a metodologia de gestão de associados actual que não tira o melhor partido das tecnologias de informação. A gestão é assim feita a partir de uma folha excel com os nomes dos associados, sendo que cada um pode efectuar o seu registo a partir de um formulário na página da associação.

A partir da falta de um mecanismo de qualidade, nasce a ideia da criação de uma aplicação web para a gestão dos associados da AISTI, ideia essa que é então proposta como dissertação de mestrado, pelo Prof. Álvaro Rocha, presidente da AISTI, com o intuito de que os mecanismos actuais e ultrapassados possam ser suplantados por uma melhor solução.

## 1.3 Motivação

Para colmatar as falhas do actual sistema de gestão de associados da AISTI foi desenvolvida uma aplicação web que possa ser acedida a partir de qualquer local sem qualquer dificuldade. Esta aplicação será ainda adaptada para dispositivos móveis graças a um design responsive que se adapta a estes dispositivos, que têm vindo a ganhar cada vez mais notoriedade no quotidiano da população.

Esta solução permitirá que as funcionalidades da aplicação web fiquem disponíveis aos seus associados e também aos seus administradores, para que os associados da AISTI possam ser geridos com a maior facilidade.

## 1.4 Objectivos

O principal objectivo do projecto foi o desenvolvimento de uma aplicação web de gestão de associados fiável e de fácil utilização para a Associação Ibérica de Sistemas e Tecnologias de Informação.

É importante ter em conta que devem ser focada durante o projecto a possibilidade de gestão de várias situações como: adesões, diferentes estados dos associados, pagamento de quotizações e comunicação com os associados.

A aplicação Web desenvolvida será agora colocada num servidor da AISTI configurado para o efeito, para que passe a ser utilizada pela associação para gerir de forma mais simples todos os seus associados, abrindo ainda as portas a uma maior divulgação da associação e um aumento do seu número de membros.

## 1.5 Estrutura

Este documento encontra-se dividido pelos seguintes capítulos: o estado da arte, a descrição do sistema, o planeamento, a implementação, o testing e a conclusão.

O capítulo do estado da arte aborda o estudo de temas relacionados com esta dissertação de mestrado, bem como um estudo da concorrência, onde são analisadas outras aplicações web de gestão de associados já presentes no mercado.

O capítulo referente à descrição do sistema é onde se encontram vários artefactos relevantes para o sistema: a análise de requisitos, os casos de uso, a arquitectura do sistema e as escolhas tecnológicas tomadas para o projecto.

No capítulo relativo ao planeamento é possível encontrar as diferentes tarefas a realizar, a sua divisão, o trabalho realizado e o trabalho futuro.

No capítulo da implementação irão ser abordados os diferentes módulos desenvolvidos e ainda o *deployment* da aplicação no servidor.

No capítulo referente aos testes da aplicação serão abordados os testes à base de dados, testes funcionais, testes de segurança, testes de performance e ainda os testes de usabilidade.

Na conclusão são feitas algumas considerações em relação ao trabalho já realizado e também referente ao trabalho futuro.



# Capítulo 2

## Estado da Arte

Neste capítulo é apresentado o estado da arte, em primeiro lugar é apresentado um estudo acerca de associações e das suas filiações, de seguida apresenta-se um estudo acerca de aplicações web e da sua utilidade. Por fim é apresentada uma análise de concorrência onde são analisadas várias aplicações de gestão de associados já existentes no mercado.

### 2.1 Associações

Nos dias que correm cada vez mais são as associações e organizações que tentam angariar membros e associados. Seja com ou sem fins lucrativos tornam-se importantes para que as comunidades possam evoluir.

Um factor importante das associações e dos seus associados é a continuidade e a lealdade destes para com a associação. Estes dois factores podem ser conseguidos mais facilmente dependendo da forma como a associação é ou não definida.

Em *Dialectic conceptions in social psychology*[3] as associações e as suas filiações são definidas como abertas ou fechadas, tendo em conta que aquelas que são consideradas abertas se caracterizam por serem mais flexíveis quanto ao movimento dos seus associados entre as fronteiras da associação [3], enquanto que as fechadas são aquelas em que as fronteiras funcionam com barreiras não permitindo movimentação de associados entre elas- [3]. No entanto para Ziller [4] o facto de uma organização ser aberta ou fechada está relacionado com esta ser estável ou variável, definindo que uma associação considerada aberta é aquela que é constituída por um grupo de pessoas que interagem, numa constante mudança dos seus associados [4] e uma fechada é aquela cuja massa de associados se mantém constante ao longo do tempo[4].

É possível afirmar que a forma como a associação é definida internamente influencia directamente as filiações que esta irá atrair ou manter. Há, no entanto, outros factores que poderão fazer com que os associados se mantenham na associação, entre eles o que a associação terá para lhes oferecer, ou até o facto de se tratar de uma filiação paga. O trabalho *Membership matters how member change and continuity affect small group structure, process, and performance* mostra-nos que o acto de uma pessoa se tornar associada perante uma associação é central para a definição e identidades dos grupos[2], apresentando ainda a tabela 2.1 que ilustra as várias alterações que um grupo organizacional poderá sofrer, alterando a sua massa associativa.

**Tabela 2.1:** Alteração dos membros da associação. Tabela retirada de [2]

Alterações ao nº de associados	Onde começa a mudança		
	Membro	Grupo	Forasteiro
Diminuição	Membro adoece ou demite-se	Membro é expulso	Membro é promovido ou despedido
Aumento	Membro regista-se na associação	O grupo recruta um novo membro	É atribuído um novo membro
Sem alteração	Membro troca de turno com outro membro	O grupo substitui o membro	Membro é transferido e substituído

Em “*Organisational Members’ Commitment to Professional Associations*” [5] fala-se de associações profissionais como sendo “*fornecedores de serviços, cujos clientes organizacionais são competidores entre eles*” [5]. Estes clientes são os associados, pois em conjunto “*ajudam a fomentar o desenvolvimento e a promover a consciência acerca da indústria em que se inserem e, separadamente têm benefícios vários como o networking<sup>1</sup>, a educação e ainda várias oportunidades de investigação*”. [5]

Como já foi discutido, uma associação é constituída pelos seus membros. Para os manter, e até mesmo atrair novos elementos, é essencial que a associação seja e se mantenha apelativa.

No trabalho *Relationship marketing activities, commitment, and membership behaviors in professional associations* é definido que a qualidade de filiação deve ser medida pela participação por parte dos seus associados [6]. Mantém-se então a questão que já foi colocada anteriormente, o que pode uma associação fazer para fomentar a participação dos seus associados?

Os benefícios oferecidos pela associação acabam por surgir da participação que os associados actualmente têm nela. Estes benefícios incluem a “*possibilidade de aumentar contactos de negócio, a ligação a redes de negócios exclusivas e ainda a ligação emocional a pessoas com os mesmos ideais*” [5].

A possibilidade de retenção de associados aumenta caso sejam necessários pagamentos para que se possa aceder aos recursos da mesma. Nestes casos o estatuto de membro é um estatuto que não se encontrará ao alcance de todos os possíveis associados. Bhattacharya [7] afirma que existem dois tipos de filiações pagas, “*(i) em que é necessário pagar para ter acesso aos serviços da associação, e (ii) em que os serviços estão disponíveis para o público*.” [7]. O primeiro destes dois tipos de filiação irá ajudar a que a participação dos associados aumente, pois como já foi referido estes sentem que a associação é mais exclusiva, motivando-os à sua participação.

<sup>1</sup>Ligações e contactos que uma pessoa estabelece com outras.



Como definido no regulamento interno[8] e nos estatutos[9] da AISTI, esta “*constitui uma associação privada sem fins lucrativos e rege-se pelo disposto no código civil, nos presentes Estatutos, e por um Regulamento interno.*” [9]. Esta segue um sistema como o que referido por Bhathacharya[7], em que apenas tem acesso aos serviços e bens da associação quem paga a sua anuidade, existindo duas categorias de associados que não necessitam de proceder ao pagamento de anuidades, os Fundadores e os Honorários, que detêm os mesmos direitos apesar de não pagarem quotas anuais.

Os associados da AISTI dividem-se em várias categorias, como descrito no regulamento interno da associação[8]:

1. Fundadores
2. Efectivos
3. Auxiliares
4. Convidados
5. Honorários

Os associados deverão ter uma conduta baseada nos estatutos e regulamento interno da associação, onde estão definidos os seus direitos e deveres.

## 2.2 Aplicações Web

A evolução da web ao longo dos últimos anos contribuiu para o aparecimento de inúmeros serviços e aplicações online. Estes serviços e aplicações podem ir desde armazenamento na cloud<sup>2</sup> até aplicações complexas de gestão e manutenção de grandes vinhas<sup>3</sup>. O desenvolvimento de serviços e aplicações tem sofrido também ele um crescimento enorme, sendo que aparecem inúmeras novas ofertas por dia.

Este aparecimento de novas ideias disponíveis em todo o mundo leva a que o mercado tradicional seja cada vez mais afectado, pois as pessoas têm um novo meio para difundir as suas ideias, não necessitando de investimentos tão elevados como aqueles que o mercado tradicional requer. O caso da Codacy<sup>4</sup> é um dos casos em que o criador da empresa, por ter conhecimentos próprios, lançou o seu produto, desenvolvendo-o com a ajuda de alguns colegas e sem custos próprios.

Estes serviços podem, como já referido, influenciar o mercado tradicional, sendo que um dos casos mais evidentes é do site de serviços de estadia AirBnb, que afectou toda a secção hoteleira, existindo mais de 800.000 ofertas de airbnb em todo mundo [10]. Outro caso de sucesso é o da aplicação Uber que oferece um serviço de transporte semelhante aos táxis. O seu sucesso pode ser espelhado pela sua avaliação recente num valor superior a 30 mil milhões de euros [11]. Apesar de todo o sucesso a polémica acompanha também o Uber, tendo já sido processada e proibida um pouco por todo o mundo[12].

Este crescimento leva também ao gradual desaparecimento das aplicações para desktop. Uma aplicação para desktop é, como nos diz Bychkov, “*qualquer software que pode*

<sup>2</sup>Para mais informações consultar [www.dropbox.com](http://www.dropbox.com)

<sup>3</sup>Para mais informações consultar <http://www.inovwine.com/>

<sup>4</sup>Para mais informações consultar <https://www.codacy.com/>

ser instalado num único computador, e usado de forma a efectuar tarefas específicas”[13]. Isto levanta algumas preocupações ao nível de portabilidade e acessibilidade. Hoje em dia “mais de metade dos utilizadores da internet acede à rede através de equipamentos portáteis”[14], sendo um dos factores que leva à adaptação do mercado para a web, pois as aplicações desktop, como já referido estão também limitadas ao hardware disponível [13].



Figura 2.1: Dispositivos móveis

Há ainda outras diferenças que tendem a favor das aplicações web, como referido por Jeff Smith[15]:

**Manutenção:** As aplicações web são instaladas apenas uma vez, num servidor. Ao efectuar updates é apenas necessário efectuar no servidor e não em todos os computadores.

**Facilidade de utilização:** Torna-se conveniente para os utilizadores poderem aceder à aplicação a partir de qualquer local.

Apesar de haver também pontos a favor das aplicações desktop, estas duas vantagens podem tornar-se mais importantes na altura de efectuar a escolha entre desktop e web. As aplicações web como as conhecemos são “*caracterizadas por uma mistura de funcionalidades sem precedentes*”[16], tornando-as cada vez mais inovadoras.

No desenvolvimento de aplicações web é necessário ter em conta um processo básico, um ciclo de vida com várias etapas. Estas etapas são descritas por Fraternali [16]:

**Análise de requisitos:** Têm como função definir a missão da aplicação.

**Conceptualização:** São criados modelos abstractos que representem os principais componentes da aplicação.

**Prototipagem e validação:** Versões mais simples da aplicação devem ser apresentadas a utilizadores, para que se possa obter feedback no caso de serem necessárias mudanças.

**Design:** Os esquemas conceptuais devem ser transformados numa representação de baixo-nível, mais próxima das necessidades de implementação.

**Implementação:** O servidor deve ser preenchido com o novo conteúdo. Nesta fase devem também ser construídas as páginas web.

**Evolução e manutenção:** Após o produto ser entregue poderá haver alterações aos requisitos e problemas que devem ser resolvidos.

Uma destas etapas torna-se importante quando nos referimos novamente ao facto de hoje em dia se utilizarem cada vez mais os dispositivos móveis, e o facto de que os dispositivos são cada vez mais pequenos e maiores, ao mesmo tempo [17], isto é explicado pela expansão dos smartphones e tablets, ao mesmo tempo que “*as consolas de jogos permitem aos utilizadores aceder aos sites em ecrãs de televisão*” [17]. Essa etapa é o design, onde é necessário ter em foco a portabilidade da mesma. Este último apresenta o mesmo website de forma a que este se adapte ao tamanho de todos os ecrãs, conseguido através de um design *responsive*. [18].



Figura 2.2: Design responsivo

## 2.3 Análise de Concorrência

Foi efectuada uma análise de concorrência de algumas das aplicações de gestão de associados existentes no mercado. A lista de aplicações de gestão de associados que foram testadas, por ordem cronológica de teste, é a seguinte:

1. WildApricot<sup>5</sup>
2. AssociaPro<sup>6</sup>
3. ClubMaster<sup>7</sup>
4. Silkstart<sup>8</sup>
5. Bigtent<sup>9</sup>
6. Yourmembership<sup>10</sup>
7. Softmanagement<sup>11</sup>

As aplicações web foram seleccionadas na sua maioria após consulta de alguns websites que listavam algumas soluções para gestão de associados online[19]. A partir das listas foram efectuadas algumas pesquisas mais por parte do autor, que levaram a seleccionar por fim cinco das sete aplicações web:

- WildApricot, que numa análise efectuada pelo site Business 2 Community conseguiu nota máxima nos campos de facilidade de utilização, características e módulos, valor e facilidade de instalação [20];
- Silkstart foi referida devido a algumas das funcionalidades que oferecem, como o grande foco em eventos;
- YourMembership.com é referida por ser de fácil utilização, permitindo aos seus utilizadores efectuarem um design das suas áreas de gestão de associados com facilidade [21];
- ClubMaster, por ser open source e se poder verificar que numa solução deste género os módulos da versão base acabam por ser limitados.
- A aplicação AssociaPro foi seleccionada por sugestão do Prof. Álvaro Rocha, que já tinha conhecimento acerca desta solução e usou mesmo algumas das funcionalidades como exemplos a seguir.
- Por fim foi seleccionada a aplicação desktop portuguesa SoftManagement, para que se possam notar ainda as vantagens perante uma aplicação desktop.

Para cada uma destas aplicações foram testadas as suas funcionalidades, analisados os seus custos e ainda o apoio que cada uma delas oferece aos seus clientes. Para cada uma delas é apresentado um estudo acerca das suas funcionalidades que é de máxima importância para se perceber quais as componentes que terão de ser trabalhadas ao longo desta dissertação de mestrado.

---

<sup>5</sup>Mais informação disponível em: [www.wildapricot.com](http://www.wildapricot.com)

<sup>6</sup>Mais informação disponível em: [www.associapro.com](http://www.associapro.com)

<sup>7</sup>Mais informação disponível em: [www.clubmaster.org](http://www.clubmaster.org)

<sup>8</sup>Mais informação disponível em: [www.silkstart.com](http://www.silkstart.com)

<sup>9</sup>Mais informação disponível em: [www.bigtent.com](http://www.bigtent.com)

<sup>10</sup>Mais informação disponível em: [www.yourmembership.com](http://www.yourmembership.com)

<sup>11</sup>Mais informação disponível em: [www.softmanagement.pt](http://www.softmanagement.pt)

### 2.3.1 WildApricot

Esta é uma aplicação que oferece vários patamares aos clientes. O mais básico é gratuito e pode ascender a \$200 por mês dependendo do número de associados que a aplicação terá que suportar. As várias funcionalidades oferecidas pela aplicação podem ser acedidas, pelos utilizadores, a partir da página principal. As funcionalidades mais importantes serão seguidamente abordadas.

Os visitantes da página poderão associar-se à organização através de um formulário simples, no qual os utilizadores poderão seleccionar a categoria de sócio a que se desejam afiliar. Estas categorias podem ser adicionadas e alteradas pelo administrador do sistema e cada uma delas poderá ter um custo associado. Este custo deverá ser definido pelo Administrador, bem como os métodos de pagamentos disponíveis, que poderão variar desde pagamento pessoal até pagamento através da internet.

O registo ficará completo assim que o associado terminar a inserção de todos os dados necessários para a sua afiliação à associação. Os dados pedidos no formulário podem ser definidos pelo administrador, para que se garanta que todos os dados necessários serão preenchidos.

O administrador poderá ainda adicionar um novo membro, através do backend<sup>12</sup>, sem que este necessite de efectuar o seu próprio registo. Numa outra instância é possível importar as informações dos associados a partir de ficheiros armazenados no computador.

No backend o administrador poderá ainda efectuar diversas consultas acerca dos membros que estão associados à organização, podendo filtrar a consulta com base em diversos dados relacionados aos seus utilizadores.

Uma funcionalidade interessante que esta aplicação web oferece é a possibilidade de criação de eventos para toda a comunidade que se encontra associada à organização. Para que a criação de um evento fique completa é necessário definir alguns dados a ele referentes, como o título, a data de início, a localização, a descrição, o preço e outros dados não obrigatórios.

Cada evento terá ainda um ou mais tipos de registos, definindo estes os vários níveis de registo num evento, e cada um dos níveis poderá ter um custo diferente, dependendo do que cada utilizador pretende, pois podem ser definidos registos com alimentação incluída, com um convidado ou apenas singular. Os diferentes tipos de registo são definidos no backend pelo administrador do sistema e é ainda possível definir se apenas utilizadores confirmados como associados se podem inscrever ou não.

Após cada pagamento é gerado um recibo e disponibilizado ao associado que acabou de proceder à sua liquidação.

O administrador poderá ainda efectuar algumas acções sobre as facturas, como efectuar consultas dos dados já existentes, podendo filtrar essa pesquisa por campos como o espaço temporal ou pelo nome. Pode ainda adicionar novas facturas de forma manual, registando o método de pagamento, a data e os itens facturados, entre outros.

No separador relativo às finanças da associação é possível consultar todas as facturas geradas pelo sistema até à data, estas facturas incluem o pagamento de anuidades e o pagamento de registos em eventos. Uma das outras funcionalidades que podem permitir à associação ter algum apoio financeiro é a que permite aos associados efectuarem donativos. Para que estes donativos se possam processar é necessário configurar pagamentos online, pois estes podem apenas ser efectuados dessa forma.

---

<sup>12</sup>Área do site destinada a fins administrativos

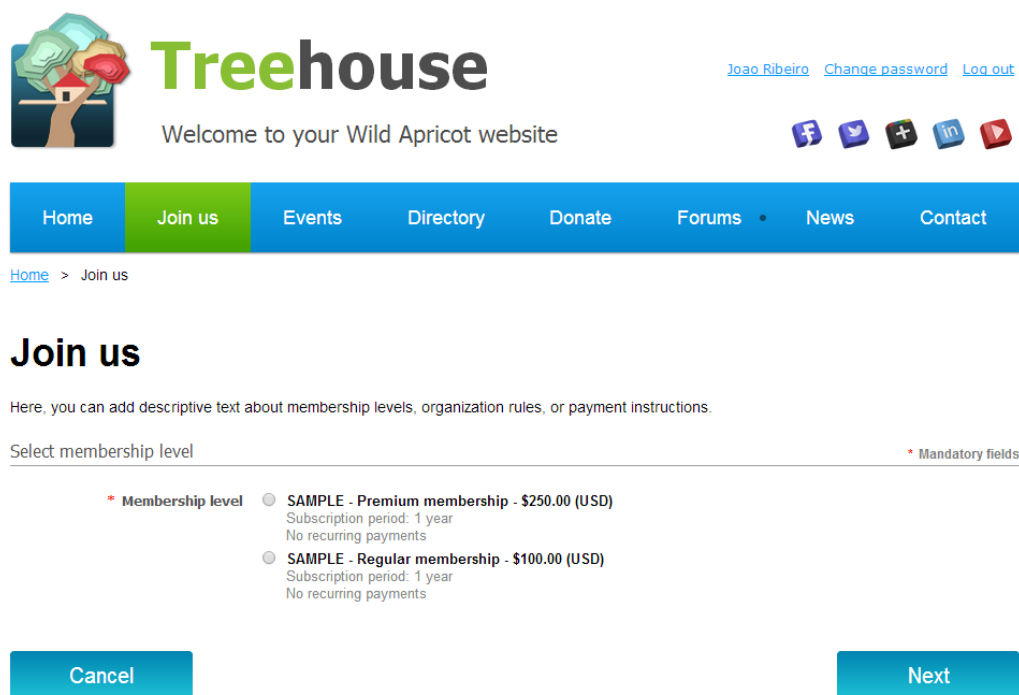


Figura 2.3: Ecrã da aplicação web WildApricot

### 2.3.2 AssociaPro

A análise que é apresentada de seguida é baseada nas informações disponibilizadas no site desta aplicação de gestão de associados. A aplicação AssociaPro permite aos administradores efectuarem a gestão de associados de forma simples e directa, sendo possível gerir também a vertente financeira da associação, permitindo que o administrador possa consultar as quotas de cada membro, incluindo aquelas que se encontram por pagar. Pode ainda emitir os recibos referentes às quotas cujos pagamentos já foram efectuados pelo associado. Este recibo pode depois ser guardado pelo utilizador em formato pdf, a partir da sua área privada. A aplicação disponibiliza ainda informações de Contas Correntes, que permitem gerir pagamentos e recebimentos através do registo de documentos. O facto de os vários extractos se encontrarem listados torna mais fácil o controlo das contas gerais da associação.

A AssociaPro oferece formas de comunicação à administração, para que possam enviar mensagens ou e-mails aos seus associados. Para que isto seja possível, as moradas e endereços de cada associado estão armazenadas numa lista. O que é enviado mais frequentemente por estes meios são avisos de quotas em atraso, votos de boas festas, convocatórias para eventos ou reuniões e pedidos de exames médicos, por exemplo.

É ainda possível trocar mensagens sem o envio de e-mails, através das mensagens privadas disponíveis na área pessoal de cada membro.

A aplicação permite ainda aos administradores colocarem num repositório próprio vários documentos que podem depois ser consultados pelos utilizadores nas suas áreas pessoais ou disponibilizados publicamente para que todos os visitantes os possam consultar.

Estes arquivos poderão conter vários tipos de informação, como um calendário que apresenta as datas de actividades e eventos da associação, permitindo o registo nos mesmos; Notícias que serão apresentadas aos utilizadores, organizadas por categorias; Docu-

mentos em pdf, que poderão conter informações importantes, como formulários ou publicações periódicas; Ficheiros multimédia que podem ser carregados pelos utilizadores ou integrados a partir do Youtube<sup>13</sup>, Vimeo<sup>14</sup> ou DailyMotion<sup>15</sup>.

Disponibilizada pela AssociaPro está também uma funcionalidade que permite criar relatórios e outros documentos. É possível imprimir cartões de sócio personalizáveis, etiquetas de morada de sócios, vinhetas que conferem o pagamento de quotas e listagens ou extractos que contém a maior parte das informações.

É ainda possível exportar relatórios e listagens para ficheiros XLS, CSV ou PDF.

A aplicação tem ainda alguns módulos adicionais que poderão dar à associação a possibilidade de aumentar as suas capacidades de gestão, apresentação e de ligação entre os associados. Estes módulos são:

- Módulo opcional que permite manter múltiplas secções da associação, separar os recibos as quotas e os sócios por secção.
- O AssociaPro Premium permite a utilização de um site modelo com a informação gerida no AssociaPro.
- Módulo de geração de referências multibanco para pagamentos.
- Módulo de gestão de formação, permitindo gerir turmas, cursos, formandos, presenças, entre outros.
- Módulo de gestão de desporto, que dá aos sócios a possibilidade de gerirem eventos desportivos em equipa, organizando os atletas das equipas, campeonatos e possíveis pagamentos.
- Loja Online, que permite aos clientes que possuam a versão AssociaPro Premium ter uma loja de comércio electrónico para que possam comercializar produtos das suas associações.
- Módulo de cópias de segurança que permite copiar todos os dados arquivados para um ficheiro XML, podendo depois ser usado no computador local ou através de uma folha de cálculo.

---

<sup>13</sup>Mais informações em: [www.youtube.com](http://www.youtube.com)

<sup>14</sup>Mais informações em: [www.vimeo.com](http://www.vimeo.com)

<sup>15</sup>Mais informações em: [www.dailymotion.com](http://www.dailymotion.com)

The screenshot displays the 'Sócios' (Members) section of the AssociaPro web application. The interface includes a top navigation bar with the logo 'associapro v4.0', the year '2012', and the user's role 'Director Assoc. Demonstração'. A sidebar on the left contains a menu with options like 'Associados e Quotas', 'Gestão', and 'Listagens e Relatórios'. The main content area shows a form for editing a member's profile. The member's name is Liliana Oliveira, and her profile picture is visible. The form is divided into tabs: 'Dados Pessoais', 'Socio-económicos', 'Familiares', and 'Outros'. The 'Dados Pessoais' tab is active, showing fields for address, birth date, civil status, and contact information. The 'Socio-económicos' tab shows membership details like 'Cargo' (Associado), 'Tipo Sócio' (Normal), and 'Quota' (Mensal). There are also checkboxes for 'Autorização' and 'Newsletter'.

Figura 2.4: Ecrã da aplicação web AssociaPro

### 2.3.3 ClubMaster

A ClubMaster é uma plataforma para gestão de associados Open-Source, comercializada sob a alçada de uma licença de Beerware<sup>16</sup>. O código é disponibilizado pela empresa, assim como uma demonstração online que é utilizada para testes — foi através desta funcionalidade oferecida ao cliente que foi feita a análise desta plataforma. De notar que esta demonstração contém apenas os módulos pré-definidos, pois como a aplicação é Open-Source é possível a qualquer programador com conhecimentos de PHP desenvolver novas funcionalidades que preencham as suas necessidades. Apesar de ser direccionado a clubes com várias equipas, o facto de poder ser alterado faz com que possa facilmente ser usado por associações ou organizações que não tenham diferentes equipas. De seguida é apresentada uma análise dos módulos principais desta aplicação.

Na versão de demonstração não é possível a um visitante do site registar-se, apenas é possível adicionar novos membros através do painel de administrador. Na criação de cada novo membro é pedido ao administrador que introduza vários dados acerca do novo membro que está a ser adicionado à plataforma. Ainda no painel de administração são colocadas à disposição da pessoa responsável várias tarefas, sendo uma delas a divisão dos utilizadores por grupos, com a respectiva criação de novos grupos.

Os diferentes grupos apresentam o número de utilizadores que pertencem a cada um. Para cada grupo é possível definir a sua localização (caso seja necessário) e o papel que os membros terão no site. Um membro poderá apenas visualizar uma lista com os nomes dos outros membros do site.

<sup>16</sup>Licença de utilização de software em que o utilizador deverá pagar uma cerveja ao criador do software



Neste módulo, os utilizadores poderão ficar a saber que tipo de eventos se passam na associação, permitindo que se possam inscrever nos eventos existentes. O Administrador terá a possibilidade de editar ou apagar um evento já criado, criar um novo evento e consultar também os utilizadores que já se encontram inscritos.

Ao criar um novo evento são pedidas várias informações ao administrador, que deverão ser preenchidas para que se torne fácil a compreensão do intuito do evento. Para além do nome e descrição de evento é ainda possível definir um número máximo de participantes, preço e a data de encerramento de inscrições. De notar que a descrição do evento permite adicionar vídeos ou fotografias.

Como membro é possível ter acesso às fotografias e vídeos disponibilizados pelos outros utilizadores, desde que seja dada a devida autorização para tal. O administrador, para além de ter acesso aos ficheiros que já se encontram na pasta e disponíveis a todos os utilizadores, pode ainda apagar ficheiros e fazer upload de novos documentos. Os documentos poderão ser de diferentes tipos, podendo ser disponibilizados para partilha pdf's importantes para todos os utilizadores da aplicação web Clubmaster.

Ao efectuar a demonstração no papel de associado, são dadas permissões para publicar novas mensagens no Blog da associação. Aqui também poderão ser apresentadas notícias e anúncios importantes aos membros. Ao criar uma nova publicação, não é possível definir que grupos de utilizadores terão acesso à mesma. As funcionalidades deste módulo não sofrem grandes alterações ao iniciar o processo com o papel de administrador, a não ser a possibilidade de editar ou apagar publicações já existentes.

Esta aplicação disponibiliza à associação uma loja onde poderão comercializar diversos produtos, físicos ou não, como subscrições premium, canecas, t-shirts, entre outros. Ao administrador é possível adicionar novos produtos e novas categorias para os mesmos, ou ainda consultar a listagem de todas as encomendas efectuadas até ao momento. A loja é dos módulos que mais opções de administrador oferece, permitindo adicionar novos produtos e novas categorias de produtos. É ainda possível ver a listagem das várias encomendas efectuadas até ao momento.

Esta funcionalidade permite ainda a criação de cupões de um determinado valor, sendo que cada um tem associado um id que é utilizado para o identificar, podendo ter também um número máximo de utilizações. Neste painel é ainda possível afirmar que o cupão já passou de validade e consultar um log de todas as utilizações referentes a um cupão. Os métodos de pagamento podem ser geridos a partir da aplicação web. No entanto para poder activar ou desactivar um determinado método de pagamento, é necessário alterar um ficheiro de configuração presente no servidor.

Para cada artigo poderá ser definido o seu custo e o tipo de envio, podendo ser com portes grátis, com desconto ou pagos em totalidade, por exemplo. É ainda possível alterar o estado das encomendas, indicando se uma encomenda se encontra paga, entregue ou cancelada. A prioridade de cada encomenda também pode ser alterada.

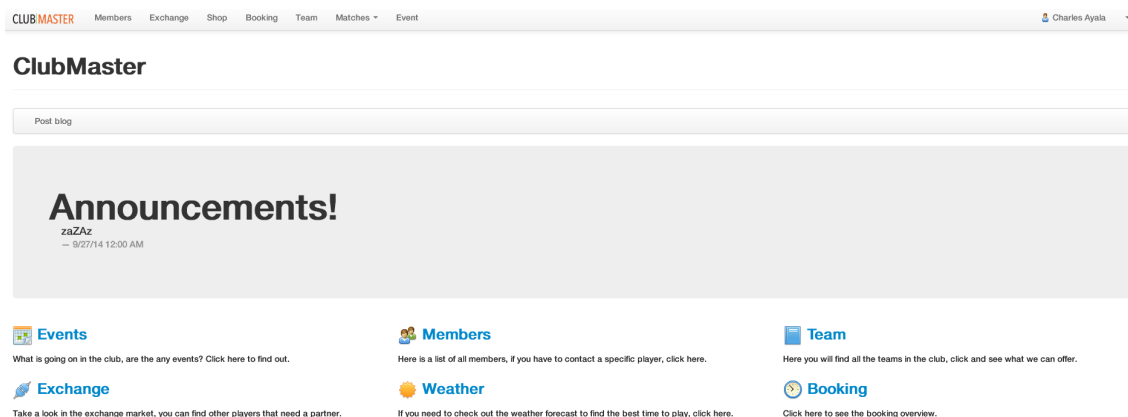


Figura 2.5: Ecrã da aplicação web ClubMaster

### 2.3.4 Silkstart

Todos os dados que são processados por esta aplicação são armazenados numa base de dados centralizada que permite consolidar toda a informação relativa a uma organização e aos seus associados. De seguida é apresentada uma análise aos principais módulos do sistema.

Para um novo membro é fácil e intuitivo proceder à sua filiação perante a associação. Após clicar no botão *Join Now* é pedido ao novo membro que introduza um leque de informações acerca de si próprio. Apenas algumas são obrigatórias, sendo que os dados facultativos serão pedidos mais tarde. Cada registo ficará completo após o associado assinar um dos planos de filiação disponíveis, cada um com um custo definido pelo administrador. O pagamento dos planos poderá ser efectuado de várias formas, por exemplo, através de métodos como o paypal<sup>17</sup>, multibanco, transferência bancária ou presencialmente. Para cada tipo de registo é possível definir os seus privilégios, sendo que as funcionalidades a que cada tipo de filiação tem acesso podem ser filtradas de perfil para perfil.

A possibilidade de criar eventos que serão apresentados num calendário é também uma realidade, sendo que cada evento tem uma página individual com as informações a ele referentes. É possível associar a um evento várias informações, sendo que na página individual será mostrada a localização do mesmo através do Google Maps, uma imagem representativa e preços de vários níveis. É ainda possível aos utilizadores adicionarem comentários acerca de cada evento. O evento poderá ser promovido através do envio de convite por correio electrónico, por grupos de pessoas ou ainda por convite directo a todas as pessoas que participaram num evento anterior. Cada evento poderá ter um custo e o pagamento do mesmo pode ser efectuado através da aplicação, podendo ser possível comprar mais que uma entrada, introduzindo os dados individuais de cada um dos participantes.

A administração poderá manter um contacto regular com os seus membros através da aplicação, pois esta permite uma comunicação activa que possa visar informar os seus associados de diversos eventos. Esta comunicação pode ser efectuada através de e-mails individuais aos seus membros ou grupos de correio electrónico. Cada e-mail pode ser customizado através de um editor HTML que coloca ainda à disposição alguns templates de comunicação, permitindo alterá-los. Poderão configurar e-mails automáticos, de forma a que sejam enviados em determinados momentos, como para felicitações de natal. A aplicação também suporta fóruns online, para que os associados possam partilhar notícias,

<sup>17</sup>Serviço que permite transacções monetárias online. Mais informações em [www.paypal.com](http://www.paypal.com).

dicas e histórias que julguem ser interessantes. Estes poderão ser criados de forma privada para que apenas determinados grupos de associados os possam visitar.

A aplicação facilita a gestão financeira da associação, facilitando ainda os diferentes pagamentos que os membros efectuam. Os pagamentos podem ser efectuados de várias maneiras, no entanto a aplicação facilita a utilização de ferramentas de pagamento online, podendo ser integrada com ferramentas como o Stripe<sup>18</sup>, Paypal, Authorize.net<sup>19</sup> ou Beanstream<sup>20</sup>, para que os utilizadores possam pagar as suas anuidades ou registos nos eventos de forma cómoda. Todas as transacções monetárias efectuadas através da aplicação geram facturas que podem depois ser enviadas por e-mail ou impressas após a criação de um documento PDF. O administrador pode ainda gerar facturas manualmente para quando são efectuadas sem qualquer registo online.

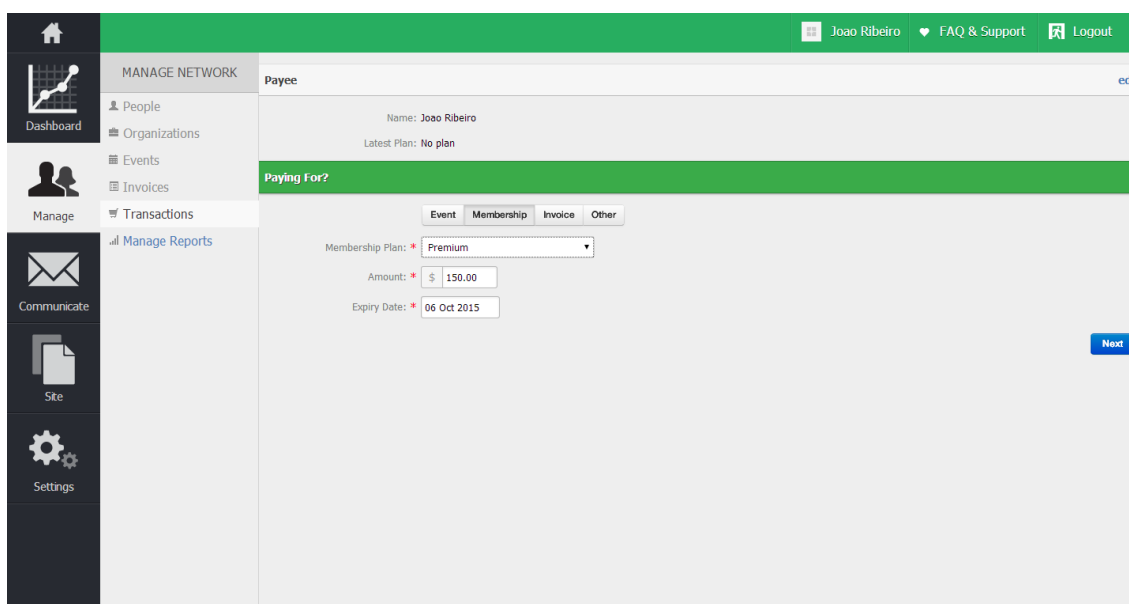


Figura 2.6: Ecrã da aplicação web SilkStart

### 2.3.5 BigTent

Esta aplicação web é gratuita desde que não seja necessário efectuar qualquer tipo de transferências entre a associação e os seus associados, caso contrário será cobrada uma taxa, por ser necessário efectuar transacções monetárias, e ainda será cobrada uma percentagem de cada transacção. O princípio desta aplicação assenta na criação de grupos. Cada grupo poderá representar uma associação, sendo definidos vários aspectos do mesmo, como por exemplo um número máximo de elementos por grupo. A página do grupo poderá ser customizada, sendo possível alterar a imagem representativa do mesmo, a cor do tema, a cor de fundo e ainda o que aparecerá na página principal.

Os administradores de um grupo, em caso de dúvidas em operar na aplicação, poderão recorrer ao seu suporte, existindo uma biblioteca de FAQs<sup>21</sup>, guias de vídeo, demonstrações

<sup>18</sup>Serviço que permite transacções monetárias online. Mais informações em [www.stripe.com](http://www.stripe.com).

<sup>19</sup>Serviço que permite transacções monetárias online. Mais informações em [www.authorize.net](http://www.authorize.net).

<sup>20</sup>Serviço que permite transacções monetárias online. Mais informações em [www.beanstream.com](http://www.beanstream.com).

<sup>21</sup>Questões colocadas mais vezes

de várias funcionalidades. Caso não consigam obter a ajuda necessária desta forma poderão mesmo entrar em contacto com a equipa que se encontra preparada para responder a todos os tipos de dúvidas através de e-mail.

É possível criar e editar tipos de filiação. Cada um dos tipos pode ter a ele associado um preço e um formulário de inscrição diferente, sendo que para cada tipo de filiação podem ser pedidos dados diferentes. Os membros podem ainda ser adicionados através do backend um a um ou importados a partir de uma spreadsheet ou do Yahoo! Groups. Cada tipo de filiação poderá também ter um custo associado e devem ser configuradas as formas de pagamento, como o paypal, cartão de crédito, cheque ou pessoalmente. É ainda possível convidar pessoas a partir de listas de e-mail, com um máximo de 100 pessoas por dia.

Para que exista uma comunicação activa entre os membros e a associação, esta aplicação permite a criação de fóruns de discussão, com capacidade para diversos tópicos. Permite ainda aos administradores criarem e editarem entradas noticiosas onde poderão partilhar informações importantes relativamente à associação. Poderão ser criadas perguntas para que a administração possa obter algum feedback acerca do seu trabalho, podendo ainda ajudar a angariar consenso para datas de eventos, entre outros. Os membros podem partilhar fotografias e vídeos, bem como ficheiros e documentos que considerem importantes para toda a comunidade, tudo isto com espaço de armazenamento ilimitado. Podem também ser enviadas newsletters<sup>22</sup> para todos os utilizadores, através de correio electrónico. Caso já existisse uma newsletter offline, apenas é necessário efectuar o upload do documento e partilhar o link com todo o grupo. Os associados poderão ainda colocar artigos numa loja online, publicitando os seus próprios negócios ou produtos, e fazer reviews de produtos que já adquiriram e se encontram publicados na loja.

Através da aplicação é possível adicionar eventos a um calendário que será posteriormente partilhado com o grupo. Desta forma torna-se fácil gerir quem irá participar no evento, consultando a lista de participantes e os pagamentos efectuados no caso de se tratar de um evento pago. É ainda possível criar um evento que se repita semanalmente ou mensalmente, por exemplo. É também possível importar o calendário para um que os utilizadores já utilizem, seja ele Google Calendar, Yahoo! ou iCal.

Após efectuar uma configuração dos métodos de pagamento, os administradores poderão seguir as transacções que são efectuadas através da plataforma. Para isso serão disponibilizadas listas com o número de identificação da factura referente a cada um dos pagamentos. A aplicação permite ainda facilitar a renovação da filiação através da configuração de e-mails automáticos, que lembre os utilizadores de que terão a anuidade para pagar ou até configurar mesmo uma renovação automática.

---

<sup>22</sup>Publicação regular para informar as pessoas que a recebem acerca de vários temas

my account ▾ my groups ▾ my calendar search help sign out

**joaoribeiro**

buzz forums news events members photos files reviews classifieds about manage ▾

---

Payment Overview

Description	Fee
Attendee	\$100
<b>Order Total:</b>	<b>\$100</b>

---

Select a Payment Method

Credit Card  PayPal

---

Pay by Credit Card

First Name  Middle  Last

Credit Card Type  Expiration Date

Credit Card Number  Security Code\*   
(CCV2)

\* The security code is the last 3 digits located on the signature stripe of your card.

Credit Card Billing Address

Street Address

Apt/Suite

City  State  ZIP Code

Phone Number

Payment Confirmation Email Address

All sales are final.

**Verify** »

Figura 2.7: Ecrã da aplicação web BigTent

### 2.3.6 YourMembership.com

Esta solução presente no mercado oferece um suporte muito pessoal. Assim que o autor se registou para uma demonstração do produto, foi contactado por telefone por uma funcionária da empresa para marcar uma reunião via GoToMeeting<sup>23</sup> para que fosse possível efectuar a demonstração da aplicação, navegando como administrador e como utilizador. Esta aplicação coloca à disposição da associação que a pretende utilizar uma página web para que os seus futuros membros se possam registar. Nesta página principal há ainda espaço para publicitar uma conferência por ano, relacionada com a associação, ou ainda uma outra secção onde poderão ser apresentados patrocinadores da associação. Aparecerão ainda notícias e informações acerca dos próximos eventos e cursos. Este registo poderá também ele ser efectuado para diferentes tipos de filiação. No caso que me foi demonstrado, os dois tipos de filiação eram para membros empresariais ou individuais. Para cada um destes tipos poderá ainda haver diferentes planos com diferentes custos e opções de renovação, como renovação automática, ou não. Inicialmente, ao novo membro são apenas pedidos um nome de utilizador, uma palavra passe e um e-mail. Numa fase mais adiantada será necessário introduzir uma variedade de informações que poderão melhorar a comunicação que a associação terá com os seus membros. Cada membro poderá visitar a página de perfil de outros membros, onde poderá encontrar várias informações, como ligações para as páginas de redes sociais. No backend é possível ao administrador efectuar algumas operações sobre os membros já existentes, como aprovar, ou rejeitar, associados cujo registo se encontre em estado pendente. A aplicação permite tanto a utilizadores como a administradores a consulta e impressão de documentos importantes; aos associados

<sup>23</sup>Mais informações em [www.gotomeeting.com](http://www.gotomeeting.com)

é possível imprimir o seu cartão de sócio, cujos campos podem ser alterados; aos administradores é possível imprimir listagens com informações dos associados, como moradas ou etiquetas para envio de informações por correio. Esta solução permite ainda consultar todas as facturas geradas pelo site, podendo filtrar por utilizador. É possível alterar os campos das facturas para que estas se adaptem às necessidades da associação cliente. Estas facturas dizem respeito a produtos adquiridos na loja, participações em eventos ou donativos. A aplicação permite ainda exportar os dados financeiros da associação para QuickBooks<sup>24</sup>.

É possível a um administrador criar eventos recreativos ou educacionais. Ambos poderão ter associado um ou mais tipos de registo, podendo cada um depois ter associado diferentes custos ou regalias, podendo dividir por dias ou por registos individuais ou de grupo. Ao seleccionar um tipo de registo, os lugares ficam reservados para o associado, tendo este 10 minutos para completar o seu registo. Caso não seja capaz de o efectuar nesse tempo os lugares são novamente listados como livres. No final do formulário de inscrição existe uma opção de gravar e adicionar outro participante. Ao seleccionar esta opção serão então pedidas as informações acerca desse participante. Cada associado pode, na sua área pessoal, consultar todos os eventos em que já participou e aqueles em que está registado. Pode ainda exportar o calendário de eventos para Google Calendar ou iCal.

A aplicação disponibiliza ainda uma funcionalidade que poderá ser activada, ou não, pelos administradores do site. Esta poderá ajudar a associação a ter mais uma fonte de rendimentos disponível a partir de qualquer local. Trata-se de uma loja online onde podem ser comercializados vários produtos da própria associação, anúncios no site da organização ou ainda a possibilidade de publicitar ofertas de emprego a partir da aplicação online.

---

<sup>24</sup>Software de gestão financeira. Mais informações em [www.quickbooks.intuit.com/](http://www.quickbooks.intuit.com/)

american association of PROFESSIONALS

Contact Us Your Cart Sign In Join AAP Enter search criteria... TRADUZIR

About Advocacy Calendar Professional Development Collaborate

Join us in **St. Petersburg, FL** for our 2014 Annual Conference

**2014 ANNUAL CONFERENCE**

REGISTER TODAY

**CONNECT. ENGAGE. GROW.**

Welcome to the website of the American Association of Professionals. AAP is the authority and advocate for association executives throughout the nation. We hope you will explore the various features our site has to offer. We created this online space for you to create a dynamic member profile, collaborate on projects, share your ideas and expertise- and, most importantly, connect with your fellow members.

2014 Annual Conference St. Petersburg, FL

Professional Career Center Engage Today!

**MEMBER LOGIN**

SIGN IN

Forgot your password?  
Haven't registered yet?

Figura 2.8: Ecrã da aplicação web YourMembership.com

### 2.3.7 SoftManagement

Esta aplicação é diferente de todas as outras, pois é uma aplicação offline, o que faz com que perca muitas das funcionalidades que serão necessárias a uma associação que pretende estar acessível a partir de qualquer lado. As funcionalidades disponibilizadas por esta aplicação representam uma vertente muito mais técnica da gestão de uma associação. É possível adicionar fornecedores e criar listas de compras a efectuar, adicionar clientes e consultar uma lista de vendas efectuadas. É também possível adicionar produtos e serviços à base de dados. A aplicação permite adicionar sócios novos, no entanto apenas um administrador poderá adicionar um sócio e este necessita de estar a utilizar o computador central da associação, onde a aplicação se encontra instalada. Ao adicionar um novo sócio são pedidos vários dados, como o número de contribuinte, o nome e as moradas, real e electrónica. Na área de sócios é possível consultar uma listagem de sócios e das suas quotas, filtrando por tipos de sócios ou de quotas. É ainda possível gerir a facturação dos produtos que o sócio já adquiriu e também gerir a quotização do mesmo. Na área de Tesouraria é possível emitir recibos e comprovativos de pagamento, adicionar bancos e caixas. É também possível efectuar consultas de todos estes documentos. É possível notar que o facto de se tratar de uma aplicação desktop retira muitas das funcionalidades necessárias neste projecto. Esta análise foi efectuada para que se possa perceber que realmente o desenvolvimento de uma aplicação web se torna a melhor opção a tomar.

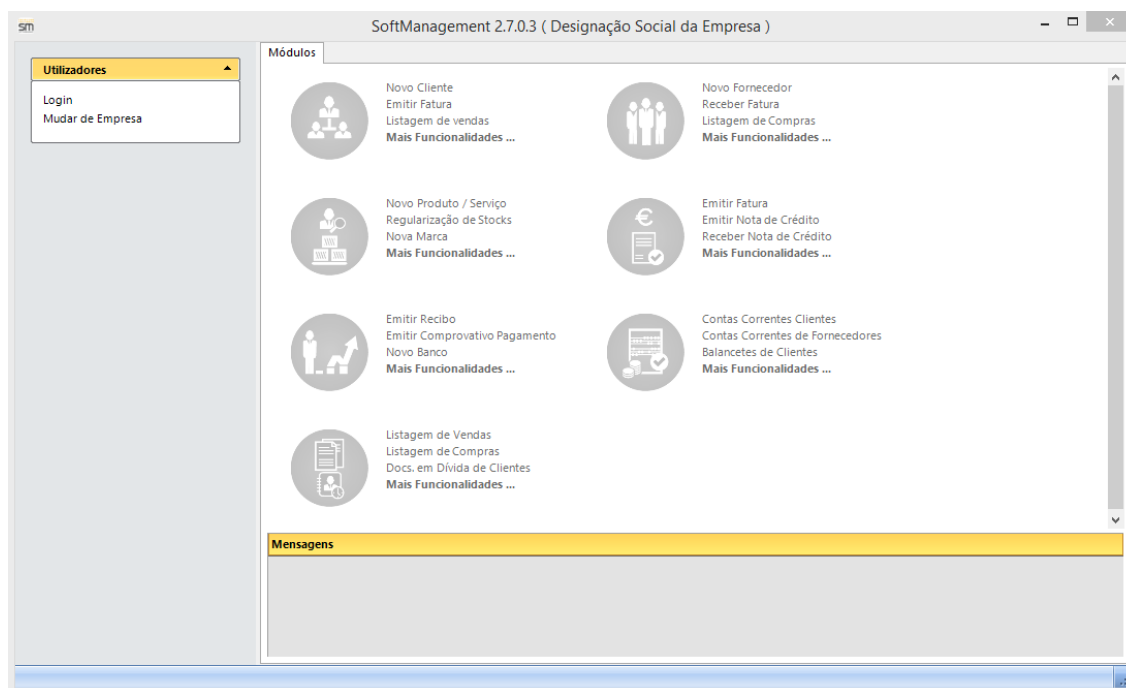


Figura 2.9: Ecrã da aplicação web SoftManagement

### 2.3.8 Discussão e Conclusões

Nesta secção é apresentada uma discussão acerca de todas as ferramentas ou aplicações analisadas, comparando-as directamente e apresentando uma conclusão sobre as funcionalidades que foram incluídas no trabalho final desta dissertação de mestrado.

De entre todas estas aplicações existem funcionalidades que apenas algumas apresentam e outras que são comuns a todas elas. Estas diferenças podem ser consultadas na tabela 2.2.

Começando pelas funcionalidades que todas têm em comum, contam-se o formulário de registo, o armazenamento das informações do novo associado, a criação e o registo em eventos e, por último, o pagamento e gestão de quotas.

A criação e o registo em eventos não eram funcionalidades pensadas inicialmente para o projecto desta dissertação de mestrado, no entanto é reconhecido o valor que estas poderão acrescentar ao projecto e, após uma discussão com o representante da AISTI, foi decidido integrar também estas funcionalidades na aplicação web.

Quanto às funcionalidades do foro financeiro, há que reconhecer que são um dos focos deste projecto. Foi possível observar as diferentes formas de funcionamento das aplicações no que diz respeito a este campo, sendo assim possível tomar uma decisão de como devem funcionar neste projecto. Algumas das aplicações analisadas permitem apenas pagamentos sem recurso a aplicações online, mas a maioria permite que os associados efectuem o pagamento da sua anuidade através de aplicações como o *paypal*. O projecto foi desenvolvido de forma a aceitar duas formas de pagamento: Pagamento offline, através de transferência bancária que permite o upload do comprovativo de pagamento, o que estranhamente nenhuma das aplicações analisadas permitia; Pagamentos online serão implementados com recurso à aplicação web *paypal*.

No entanto, a ausência da funcionalidade que nos permite efectuar o upload de um comprovativo de pagamento é um factor que acaba por aumentar a complexidade do processo de registo de novos associados, indo assim contra o que se espera desta aplicação



web para gestão de associados: que torne mais fácil todo o processo de movimentações de dinheiro.

Ainda relacionadas com as finanças da associação estão o pagamento e a gestão de quotas, que em todas as aplicações analisadas funcionam de forma idêntica, pois permitem aos associados que efectuem o pagamento das anuidades. Os administradores das associações podem posteriormente gerir os pagamentos de cada um dos associados, de forma a ver os detalhes de cada umas das transacções monetárias. De entre as sete aplicações analisadas apenas quatro geram recibos em formato digital de forma automática. Esta é também uma funcionalidade que se encontra presente no projecto desta dissertação de mestrado, uma vez que é bastante importante que cada um dos associados tenha os recibos na sua posse rapidamente e em formato digital.

Apesar da possibilidade de gerar etiquetas para o envio de cartas e gerar cartões de membro só estar implementada em duas das aplicações analisadas, considerou-se que estas funcionalidades eram importantes para a realização deste projecto, marcando presença na aplicação web de gestão de associados a ser desenvolvida. Uma funcionalidade que foi também integrada no projecto é a de possibilitar a importação de associados a partir de uma folha de cálculo, bem como a exportação dos contactos já existentes para possível visualização offline.

As funcionalidades implementadas foram decididas em conjunto com o representante da AISTI. Algumas funcionalidades, como a partilha de notícias, os fóruns de discussão e a possibilidade dos utilizadores efectuarem donativos à associação, não foram implementadas no âmbito desta dissertação de mestrado. Foram desenvolvidas algumas funcionalidades extra, como a troca de mensagens privadas entre utilizadores, a possibilidade de efectuar o login com o Facebook<sup>25</sup> ou linkedIn<sup>26</sup> e ainda a possibilidade de um membro tirar a sua foto de perfil no momento através da sua WebCam.

---

<sup>25</sup>Mais informações em [www.facebook.com](http://www.facebook.com)

<sup>26</sup>Mais informações em [www.linkedin.com](http://www.linkedin.com)

**Tabela 2.2:** Tabela comparativa de concorrência

Requisitos	WA	AP	CM	SS	BT	YM	SM
O Próprio utilizador pode efectuar o seu registo	✓	✓	✓	✓	✓	✓	
Permitir vários níveis de filiação	✓	Opc	✓	✓	✓	✓	
Armazena informações de utilizador	✓	✓	✓	✓	✓	✓	✓
Gestão de quotas	✓	✓		✓	✓	✓	✓
Permite pagamento de anuidade	✓	✓	✓	✓	✓	✓	✓
Gera etiquetas para envio de cartas		✓				✓	
Gera cartões de membro		✓				✓	
Emite recibos	✓	✓		✓	✓	✓	✓
Gera recibos em PDF	✓	✓		✓		✓	
Permite o upload de comprovativo de pagamento							
Permite efectuar o pagamento online	✓	✓		✓	✓	✓	
Permite criação de eventos	✓	✓	✓	✓	✓	✓	
Permite registo em eventos	✓	✓	✓	✓	✓	✓	
Permite pagamento de eventos	✓	✓		✓	✓	✓	
Permite partilha de notícias	✓	✓	✓		✓	✓	
Tem fóruns de discussao	✓			✓	✓		
Permite Donativos	✓					✓	

WA - Wild Apricot — AP - AssociaPro — CM - ClubMaster — SS - SilkStart — BT - BigTent — YM - YourMembership.com — SM - SoftManagement

Após esta análise de concorrência foi efectuada uma reunião de forma a efectuar o levantamento de requisitos. Este levantamento poderá ser consultado na secção 3.1.

# Capítulo 3

## Descrição do Sistema

Neste capítulo é apresentada a descrição do sistema, onde são descritos diversos factores importantes para o desenvolvimento do sistema. Inicialmente é apresentado o levantamento de requisitos, onde se podem consultar as tabelas de requisitos e as restrições do sistema, de seguida apresenta-se a descrição de casos de uso e os respectivos diagramas UML de casos de uso. Em terceiro lugar é descrita a arquitectura do sistema através de um diagrama de componentes e um diagrama de entidade e relacionamentos utilizado para descrever a base de dados do sistema.

### 3.1 Análise de Requisitos

O levantamento dos requisitos do sistema foi efectuado em reuniões realizadas com um representante da AISTI, onde se compreenderam também quais os objectivos para o projecto. Os requisitos são a base para qualquer projecto, definindo o que os *stakeholders* de um novo sistema necessitam dele.[22].

#### 3.1.1 Requisitos Funcionais

De seguida é possível consultar as tabelas de requisitos elaboradas para esta dissertação de mestrado. Na tabela 3.1 encontram-se os requisitos relacionados com as interacções dos associados, e na tabela 3.2 encontram-se aqueles que dizem respeito à geração de documentos em PDF. Todos os requisitos relacionados com a área financeira do site podem ser encontrados na tabela 3.3, e por fim, será possível encontrar na tabela 3.4 os requisitos extra, que são também os únicos que não têm prioridade alta.

#### Requisitos de associados

Todos os requisitos presentes na tabela 3.1 apresentam uma prioridade alta, pois são necessários a toda a gestão da plataforma e ao registo de novos membros perante a associação. Os RQ06 e RQ07 são importantes de forma a que sejam enviadas informações importantes acerca das pequenas alterações, o primeiro para que o novo associado saiba que passos deverá tomar para proceder ao pagamento da sua anuidade e o segundo para que o Administrador tome conhecimento do novo registo perante a associação. Após conversas mais avançadas com o representante da AISTI o RQ08 viu a sua prioridade reduzida para L - Baixa, devido ao facto de não se tornar fulcral para o funcionamento da aplicação. Apesar de não ter sido testado com os dados oficiais dos associados da AISTI, a aplicação encontra-se já preparada para satisfazer o RQ08, podendo importar os associados existentes a partir de um documento Excel.

### **Requisitos de geração de documentos**

De todos os requisitos apresentados na tabela 3.2 apenas um deles não tem a prioridade alta. Os outros foram definidos pelo representante da AISTI como muito importantes para a aplicação web de gestão dos seus associados. O último é representado por uma prioridade Média pois apesar de ser uma boa forma de consultar estatísticas acerca da utilização da aplicação, não se torna uma peça fundamental do motor para a gestão de associados.

### **Requisitos financeiros**

Na tabela 3.3. Quanto ao requisitos financeiros todos, à excepção do RQ16, têm uma prioridade alta, pois sem eles não seria possível que a aplicação surtisse efeito na forma como a associação trabalha. O facto de permitir que sejam efectuados pagamentos através do paypal é uma melhoria em relação a ter apenas uma opção, a transferência bancária. O RQ16 tem uma prioridade baixa, pois depende de outros requisitos com a mesma prioridade, estes serão falados de seguida.

### **Requisitos extra**

Na tabela 3.4 podem ser consultados os requisitos extra, que visam acrescentar algo mais à aplicação. No caso da criação, gestão e registo em eventos torna-se um foco secundário, pois para as conferências a AISTI tem já um sistema próprio separado deste. No entanto, esta funcionalidade, caso seja implementada, poderá adicionar muito à aplicação web para gestão de associados. A aplicação encontra-se também preparada para efectuar um backup da base de dados, satisfazendo o RQ21 com um click por parte do administrador.

**Tabela 3.1:** Tabela de requisitos de associados.

ID	Descrição	Dep	P
RQ01	Deverão poder registar-se como associados.		H
RQ02	Deverão poder pagar anuidade.		H
RQ03	Ao efectuar o registo e respectivo pagamento na hora, o associado deverá ter o seu estado alterado para “Auxiliar” de forma automática.		H
RQ04	Ao efectuar o pagamento do registo mais tarde, o associado deverá ter o seu estado alterado para “Pendente”.		H
RQ05	O Administrador deverá poder alterar o estado dos associados.		H
RQ06	Deverá ser enviado um e-mail com os dados de pagamento, ao associado que selecione efectuar o pagamento mais tarde.	RQ01	H
RQ07	Deverá ser enviado um e-mail ao administrador para lhe dar conhecimento de que há um novo associado “Pendente”.		H
RQ08	Deverá ser possível importar associados a partir de um documento Excel.		H

**ID** - Identificação do Requisito — **Dep** - Dependência do Requisito — **P** - Prioridade (H - High, M - Medium, L - Low)

**Tabela 3.2:** Tabela de requisitos de geração de documentos.

ID	Descrição	Dep	P
RQ09	Deverá ser possível ao associado gerar o seu cartão de sócio para impressão.		H
RQ10	Deverá ser possível ao administrador gerar uma lista de etiquetas de associados (podendo filtrar aqueles cujas informações serão impressas).		H
RQ11	Deverá ser possível gerar cartas em série para envio aos associados.		H
RQ12	Deverá ser possível gerar um documento com resultados de consultas parametrizadas.		H

**ID** - Identificação do Requisito — **Dep** - Dependência do Requisito — **P** - Prioridade (H - High, M - Medium, L - Low)

**Tabela 3.3:** Tabela de requisitos financeiros.

ID	Descrição	Dep	P
RQ13	Deverão ser emitidas facturas para pagamento.		H
RQ14	Deverão ser emitidos recibos de pagamento.		H
RQ15	Deverá ser permitido o upload de um comprovativo de pagamento.		H
RQ16	Deverá ser possível efectuar o pagamento de registos em eventos.	RQ20	H
RQ17	Deverá ser emitido um aviso aos associados cuja anuidade se encontre prestes a expirar.		H
RQ18	Anualmente deverão ser colocados no estado “Congelado” todos os associados cuja anuidade não foi paga no ano anterior.		H

**ID** - Identificação do Requisito — **Dep** - Dependência do Requisito — **P** - Prioridade (H - High, M - Medium, L - Low)

**Tabela 3.4:** Tabela de requisitos extra.

ID	Descrição	Dep	P
RQ19	Deverá ser possível ao administrador criar e gerir eventos.		L
RQ20	Deverá ser possível aos associados efectuarem o registo em eventos.	RQ19	L
RQ21	Deverá ser possível efectuar um backup da base de dados.		M
RQ22	Deverá ser possível aos associados trocarem mensagens privadas entre si.		L

**ID** - Identificação do Requisito — **Dep** - Dependência do Requisito — **P** - Prioridade (H - High, M - Medium, L - Low)

### 3.1.2 Requisitos Não Funcionais

Existe um leque de requisitos não funcionais que se deve ter em conta no desenvolvimento da aplicação:

- **Performance**

- *Tempo de resposta* - O tempo de carregamento da aplicação deverá ser inferior a 20 segundos.
- *Tempos de processamento* - O tempo de processamento de informações deverá ser inferior a 10 segundos.
- *Número de transacções* - A aplicação web deverá poder suportar 10.000 utilizadores no total, e mantendo as características pedidas pelos requisitos não funcionais anteriores.
- *Horas online* - A aplicação deverá apresentar um uptime de aproximadamente 24h por dia, exceptuando apenas os tempos em que é necessário efectuar manutenção ao servidor.

- **Manutenção**

- *Padrões de desenvolvimento* - Para o desenvolvimento do código existiam diversas opções, seleccionando-se Ruby on Rails. A análise às várias ferramentas pode ser consultada no capítulo refsec:choices. Assim sendo, os padrões de desenvolvimento serão adaptados à chama “The Rails Way” sendo que, seguindo este padrão, o entendimento do código será facilitado e cada elemento necessário ao desenvolvimento encontra-se num local específico do projecto, facilitando a manutenção do mesmo.
- *Padrões de arquitectura* - A arquitectura será definida segundo o modelo MVC, de forma a que os vários componentes estejam separados e organizados da melhor maneira.

- **Recuperação**

- *Tempos de backup* - O backup dos dados armazenados na base de dados deve ser efectuado em menos de um minuto.

- **Segurança**

- Os utilizadores sem permissões de administrador não devem poder aceder à área destinada para a administração.
  - Ao lidar com pagamentos as informações críticas não devem ser manipuladas pela aplicação web. Neste caso são tratadas pelo paypal que apenas devolve o estado do pagamento e os respectivos dados não sensíveis, como o nome e a morada do utilizador.
- Há ainda outras questões de segurança que devem ser tidas em conta durante o desenvolvimento de uma aplicação web, como:
    - Sessões de utilizadores - As sessões permitem às aplicações web manter um registo do estado de um determinado utilizador, para que possam ser mostradas apenas páginas para as quais esse utilizador tem permissões de acesso. No rails é guardada a hash de sessão de cada utilizador com um método existente para o efeito.[23]
    - Cross-Site Request Forgery - Este ataque consiste em efectuar pedidos através de um site externo à aplicação. A framework Ruby on Rails efectua a protecção a estes pedidos no background da execução da aplicação web.[23]
    - Injecções - As injecções de SQL são uma das vulnerabilidades mais exploradas por utilizadores com intenções malignas, no entanto a framework Ruby on Rails efectua protecção contra esse tipo de ataque. Apenas os parâmetros que são indicados pelo programador são aceites num pedido, por exemplo, se num pedido apenas devem ser transportados os campos “Nome” e “Idade”, se houver um campo não autorizado de “Email”, será ignorado. Se forem utilizados os métodos disponibilizados pela framework não terá que haver preocupações com injecções de código SQL maligno.[23]

- **Portabilidade**

- A aplicação web deverá ser acedida a partir de dispositivos de vários tamanhos, sendo possível graças à propriedade responsive que o design da aplicação irá apresentar aos utilizadores.

- **Compatibilidade**

- A aplicação Web deverá ser compatível entre os vários Browsers. Para conseguir isso são usadas várias frameworks que permitem que as funcionalidades sejam compatíveis com as diferenças internas entre os vários browsers do mercado.



Alguns dos requisitos não funcionais não puderam ser verdadeiramente medidos no momento de escrita desta dissertação de mestrado, devido ao facto de esta não estar a ser utilizada por um número elevado de associados. No entanto a plataforma encontra-se já preparada para efectuar as medições necessárias através da ferramenta New Relic, mostrando estatísticas importantes para facilitar a manutenção e medição da performance da aplicação web.

### **3.1.3 Restrições**

#### **Restrições de Negócio**

- Deverão existir vários níveis de utilizador, um para cada categoria de associado.
  - Fundador
  - Efectivo
  - Auxiliar
  - Convidado
  - Honorário
  - Pendente
  - Congelado

#### **Restrições Técnicas**

- A Base de Dados deverá ser MySQL.
- Deverá existir perfil de administrador para que se possa fazer a gestão dos associados.
- O design da aplicação deverá estar de acordo com as cores e logótipo da AISTI.

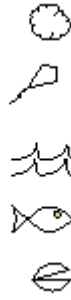
## 3.2 Casos de Uso

### 3.2.1 Descrição dos Casos de Uso

A descrição dos casos de uso será efectuada seguindo a metodologia de Allistair Cockburn [1] para a definição de casos de uso eficazes. Um caso de uso tem como função capturar o contrato entre os intervenientes e o sistema, acerca do que este terá que oferecer [1]. Os casos de uso descrevem o comportamento que o sistema deverá apresentar na interacção com os seus associados.

Cada caso de uso será descrito com recurso a alguns campos que ajudam a preencher a sua, de forma mais eficaz. Os campos que poderão apresentar alguma confusão são os seguintes:

- *Scope*: É o que define aquilo que se considera que foi desenvolvido pelos criadores de um novo sistema, ou não. Neste caso coloca frente a frente tudo o que foi desenvolvido pelo autor contra o que seria trabalho externo ao projecto. Neste caso em particular todo o sistema foi criado de raiz pelo autor, e dessa forma os casos de uso pertencem à scope do Sistema.[1].
- *Level*: Os níveis e os seus ícones referentes a cada nível podem ser consultados na figura 3.1 sendo definidos em Writing effective use cases da seguinte forma:
  - *Very Summary (Cloud)*: Este nível está reservado a casos de uso muito sumariados.
  - *Summary (Kite)*: Este nível diz respeito a casos de uso sumarizados, cujos passos normalmente são casos de uso de *Sea-level*.
  - *User-Goal (Sea-Level)*: Neste nível encaixam-se os casos de uso que podem ser efectuados por um actor apenas.
  - *Subfunction (Fish)*: Os casos de uso que se encaixam neste nível são aqueles que são necessários para completar caso de uso User-Goal.
- *Stakeholders and Interests*: Os *stakeholders* são actores externos que têm direito aos seus próprios interesses. O sistema terá que efectuar acções específicas para que esses interesses possam ser satisfeitos. Os *stakeholders* participam assim no contracto que o caso de uso descreve.
- *Minimal Guarantees*: Este campo é onde deve ser especificada a garantia mínima que o sistema irá atribuir ao caso de uso.
- *Success Guarantees*: Estas garantias representam as promessas mais básicas que o sistema faz perante os stakeholders, ganhando maior importância quando o objectivo principal do caso de uso é falhado.
- *Main success scenario*: As garantias de sucesso representam os interesses dos stakeholders que são satisfeitos após concluir correctamente a conclusão do caso de uso.



**Figura 3.1:** Níveis de casos de uso - Imagem retirada de [1]

Agora serão apresentadas as descrições dos diferentes casos de uso. Nas tabelas 3.5, 3.6, 3.7, 3.8, 3.9, 3.10 e 3.11 é possível consultar as descrições dos casos de uso que se relacionam com os requisitos de associados. Por sua vez, as tabelas 3.12, 3.13, 3.14 e 3.15 apresentam as descrições dos casos de uso referentes aos requisitos de geração de documentos. Nas tabelas 3.16, 3.17, 3.18, 3.19, 3.20 e 3.21 encontram-se as descrições relacionadas com os requisitos financeiros. Por fim, as descrições dos casos de uso dos requisitos extra podem ser encontradas nas tabelas 3.22, 3.23, 3.24 e 3.25.

Tabela 3.5: Descrição de caso de uso RQ01


<b>Caso de Uso RQ01</b>	Deverão poder registar-se como associados.	
<b>Primary Actor</b>	Visitantes	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Visitante</b> - O visitante da <b>aplicação web</b> deverá efectuar o seu registo como associado.</li> </ul>	
<b>Minimal Guarantees</b>	No caso da existência de algum problema o <b>visitante</b> será avisado.	
<b>Success Guarantees</b>	O <b>visitante</b> consegue concluir o seu registo perante a associação.	
<b>Main Success Scenario</b> <ol style="list-style-type: none"> <li>1. O <b>visitante</b> entra na aplicação para gestão de associados e selecciona a opção <i>Registar</i>.</li> <li>2. A <b>página</b> apresenta ao <b>visitante</b> um formulário que este deve preencher.</li> <li>3. O <b>visitante</b> preenche o formulário que lhe é apresentado e submete os seus dados.</li> <li>4. A <b>página</b> apresenta ao utilizador uma mensagem de sucesso.</li> <li>5. A <b>aplicação</b> envia ao novo associado um e-mail com as informações de pagamento.</li> </ol>		

Tabela 3.6: Tabela referente ao caso de uso RQ02


<b>Caso de Uso RQ02</b>	Deverão poder pagar anuidade.	
<b>Primary Actor</b>	Associados	
<b>Scope:</b> Sistema 		<b>Level:</b> User-Goal
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Associado</b> - O <b>associado</b> poderá pagar a sua anuidade.</li> </ul>	
<b>Minimal Guarantees</b>	Caso não seja possível efectuar o pagamento, o <b>associado</b> será avisado.	
<b>Success Guarantees</b>	O <b>associado</b> consegue efectuar o pagamento da sua anuidade e a aplicação coloca o seu estado como <b>auxiliar</b> .	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>associado</b> selecciona o método de pagamento, entre: <ol style="list-style-type: none"> <li>(a) Pagamento online.</li> <li>(b) Pagamento por transferência bancária.</li> </ol> </li> <li>2. O <b>associado</b> procede ao pagamento da anuidade.</li> <li>3. O <b>associado</b> recebe uma mensagem que confirma o pagamento da anuidade.</li> </ol>		
<b>Alternate Flow</b>		
<ol style="list-style-type: none"> <li>2.1. No caso do <b>associado</b> ter seleccionado <b>pagamento online</b>, e este seja confirmado, deverá passar para o caso de uso <b>RQ03</b>.</li> <li>2.2. No caso do <b>associado</b> ter seleccionado <b>pagamento por transferência bancária</b>, e este seja confirmado, deverá passar para o caso de uso <b>RQ04</b>.</li> </ol>		

Tabela 3.7: Descrição de caso de uso RQ03


<b>Caso de Uso RQ03</b>	Ao efectuar o registo e respectivo pagamento na hora, o <b>associado</b> deverá ter o seu estado alterado para “Auxiliar” automaticamente.	
<b>Primary Actor</b>	Aplicação Web	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> deverá alterar correctamente o estado do <b>associado</b>.</li> <li>• <b>Associado</b> - Deverá ver o seu estado ser actualizado correctamente pela <b>aplicação web</b>.</li> </ul>	
<b>Minimal Guarantees</b>	O <b>associado</b> receberá uma mensagem de erro caso não seja possível alterar o seu estado, a aplicação notifica também o <b>administrador</b> para que este possa alterar o estado manualmente, de acordo com o caso de uso RQ05.	
<b>Success Guarantees</b>	O estado “Auxiliar” será atribuído ao <b>associado</b> .	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. Após confirmação do pagamento online a aplicação altera o estado do <b>associado</b> para “Auxiliar”</li> <li>2. O <b>associado</b> comprova na sua área pessoal que o seu estado é “Auxiliar”.</li> </ol>		

Tabela 3.8: Descrição de caso de uso RQ04


<b>Caso de Uso RQ04</b>	Ao efectuar o pagamento do registo mais tarde, o <b>associado</b> deverá ter o seu estado alterado para “Pendente”.	
<b>Primary Actor</b>	Aplicação Web	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> deverá alterar correctamente o estado do <b>associado</b>.</li> <li>• <b>Associado</b> - Deverá ver o seu estado ser actualizado correctamente pela <b>aplicação web</b>.</li> </ul>	
<b>Minimal Guarantees</b>	O <b>associado</b> receberá uma mensagem de erro caso não seja possível alterar o seu estado, o <b>administrador</b> será também notificado para que possa proceder à alteração de acordo com o caso de uso RQ05.	
<b>Success Guarantees</b>	O estado “Pendente” será atribuído ao <b>associado</b> .	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. Após confirmação da escolha do <b>associado</b>, a aplicação altera seu estado para “Pendente”.</li> <li>2. O <b>associado</b> comprova na sua área pessoal que o seu estado é “Pendente”.</li> </ol>		

Tabela 3.9: Descrição de caso de uso RQ05


<b>Caso de Uso RQ05</b>	O Administrador deverá poder alterar o estado dos <b>associados</b> .	
<b>Primary Actor</b>	Administrador	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Administrador</b> - O <b>administrador</b> deverá conseguir alterar correctamente o estado do <b>associado</b>.</li> <li>• <b>Associado</b> - O <b>associado</b> deverá ter o seu estado alterado.</li> </ul>	
<b>Minimal Guarantees</b>	Será apresentada ao <b>administrador</b> uma mensagem de erro, caso não seja possível alterar o estado do <b>associado</b> .	
<b>Success Guarantees</b>	O estado do <b>associado</b> é alterado com sucesso.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>administrador</b> selecciona o <b>associado</b> ao qual deseja alterar o estado.</li> <li>2. O <b>administrador</b> selecciona o novo estado do <b>associado</b>.</li> <li>3. O <b>administrador</b> selecciona o novo estado do <b>associado</b>.</li> <li>4. O <b>associado</b> é notificado acerca da alteração do seu estado.</li> </ol>		



Tabela 3.10: Descrição de caso de uso RQ06


<b>Caso de Uso RQ06</b>	Deverá ser enviado um e-mail com os dados de pagamento ao <b>associado</b> que seleccione efectuar o pagamento mais tarde.	
<b>Primary Actor</b>	Aplicação Web	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Aplicação Web</b> - Deverá agendar o envio de um e-mail automático.</li> <li>• <b>Associado</b> - Receberá o e-mail com os dados de pagamento por transferência bancária.</li> </ul>	
<b>Minimal Guarantees</b>	A aplicação reenvia o e-mail caso reconheça algum problema com o envio.	
<b>Success Guarantees</b>	O <b>associado</b> recebe o e-mail com os dados para proceder ao pagamento.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. Assim que o <b>associado</b> termine o seu registo, a <b>aplicação web</b> irá enviar para o seu contacto um e-mail com informações acerca de como proceder ao pagamento da anuidade.</li> <li>2. O <b>associado</b> consulta o e-mail para proceder ao pagamento.</li> </ol>		

Tabela 3.11: Descrição de caso de uso RQ07


<b>Caso de Uso RQ07</b>	Deverá ser enviado um e-mail ao <b>administrador</b> para lhe dar conhecimento de que há um novo <b>associado</b> “Pendente”.	
<b>Primary Actor</b>	Aplicação Web	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> deverá agendar o envio de um e-mail automático.</li> <li>• <b>Administrador</b> - O <b>administrador</b> receberá o e-mail que o notifica de um novo <b>associado</b> em estado pendente.</li> </ul>	
<b>Minimal Guarantees</b>	A aplicação reenvia ao <b>administrador</b> o e-mail, caso reconheça algum problema na entrega.	
<b>Success Guarantees</b>	O <b>administrador</b> recebe o e-mail que o notifica do registo de um novo utilizador no estado “Pendente”.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. Assim que o <b>associado</b> termine o registo a <b>aplicação web</b> irá enviar ao <b>administrador</b> um e-mail, notificando-o de um novo registo.</li> <li>2. O <b>administrador</b> consulta o e-mail e aguarda que o comprovativo de pagamento seja entregue pelo <b>associado</b>.</li> </ol>		

Tabela 3.12: Descrição de caso de uso RQ09


<b>Caso de Uso RQ09</b>	Deverá ser enviado um e-mail ao <b>administrador</b> para lhe dar conhecimento de que há um novo <b>associado</b> “Pendente”.	
<b>Primary Actor</b>	Aplicação Web	
<b>Scope:</b> Sistema 		<b>Level:</b> User-Goal
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Associado</b> - O <b>associado</b> pretende efectuar o download do seu cartão de sócio para que o possa imprimir.</li> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> apresenta ao <b>associado</b> o seu cartão de sócio.</li> </ul>	
<b>Minimal Guarantees</b>	É apresentada ao <b>associado</b> uma mensagem de erro caso não seja possível gerar o cartão.	
<b>Success Guarantees</b>	O <b>associado</b> consegue gerar, guardar e imprimir o seu cartão de sócio.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>associado</b> selecciona a opção para gerar cartão de sócio.</li> <li>2. O <b>associado</b> selecciona as informações que deseja no cartão de sócio.</li> <li>3. A <b>aplicação web</b> gera o cartão de sócio no momento em que este é pedido e apresenta-o ao <b>associado</b>.</li> <li>4. O <b>associado</b> guarda o cartão gerado.</li> </ol>		

Tabela 3.13: Descrição de caso de uso RQ10


<b>Caso de Uso RQ10</b>	Deverá ser possível ao <b>administrador</b> gerar uma lista de etiquetas de <b>associados</b> , podendo filtrar aqueles cujas informações serão impressas	
<b>Primary Actor</b>	Administrador	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Administrador</b> - O <b>administrador</b> pretende gerar uma lista de etiquetas com as moradas dos <b>associados</b>.</li> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> apresenta ao <b>administrador</b> a lista de etiquetas gerada.</li> </ul>	
<b>Minimal Guarantees</b>	É apresentada ao <b>administrador</b> uma mensagem de erro caso não seja possível gerar a lista de etiquetas.	
<b>Success Guarantees</b>	O <b>administrador</b> consegue gerar e guardar a listagem de etiquetas.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>administrador</b> selecciona a opção para gerar as etiquetas.</li> <li>2. O <b>administrador</b> filtra os <b>associados</b> dos quais pretende imprimir as etiquetas.</li> <li>3. A <b>aplicação web</b> gera as etiquetas assim que estas são pedidas pelo <b>administrador</b>.</li> <li>4. O <b>administrador</b> guarda e imprime as etiquetas geradas.</li> </ol>		

Tabela 3.14: Descrição de caso de uso RQ11


<b>Caso de Uso RQ11</b>	Deverá ser possível gerar cartas em série para o envio aos <b>associados</b> .	
<b>Primary Actor</b>	Administrador	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Administrador</b> - O <b>administrador</b> pretende gerar um conjunto de cartas para envio aos <b>associados</b>.</li> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> apresenta ao <b>administrador</b> o conjunto de cartas.</li> </ul>	
<b>Minimal Guarantees</b>	É apresentada ao <b>administrador</b> uma mensagem de erro caso não seja possível gerar as cartas.	
<b>Success Guarantees</b>	O <b>administrador</b> consegue gerar e guardar as cartas pretendidas.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>administrador</b> selecciona a opção de gerar cartas.</li> <li>2. A <b>aplicação web</b> pede ao <b>administrador</b> que escreva o texto da carta.</li> <li>3. A <b>aplicação web</b> gera uma série de cartas, que pode ser impressa e enviada aos <b>associados</b>.</li> <li>4. O <b>administrador</b> imprime e envia as cartas aos <b>associados</b>.</li> </ol>		

Tabela 3.15: Descrição de caso de uso RQ12


<b>Caso de Uso RQ12</b>	Deverá ser possível gerar um documento com resultados de consultas parametrizadas.	
<b>Primary Actor</b>	Administrador	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Administrador</b> - O <b>administrador</b> pretende gerar um documento a partir de uma consulta parametrizada.</li> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> apresenta ao <b>administrador</b> o documento pedido.</li> </ul>	
<b>Minimal Guarantees</b>	É apresentada ao <b>administrador</b> uma mensagem de erro caso não seja possível apresentar o documento com os resultados.	
<b>Success Guarantees</b>	O <b>administrador</b> consegue gerar o documento que lhe apresenta os resultados da consulta parametrizada.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>administrador</b> acede à lista de <b>associados</b>.</li> <li>2. A <b>aplicação web</b> apresenta ao <b>administrador</b> a listagem de todos os membros já registados.</li> <li>3. O <b>administrador</b> selecciona os filtros que deseja aplicar à lista e pede para gerar o documento.</li> <li>4. A <b>aplicação web</b> gera e apresenta ao <b>administrador</b>, o documento pedido.</li> </ol>		

Tabela 3.16: Descrição de caso de uso RQ13


<b>Caso de Uso RQ13</b>	Deverão ser emitidas facturas para pagamento.	
<b>Primary Actor</b>	Aplicação Web	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> deverá emitir facturas para pagamento.</li> </ul>	
<b>Minimal Guarantees</b>	Caso não seja possível emitir a factura, deverão ser armazenados os dados desta, para que possa ser emitida posteriormente.	
<b>Success Guarantees</b>	A factura é emitida com sucesso.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. É requisitada à <b>aplicação web</b> uma ordem de pagamento.</li> <li>2. A <b>aplicação web</b> emite uma factura de pagamento.</li> </ol>		

Tabela 3.17: Descrição de caso de uso RQ14


<b>Caso de Uso RQ14</b>	Deverão ser emitidos recibos de pagamento.	
<b>Primary Actor</b>	Aplicação Web	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> deverá emitir um recibo cada vez que é efectuado um pagamento.</li> <li>• <b>Associado</b> - O <b>associado</b> aguarda que lhe seja entregue um recibo emitido após o pagamento.</li> </ul>	
<b>Minimal Guarantees</b>	Caso não seja possível emitir o recibo as informações são guardadas para que este possa ser emitido mais tarde e enviado por e-mail ao <b>associado</b> que efectuou o pagamento.	
<b>Success Guarantees</b>	A <b>aplicação web</b> emite o recibo que é depois entregue ao <b>associado</b> .	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. A operação de pagamento é confirmada.</li> <li>2. A <b>aplicação web</b> emite um recibo que confirma o pagamento.</li> <li>3. A <b>aplicação web</b> agenda o envio de um e-mail para o <b>associado</b> que efectuou o pagamento.</li> </ol>		

Tabela 3.18: Descrição de caso de uso RQ15


<b>Caso de Uso RQ15</b>	Deverá ser permitido o upload de um comprovativo de pagamento.	
<b>Primary Actor</b>	Associado	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Associado</b> - O <b>associado</b> efectua o upload do comprovativo de pagamento por transferência bancária.</li> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> armazena o comprovativo e notifica o <b>administrador</b>.</li> <li>• <b>Administrador</b> - O <b>administrador</b> confirma que o pagamento foi efectuado de forma correcta pelo <b>associado</b>.</li> </ul>	
<b>Minimal Guarantees</b>	Na ocorrência de alguma falha no upload, é apresentada ao <b>associado</b> uma mensagem de erro e é-lhe pedido que tente de novo.	
<b>Success Guarantees</b>	O upload é efectuado com sucesso e o <b>administrador</b> é notificado pela <b>aplicação web</b> .	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>associado</b> selecciona a opção que lhe permite efectuar o upload do comprovativo.</li> <li>2. A <b>aplicação web</b> apresenta uma janela de pop-up para seleção do documento comprovativo.</li> <li>3. O <b>associado</b> selecciona o documento.</li> <li>4. A <b>aplicação web</b> armazena o comprovativo no servidor e notifica o <b>administrador</b>.</li> <li>5. O <b>administrador</b> consulta o comprovativo e marca-o como “confirmado”.</li> <li>6. A <b>aplicação web</b> elimina do servidor todos os comprovativos marcados como “confirmado”.</li> </ol>		



Tabela 3.19: Descrição de caso de uso RQ16


<b>Caso de Uso RQ16</b>	Deverá ser possível efectuar o pagamento de registos em eventos.	
<b>Primary Actor</b>	Associado	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Associado</b> - O <b>associado</b> deverá conseguir efectuar o pagamento do seu registo em eventos.</li> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> deverá registar correctamente os pagamentos dos <b>associados</b>.</li> </ul>	
<b>Minimal Guarantees</b>	É apresentada ao <b>associado</b> uma mensagem de erro caso não seja possível processar o pagamento.	
<b>Success Guarantees</b>	A <b>aplicação web</b> processa correctamente o pagamento do <b>associado</b> , emitindo um recibo de acordo com o <b>caso de uso RQ14</b> .	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>associado</b> efectua o pagamento do registo no evento.</li> <li>2. A <b>aplicação web</b> confirma o pagamento.</li> <li>3. A <b>aplicação web</b> emite um recibo de acordo com o caso de uso RQ14.</li> </ol>		

Tabela 3.20: Descrição de caso de uso RQ17


<b>Caso de Uso RQ17</b>	Deverá ser emitido um aviso aos <b>associados</b> cuja anuidade se encontra prestes a expirar.	
<b>Primary Actor</b>	Aplicação Web	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> deverá verificar e avisar anualmente quais os <b>associados</b> cujas anuidades se encontram prestes a expirar.</li> </ul>	
<b>Minimal Guarantees</b>	Caso não seja possível efectuar a verificação, a <b>aplicação web</b> irá notificar o <b>administrador</b> acerca da falha e tentar de novo até emitir o aviso.	
<b>Success Guarantees</b>	Os <b>associados</b> são notificados de que a sua anuidade está prestes a expirar.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. A <b>aplicação web</b> verifica quais os <b>associados</b> que se encontram com a anuidade prestes a expirar.</li> <li>2. A <b>aplicação web</b> envia e-mails a todos os <b>associados</b> que se encontrem nessa situação.</li> <li>3. Os <b>associados</b> recebem a notificação e decidem se irão, ou não, renovar a sua anuidade.</li> </ol>		

Tabela 3.21: Descrição de caso de uso RQ18


<b>Caso de Uso RQ18</b>	Anualmente deverão ser colocados no estado “Congelado” todos os <b>associados</b> cuja anuidade não foi paga no ano anterior.	
<b>Primary Actor</b>	Associado	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> deverá congelar as contas dos <b>associados</b> que não pagaram a anuidade no ano anterior.</li> </ul>	
<b>Minimal Guarantees</b>	A <b>aplicação web</b> notifica o <b>administrador</b> no caso de não conseguir efectuar a operação, tentando novamente mais tarde.	
<b>Success Guarantees</b>	A <b>aplicação web</b> congela com sucesso as contas dos <b>associados</b> , notificando-os de seguida de que se encontram no estado “congelado”.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. A <b>aplicação web</b> verifica quais os <b>associados</b> que não renovaram a sua anuidade.</li> <li>2. A <b>aplicação web</b> altera o estado destes <b>associados</b> para “congelado”.</li> <li>3. A <b>aplicação web</b> agenda o envio de um e-mail que notifique os <b>associados</b> cujos estados foram alterados.</li> </ol>		

Tabela 3.22: Descrição de caso de uso RQ19


<b>Caso de Uso RQ19</b>	Deverá ser possível ao <b>administrador</b> criar e gerir eventos	
<b>Primary Actor</b>	Administrador	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Administrador</b> - O <b>administrador</b> deverá proceder à criação de novos eventos.</li> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> armazenará os dados referentes ao novo evento, adicionando-o ao calendário.</li> </ul>	
<b>Minimal Guarantees</b>	Caso seja verificado algum erro na criação do evento, a <b>aplicação web</b> irá notificar o <b>administrador</b> , pedindo-lhe que tente novamente mais tarde.	
<b>Success Guarantees</b>	O evento é criado com sucesso e listado no calendário de eventos.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>administrador</b> selecciona a opção de criar novo evento.</li> <li>2. A <b>aplicação web</b> apresenta ao <b>administrador</b> um formulário que este deverá preencher com as informações do novo evento.</li> <li>3. O <b>administrador</b> preenche e submete o formulário.</li> <li>4. A <b>aplicação web</b> armazena os dados do novo evento e lista-o no calendário, para que os <b>associados</b> o possam ver.</li> </ol>		

Tabela 3.23: Descrição de caso de uso RQ20


<b>Caso de Uso RQ20</b>	Deverá ser possível aos <b>associados</b> efectuarem o registo em eventos.	
<b>Primary Actor</b>	Associados	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Associados</b> - Os <b>associados</b> pretendem efectuar o registo nos eventos existentes.</li> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> deverá registar os <b>associados</b> correctamente.</li> </ul>	
<b>Minimal Guarantees</b>	A <b>aplicação web</b> notifica o <b>associado</b> caso não seja possível registá-lo no evento, pedindo-lhe para tentar novamente mais tarde.	
<b>Success Guarantees</b>	O <b>associado</b> regista-se com sucesso no evento e recebe um e-mail que o notifica de tal.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>associado</b> consulta o evento em que deseja efectuar o registo.</li> <li>2. A <b>aplicação web</b> apresenta ao <b>associado</b> um formulário de inscrição.</li> <li>3. O <b>associado</b> preenche e submete o formulário.</li> <li>4. A <b>aplicação web</b> agenda o envio de um e-mail ao <b>associado</b>, que o notifique do sucesso da operação e o informe das formas de pagamento.</li> </ol>		

Tabela 3.24: Descrição de caso de uso RQ21



<b>Caso de Uso RQ21</b>	Deverá ser possível efectuar um backup da base de dados.	
<b>Primary Actor</b>	Administrador	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Administrador</b> - O <b>administrador</b> deverá ordenar o backup da base de dados.</li> <li>• <b>Aplicação Web</b> - Ao ser ordenado um backup a <b>aplicação web</b> deverá efectuar uma cópia de segurança da base de dados para outras duas pastas no servidor, guardando os dados actuais.</li> </ul>	
<b>Minimal Guarantees</b>	A <b>aplicação web</b> notifica o <b>administrador</b> no caso de não conseguir efectuar o backup, pedindo-lhe que tente mais tarde.	
<b>Success Guarantees</b>	A <b>aplicação web</b> efectua com sucesso o backup da base de dados, notificando o <b>administrador</b> do sucesso da operação.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>administrador</b> selecciona a opção de backup.</li> <li>2. A <b>aplicação web</b> efectua uma cópia de segurança da base de dados para outras duas pastas no servidor.</li> <li>3. A <b>aplicação web</b> notifica o <b>administrador</b> de que o backup foi efectuado com sucesso.</li> </ol>		

Tabela 3.25: Descrição de caso de uso RQ22

<b>Caso de Uso RQ22</b>	Deverá ser possível aos <b>associados</b> trocarem mensagens privadas entre si.	
<b>Primary Actor</b>	Associados	
<b>Scope:</b> Sistema 	<b>Level:</b> User-Goal	
<b>Stakeholders and Interests</b>	<ul style="list-style-type: none"> <li>• <b>Associados</b> - Os <b>associados</b> deverão poder trocar mensagens privadas entre si.</li> <li>• <b>Aplicação Web</b> - A <b>aplicação web</b> deverá gerir a troca de mensagens privadas entre os <b>associados</b>.</li> </ul>	
<b>Minimal Guarantees</b>	Caso não seja possível entregar uma mensagem privada a <b>aplicação web</b> deverá notificar o <b>associado</b> do erro ocorrido.	
<b>Success Guarantees</b>	A <b>aplicação web</b> efectua com sucesso a gestão da troca de mensagens, permitindo aos <b>associados</b> que troquem mensagem entre si.	
<b>Main Success Scenario</b>		
<ol style="list-style-type: none"> <li>1. O <b>associado</b> que se encontra a utilizar a <b>aplicação web</b> selecciona o <b>associado</b> para o qual pretende enviar a mensagem, visualizando o seu perfil.</li> <li>2. O <b>associado</b> selecciona a opção que lhe permite enviar uma mensagem privada.</li> <li>3. A <b>aplicação web</b> apresenta ao <b>associado</b> uma página que este deve preencher com o assunto e corpo da mensagem.</li> <li>4. O <b>associado</b> preenche e envia a mensagem privada.</li> <li>5. A <b>aplicação web</b> gere a troca de mensagens.</li> <li>6. O <b>associado</b> que recebeu a mensagem abre e lê a mensagem.</li> </ol>		

### 3.2.2 Diagrama UML

Os casos de uso são utilizados para ajudar na análise de requisitos através da forma como o utilizador irá dar uso ao sistema[24]. Os diagramas UML de casos de uso têm como objectivo capturar a forma como os utilizadores interagem com o sistema representado. Estes representam as funcionalidades do sistema a partir do ponto de vista do utilizador [25]. É possível ver o diagrama de casos de uso construído para a aplicação na figura 3.2 de uma forma geral e de formas mais particulares para visitantes na figura 3.3, para associados na figura 3.4 e para administradores na figura 3.5.

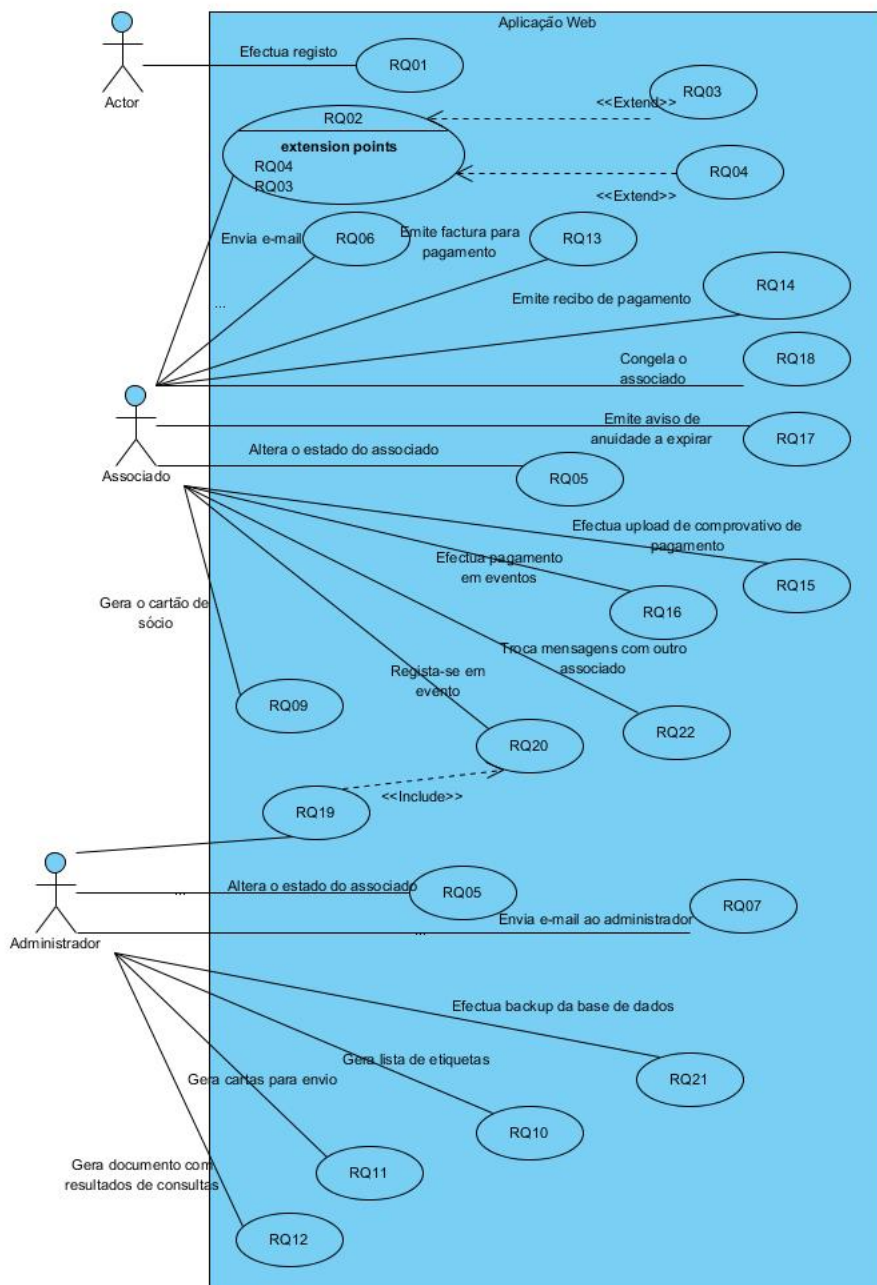


Figura 3.2: Diagrama UML de casos de uso geral



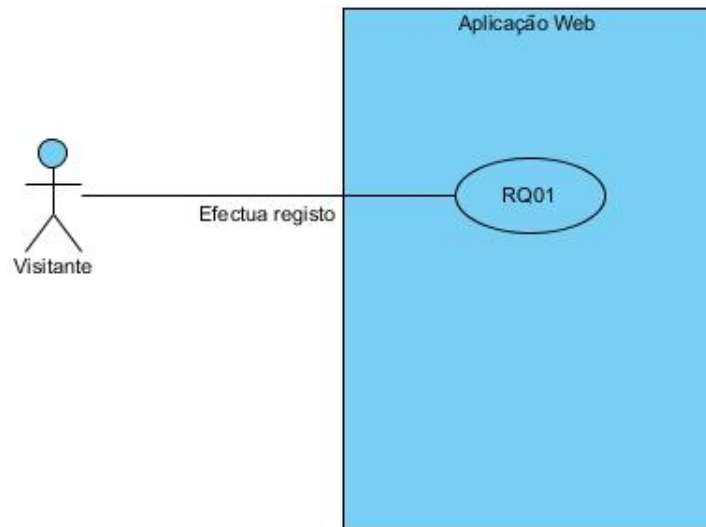


Figura 3.3: Diagrama UML de casos de uso de Visitantes

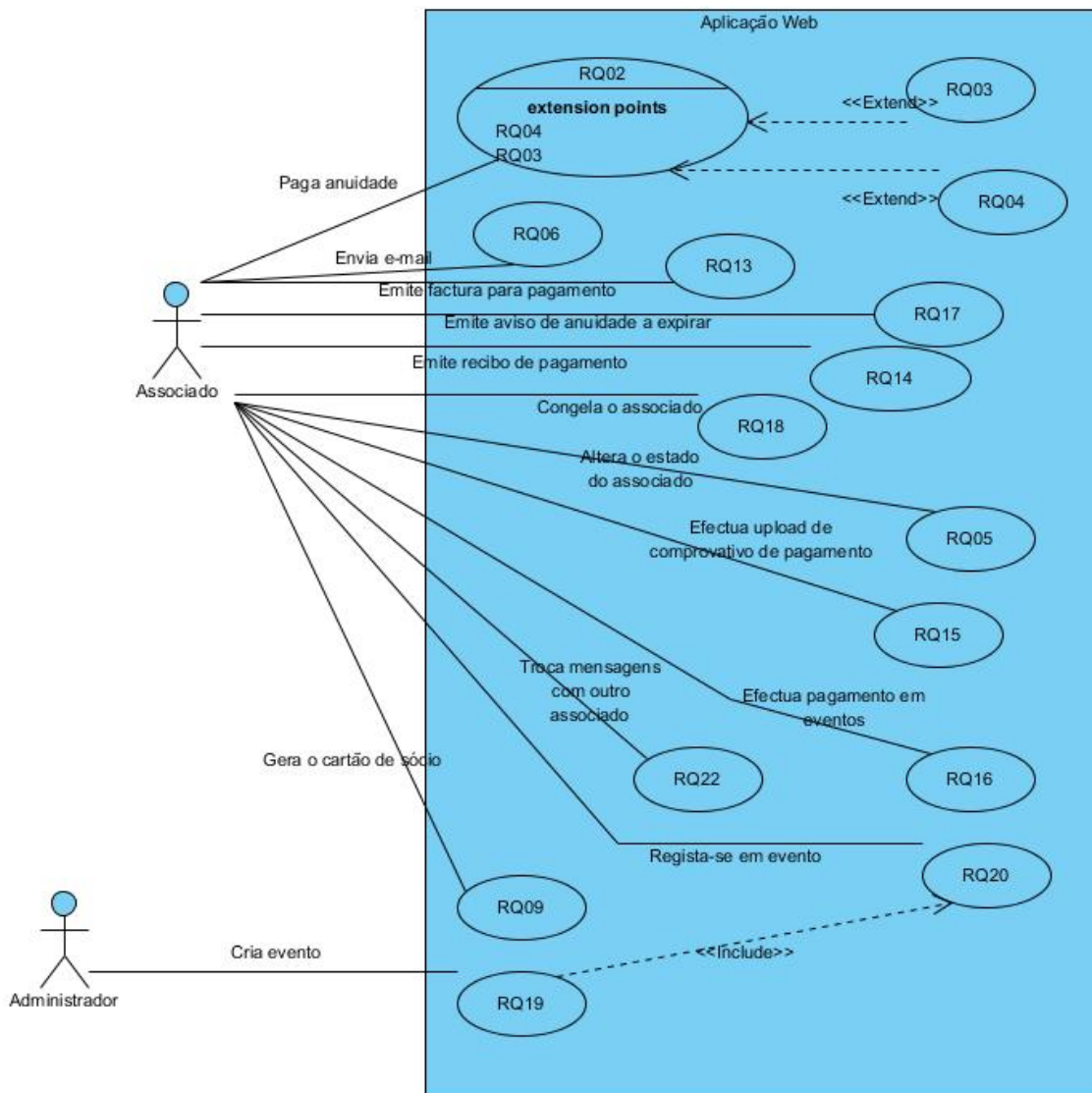


Figura 3.4: Diagrama UML de casos de uso de Associados

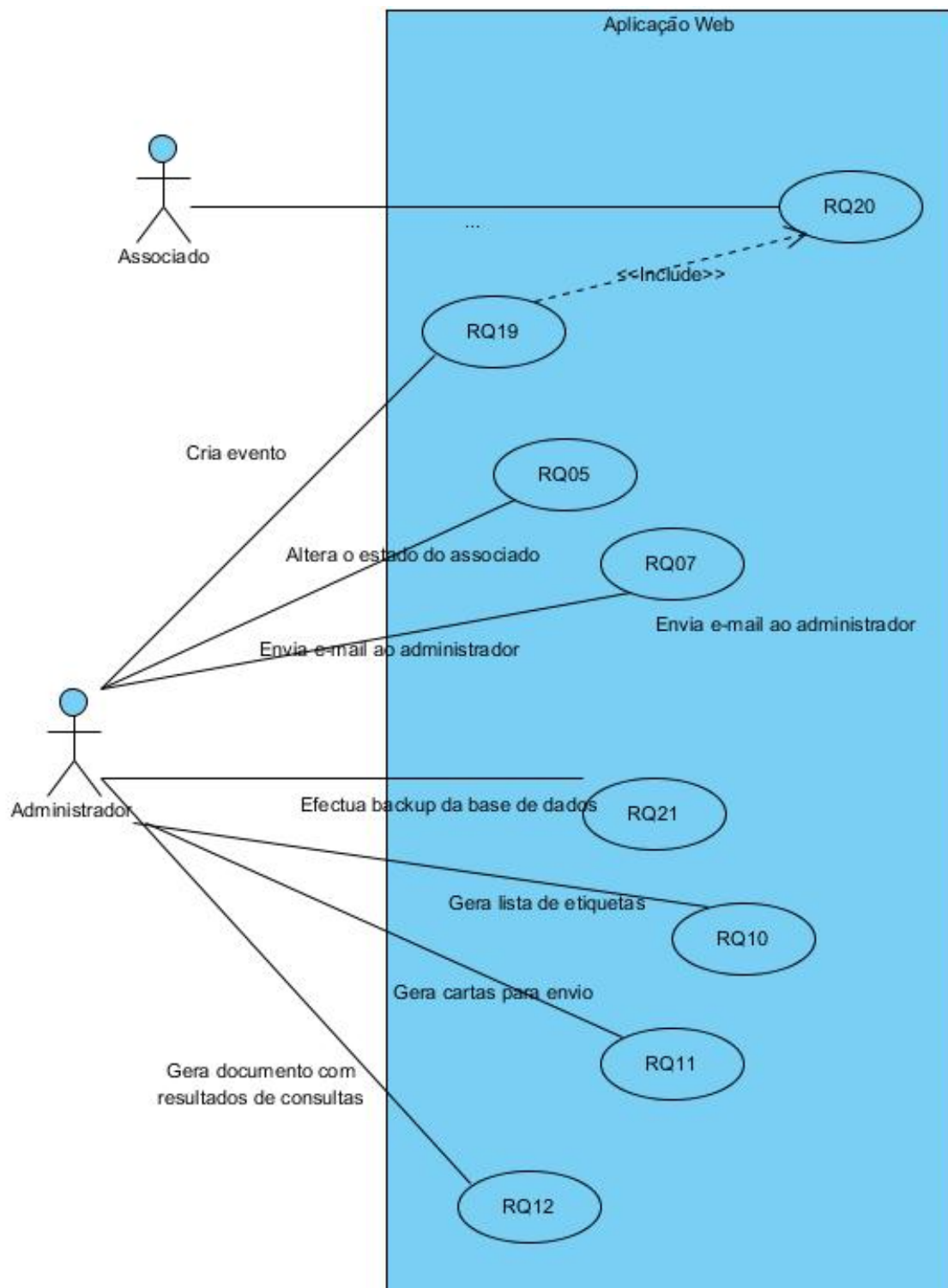


Figura 3.5: Diagrama UML de casos de uso de Administradores

## 3.3 Arquitectura do Sistema

Nesta secção é apresentada a arquitectura da aplicação web, que foi desenvolvida no decorrer da dissertação de mestrado. São apresentados os diferentes diagramas necessários para a compreensão do sistema.

### 3.3.1 Diagrama de entidade e relacionamentos

Tal como descrito nas restrições técnicas, na secção 3.1.3, a base de dados a utilizar deverá ser MySQL. Ao recorrer a uma base de dados há várias vantagens relativamente à utilização de uma folha de cálculo, uma das vantagens é que todas as transacções garantem Atomicidade, Consistência, Isolamento e Durabilidade. A informação torna-se facilmente acessível ao efectuar algumas queries, que podem retornar informação ordenada e com restrições opcionais. A base de dados foi construída seguindo a arquitectura definida no diagrama ER que pode ser consultado na figura 3.6.

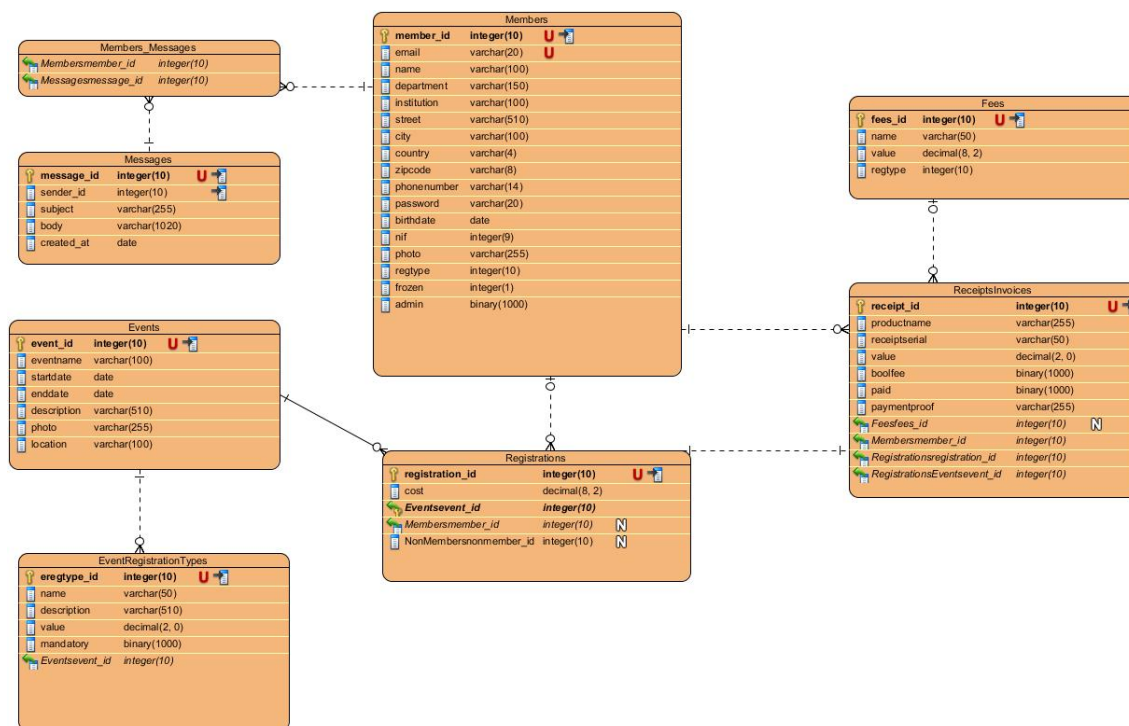


Figura 3.6: Diagrama ER da base de dados

### 3.3.2 Modelo de arquitectura

O projecto está desenvolvido segundo o modelo de arquitectura MVC, que tem como objectivo principal fazer a ligação entre os modelos humano e digital. A solução ideal cria ao utilizador uma sensação de que ele controla a informação directamente.[26] Esta solução divide-se então em três componentes principais:

- *Model*: Esta é componente responsável pelo acesso e alteração dos dados e respectivos valores. Este componente não é visível ao utilizador pois este não tem um acesso directo à base de dados.
- *View*: Este é o componente de visualização da aplicação, e dedica-se a responder aos pedidos do *controller*, apresentando da melhor forma toda a informação necessária.
- *Controller*: Este componente é aquele que efectua a ligação entre todos os componentes do MVC. Pode ser considerado o elo principal do sistema, tratando da execução de tarefas, pois efectua os pedidos de informação ao *Model* e transmite-os *View* para que possam ser apresentados.

Na imagem 3.7 encontra-se uma representação do modelo MVC.

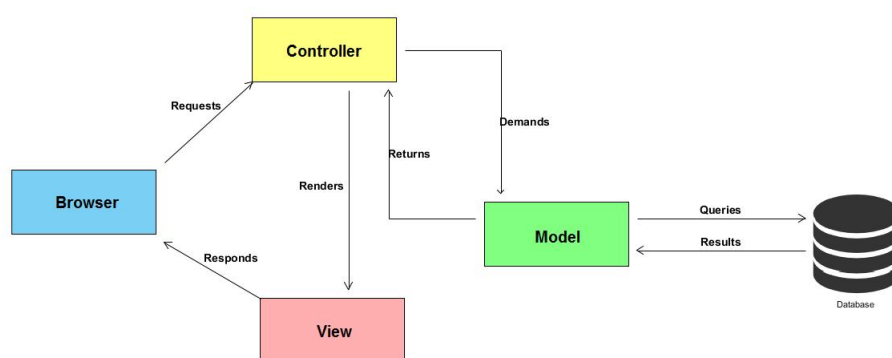
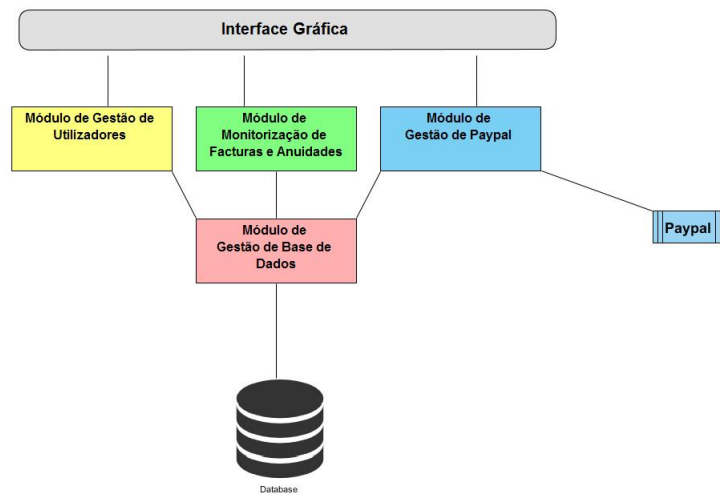


Figura 3.7: Modelo de Arquitectura MVC

Tendo estas informações podemos efectuar a relação dos três componentes com o projecto de dissertação de mestrado. O *model* diz portanto respeito a todo o sistema de armazenamento de dados na base de dados. Este modelo permite as quatro operações básicas a efectuar sobre uma base de dados: a criação, a leitura, a actualização e a destruição dos dados.

O *Controller* diz respeito a todas as acções que são processadas entre a *View* e o *Model*. Este componente recebe todos os pedidos efectuados pela *View* para aceder às informações necessárias, e divide-se em três módulos: Gestão de Utilizadores e Monitorização de Facturas e Anuidades e Gestão de Paypal, o primeiro diz respeito a tudo o que envolve o utilizador, desde o seu registo perante a organização, visualização de eventos, registo nos mesmos e à troca de mensagens entre utilizadores; o segundo encontra-se constantemente a verificar se as facturas actuais já se encontram pagas dentro dos limites temporais, e caso as anuidades dos associados não expiraram; o último efectua um pedido de pagamento ao *Paypal*, passando as informações do pagamento para a aplicação web do *paypal*, onde o utilizador deverá poder efectuar o pagamento. No final deste pagamento o utilizador será retornado para a aplicação web de gestão de associados, não lidando esta com os dados de pagamento do associado directamente.

A *View* representa tudo o que é apresentado ao utilizador da aplicação web, dizendo respeito ao interface gráfico, que é o único módulo desta camada.

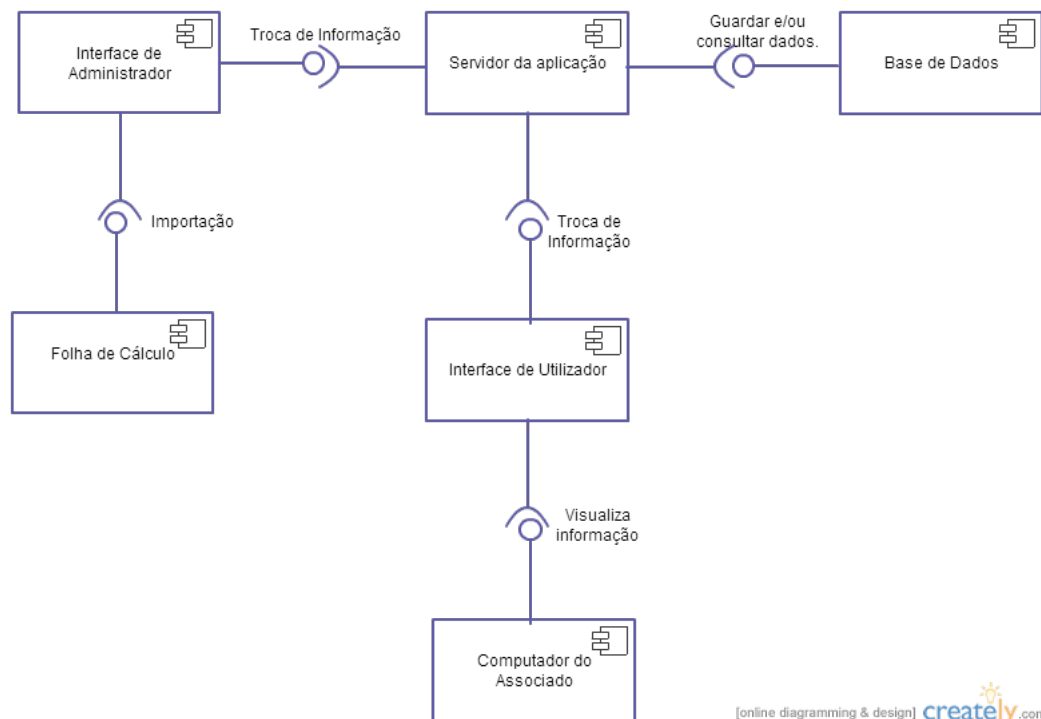


**Figura 3.8:** Modelo de arquitectura do projecto

Na imagem 3.8 é possível notar a relação entre os componentes de Model, View e Controller com os diferentes módulos do projecto de dissertação de mestrado.

### 3.3.3 Diagrama de componentes

Um diagrama de componentes tem como principal objectivo representar as relações estruturais entre os componentes de um sistema[27]. Sendo que se torna um modelo bastante importante no desenvolvimento de um sistema, ajudando à compreensão do mesmo.



**Figura 3.9:** Diagrama de componentes

**Computador do associado:** Responsável por permitir ao utilizador aceder à interface da aplicação fornecida pelo servidor.

**Interface do utilizador:** Responsável por fazer a ponte entre o associado e o servidor da aplicação, permitindo mostrar toda a informação ao utilizador de maneira a que este possa aceder às diferentes funcionalidades.

**Servidor da aplicação:** Gere as ligações entre o utilizador e a interface, bem como todas as funcionalidades que a aplicação irá disponibilizar. Também interage com a base de dados.

**Base de dados:** Responsável por armazenar todos os dados relativos aos associados registados.

**Interface de administrador:** Responsável por permitir ao administrador gerir a associação, dando-lhe acesso a funcionalidades diferentes das de um associado.

**Folha de cálculo:** Ficheiro que contém as informações actuais de todos os associados da AISTI, deverá ser importada para o novo sistema.

## 3.4 Escolhas Tecnológicas

### 3.4.1 Desenvolvimento

Foram comparadas ferramentas com as quais o autor já estava familiarizado e também outras que seriam novas para o mesmo, de forma a poder adoptar aquela que seria a melhor opção para o desenvolvimento da aplicação web.

A escolha acabou por recair na framework Ruby on Rails. Esta framework foi criada em 2003 e desde então tem crescido graças à colaboração de mais de 3700 colaboradores, existindo ainda dezenas de milhar de aplicações web desenvolvidas nesta framework como o GitHub<sup>1</sup>, o Shopify<sup>2</sup> ou ainda o Slideshare<sup>3</sup>.

Uma das razões que deu peso a essa decisão foi o facto de o autor ter já adquirido alguma experiência com a linguagem de programação Ruby. A esta aliam-se outras vantagens que permitem fundamentar esta escolha.

O Ruby on Rails tem vários pontos positivos que levam a que possa superar alguns dos seus concorrentes, estas vantagens são: Rápido desenvolvimento graças à natureza orientada a objectos que o Rails herda do Ruby [28]; os Custos são reduzidos devido ao facto de o Rails e a maioria das suas bibliotecas serem open source [28]; o apoio da Comunidade é enorme, com imensos colaboradores que se certificam de que tudo se mantém em condições e topo para desenvolvimento. [29]; o conceito de The Ruby Way pode ser considerado também uma vantagem pela forma como a arquitectura do sistema fica construída, tornando-se mais fácil a colaboração entre vários developers [30]. No entanto existem também alguns pontos negativos que não se sobrepõem aos pontos positivos, como exemplo disso temos as preocupações ao nível da velocidade e escalabilidade das aplicações desenvolvidas em RoR<sup>4</sup> comparativamente com aquelas que são desenvolvidas em Java ou C. Há, no entanto, várias aplicações com bases de utilizadores, na casa dos milhões, que estão construídas numa base de Ruby on Rails, como por exemplo o Airbnb.

De seguida serão apresentadas algumas comparações entre Ruby on Rails e outras frameworks de desenvolvimento web.

O Ruby on Rails apresenta funcionalidades que lhe são características em relação ao Django, como por exemplo: os testes que são facilitados pelas ferramentas de testes existentes para rails; as ligações à base de dados em rails permitem efectuar todas as operações com alguma facilidade acrescida, graças às migrations; a facilidade de extensão da aplicação é também louvável, e é possível graças ao elevado número de plugins que a comunidade desenvolve e coloca à disposição dos outros. [31] Das características singulares do django a que merece maior destaque é a camada de templates disponibilizada, que permite criar templates customizados de forma facilitada. [31] Pode ainda ser efectuada uma comparação das funcionalidades de cada framework: Ambas as frameworks utilizam templates HTML, no entanto os templates do Rails podem conter código Ruby permitindo funções mais complexas, enquanto que o Django utiliza um conjunto de tags específicas para que quem esteja a criar um novo template não necessite de elevada experiência de programação. Quanto à camada de apresentação o Rails include um protótipo de uma framework de Javascript, já o Django apenas suporta JSON. Ambas as frameworks efectuem a ligação à base de dados através de objectos relacionais, no entanto diferem na forma como estes modelos são definidos: através de Django é o utilizador que especifica cada atributo das classes, no entanto em Ruby on Rails o utilizador não necessita de especificar estes atributos no modelo da classe, pois eles são identificados a partir da base

<sup>1</sup>Para mais informações consultar <https://github.com/>

<sup>2</sup>Para mais informações consultar <https://www.spotify.com/pt/>

<sup>3</sup>Para mais informações consultar <http://www.slideshare.net/>

<sup>4</sup>Ruby on Rails

de dados. [31]

Após terem sido já apresentadas algumas das vantagens que o Rails oferece, serão agora apresentadas algumas vantagens e desvantagens que a framework Play! oferece. Esta framework é relativamente recente, criada em 2007 em scala e java, e evoluiu de forma impressionante, aliado ao facto de que o Heroku<sup>5</sup> suportar aplicações desenvolvidas em Play!Framework. [32] No entanto há alguns factores que demonstram fragilidades desta framework, por exemplo a ausência de plugins, que irá fazer com que uma pessoa perca mais tempo a desenvolver módulos que, noutra framework, podem já existir perto do estado necessário. [32] A Play!Framework poderá ainda causar alguns problemas na altura do deployment por incompatibilidade. [33]

Seguidamente irão ser comparadas Rails e PHP, que é uma linguagem orientada para a web e foi uma das primeiras linguagens a poder ser integrada directamente com HTML. Todo o código é interpretado do lado do servidor gerando a página a ser visualizada pelo cliente. O PHP apresenta algumas vantagens, permitindo criar aplicações web simples, com alguma rapidez mas ,no entanto, torna-se mais complexo caso seja necessária uma aplicação web complexa e recheada de funcionalidades. Tanto PHP como Rails permitem ser incluídos em páginas HTML de forma simples permitindo uma boa integração com Web services. [34] Existem ainda alguns factores que o Rails apresenta em comparação com PHP. Destes fazem parte os namespaces, que em Rails permitem que seja possível ter diversas funcionalidades sem que os seus códigos colidam, ou que interfiram entre as diversas zonas da aplicação web.

Os métodos de organização de código diferem também. Quem programa em PHP está habituado a criar ficheiros de código e a colocá-los num servidor, no entanto em Rails não é assim tão simples, visto que um projecto de Rails inclui diversas peças que funcionam conjuntamente para que tudo corra da melhor forma, ligando a base de dados, repositórios e diversas outras funcionalidades de uma máquina bem oleada. [35]

Concluindo, das diversas frameworks analisadas há algumas que se destacam mais que as outras no desenvolvimento de aplicações web de forma facilitada e completa. Apesar de todos os pontos positivos que tanto a Play!Framework como PHP apresentaram são duas frameworks que, aos olhos do autor, não se enquadram nas opções viáveis para desenvolvimento. Sobram então Ruby on Rails e Django, duas frameworks que não diferem muito no que à performance diz respeito. As diferenças que poderão fazer uma ser opção sobre a outra são poucas e a decisão acaba por recair na preferência do programador. Apesar de estar familiarizado tanto com Python como com Ruby o autor identificou-se mais com a linguagem Ruby, preferindo-a e à sua comunidade, onde 99% do que é necessário está documentado[33]. Entre estas duas frameworks a decisão é Ruby on Rails.

### 3.4.2 Apoio ao desenvolvimento Web

Para suportar o desenvolvimento back-end foi necessário seleccionar ferramentas de apoio ao front-end. Neste caso estão a ser utilizadas Sass<sup>6</sup> e Bootstrap<sup>7</sup>. O autor optou por Sass para o desenvolvimento de stylesheets, recorrendo à sua sintaxe Scss que é bastante similar à do CSS normal, em forma de blocos. O Sass é um pré-processador de CSS, significando que as stylesheets criadas serão depois compiladas para gerar as Stylesheets CSS finais, que serão interpretadas pelo Browser, oferecendo ao programador funcionalidades que o CSS não tem, como variáveis ou blocos de estilos que podem ser usados várias vezes. [36]

Para apoiar o desenvolvimento foi também escolhida a framework Bootstrap, que assenta sobre o Sass, de forma a que a aplicação desenvolvida se torne apto para dispositivos

<sup>5</sup>Para mais informações consultar <https://www.heroku.com/>

<sup>6</sup>Para mais informações consultar <http://sass-lang.com/>

<sup>7</sup>Para mais informações consultar <http://getbootstrap.com/>



móveis, e é ainda compatível com as versões mais recentes dos cinco browsers mais utilizados. O bootstrap tem algumas vantagens que permitem um melhor desenvolvimento do front-end da aplicação: a responsividade através do sistema de grelha com linhas e colunas que se adaptam automaticamente, a customização que pode atingir, permitindo que os programadores seleccionem apenas as funcionalidades de que necessitam [37] e ainda a documentação muito completa que inclui todas as suas funcionalidades bem explicadas. [38]

### 3.4.3 Base de dados

No que à base de dados diz respeito, a escolha recaiu sobre MySQL, que é um linguagem de *queries* muito utilizada no desenvolvimento de aplicações Web. Esta foi seleccionada por se tratar de uma restrição técnica apresentada na altura do levantamento de requisitos.



# Capítulo 4

## Planeamento

Neste capítulo é apresentado todo o planeamento que levou ao desenvolvimento deste projecto de dissertação de mestrado. Em primeiro lugar são apresentadas as tarefas realizadas no primeiro semestre e as tarefas realizadas durante o segundo semestre. Esta última secção encontra-se ainda dividida em datas planeadas e datas reais, para que possa ser possível ver a diferença entre as duas.

### 4.1 Tarefas Realizadas

Todas as tarefas necessárias para o desenvolvimento do trabalho relativo a esta dissertação de mestrado foram divididas por dois semestres, ficando distribuídas da seguinte forma:

#### 1º Semestre

1. *Estudo de soluções existentes no mercado*
2. *Levantamento e definição de requisitos*
3. *Arquitectura do sistema*
4. *Desenvolvimento e teste dos módulos de software*
5. *Relatório e apresentação intermédia*

#### 2º Semestre

6. *Desenvolvimento e integração dos restantes módulos de software*
7. *Deployment da aplicação web*
8. *Testes da aplicação*
9. *Escrita do relatório final*

### 4.1.1 Primeiro Semestre

No diagrama de GANTT apresentado na figura 4.1 é possível perceber quais as tarefas delineadas para conclusão durante o primeiro semestre do ano lectivo 2014-2015. Cada uma das tarefas é subdividida em outras, como representado nas tabelas que irão ser apresentadas de seguida.

Cada uma das tarefas será decomposta em várias subtarefas e descrita textualmente. Para cada uma serão também apresentadas datas iniciais e finais. Na tabela 4.1 encontram-se as tarefas relativas ao estudo de soluções existentes no mercado e na tabela 4.2 as que dizem respeito ao levantamento e definição de requisitos. Todas as subtarefas relativas à arquitectura do sistema podem ser encontradas na tabela 4.3, e as que dizem respeito ao desenvolvimento e teste dos módulos de software encontram-se na tabela 4.4. Por fim, na tabela 4.5 encontram-se as tarefas que dizem respeito ao relatório e apresentação intermédia.

**Tabela 4.1:** Tarefas de estudo de soluções existentes no mercado.

<b>1. Estudo de soluções existentes no mercado</b>	<b>Descrição</b>	<b>Data inicial</b>	<b>Data final</b>
1.1. WildApricot.	Estudo da aplicação web WildApricot, disponível na secção 2.3.1.	05-09-2014	12-09-2014
1.2. AssociaPro.	Estudo da ferramenta AssociaPro, disponível na secção 2.3.2.	13-09-2014	17-09-2014
1.3. ClubMaster.	Estudo da solução ClubMaster, disponível na secção 2.3.3.	18-09-2014	22-09-2014
1.4. Silkstart.	Estudo da ferramenta SilkStart, disponível na secção 2.3.4.	23-09-2014	28-09-2014
1.5. BigTent.	Estudo da aplicação web BigTent, disponível na secção 2.3.5.	29-09-2014	04-10-2014
1.6. YourMembership.com.	Estudo da solução online YourMembership.com, disponível na secção 2.3.6.	04-10-2014	18-10-2014
1.7. SoftManagement.	Estudo da aplicação desktop SoftManagement, disponível na secção 2.3.7.	19-09-2014	24-10-2014
1.8. Conclusão.	Conclusão efectuada após a análise de todas as soluções apresentadas. É possível consultar esta conclusão na secção 2.3.8.	24-10-2014	29-10-2014

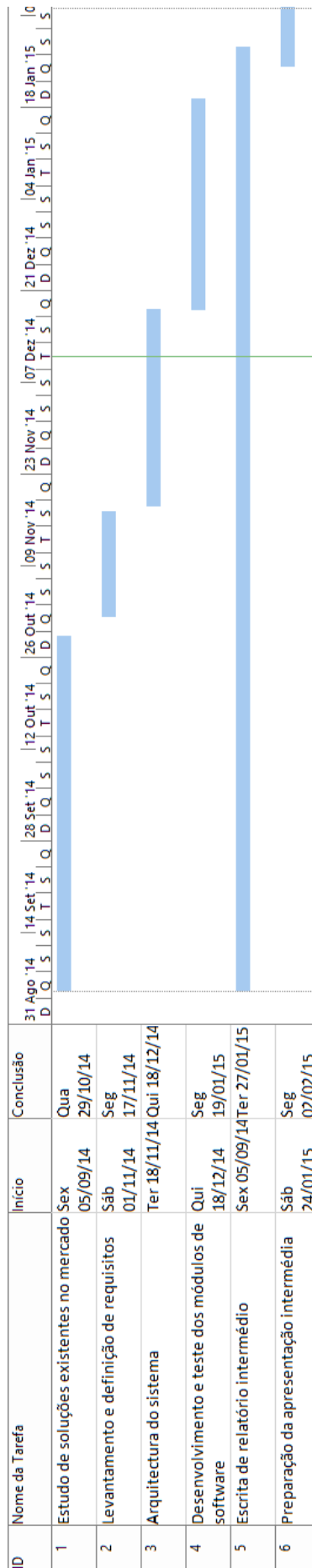


Figura 4.1: Diagrama de GANTT do 1º Semestre

**Tabela 4.2:** Tarefas de levantamento e definição de requisitos.

<b>2. Levantamento e definição de requisitos.</b>	<b>Descrição</b>	<b>Data inicial</b>	<b>Data final</b>
2.1. Levantamento de requisitos.	Levantamento de requisitos e das restrições, técnicas e de negócio, com o Professor Álvaro Rocha.	01-11-2014	03-11-2014
2.2. Construção das tabelas de requisitos.	Construção das tabelas de requisitos, disponível na secção 3.1.1.	04-11-2014	08-11-2014
2.3. Definição de casos de uso.	O casos de uso foram definidos segundo a notação de Allistair Cockburn, e podem ser consultados na secção 3.2.1.	08-11-2014	15-11-2014
2.4. Construção de elaboração do diagrama UML.	Os diagramas UML foram desenvolvidos a partir dos casos de uso criados, e podem ser consultados na secção 3.2.2.	15-11-2014	17-11-2014

**Tabela 4.3:** Tarefas de arquitectura do sistema .

<b>3. Arquitectura do sistema</b>	<b>Descrição</b>	<b>Data inicial</b>	<b>Data final</b>
3.1. Definição dos componentes do sistema.	Definição dos componentes do sistema, disponível na secção 3.3.3.	24-11-2014	04-01-2015
3.2. Definição de diagrama ER.	Desenvolvimento do desenho da base de dados. O diagrama ER pode ser consultado na secção 3.3.1.	05-12-2014	18-12-2014

**Tabela 4.4:** Tarefas de desenvolvimento e teste dos módulos de software.

<b>4. Desenvolvimento e teste dos módulos de software</b>	<b>Descrição</b>	<b>Data inicial</b>	<b>Data final</b>
4.1. Desenvolvimento da página inicial.	Foi desenvolvida a página inicial que irá ser mostrada a todos os utilizadores.	18-12-2014	22-12-2014
4.2. Desenvolvimento dos módulos de registo e login.	Foi construído o módulo que permite aos associados efectuarem o seu registo e login na aplicação web.	22-12-2014	29-12-2014
4.3. Desenvolvimento do módulo criação de eventos.	Foi criada a funcionalidade que permite aos administradores adicionarem novos eventos.	02-01-2015	19-01-2015

**Tabela 4.5:** Tarefas de relatório e apresentação intermédia.

<b>5. Relatório e apresentação intermédia</b>	<b>Descrição</b>	<b>Data inicial</b>	<b>Data final</b>
5.1. Estudo do estado da arte.	O estudo do Estado da Arte engloba também a análise de concorrência efectuada, e pode ser consultado no capítulo 2.	05-09-2014	28-12-2014
5.2. Planeamento.	O planeamento do projecto foi efectuado tendo em conta o período disponível e as tarefas atribuídas, e pode ser consultado no capítulo 4.	14-09-2014	15-10-2014
5.3. Descrição do sistema.	A descrição do sistema engloba os requisitos e a arquitectura, bem como mockups das principais funcionalidades. Pode ser consultada no capítulo 3.	10-12-2014	07-01-2015
5.4. Apresentação intermédia.	Construção e treino da apresentação intermédia da tese de mestrado.	24-01-2015	02-02-2015

### 4.1.2 Segundo Semestre

Na tabela 4.6 é possível consultar as tarefas relacionadas com o desenvolvimento dos restantes módulos de software, na tabela 4.7 as tarefas que estão ligadas ao deployment da aplicação web, na tabela 4.8 as que se relacionam com os testes da aplicação e por fim as tarefas acerca da escrita do relatório final na tabela 4.9. Todas estas tabelas são referentes às datas planeadas, que podem também ser vistas na figura 4.2, que demonstra o diagrama de GANTT para estas tarefas. As datas planeadas no entanto não foram sempre cumpridas, e nas tabelas 4.10, 4.11, 4.12 e 4.13 é possível ver as mestas tarefas mas com as datas que realmente foram cumpridas, e ainda um indicador da diferença entre as datas planeadas e as datas cumpridas.

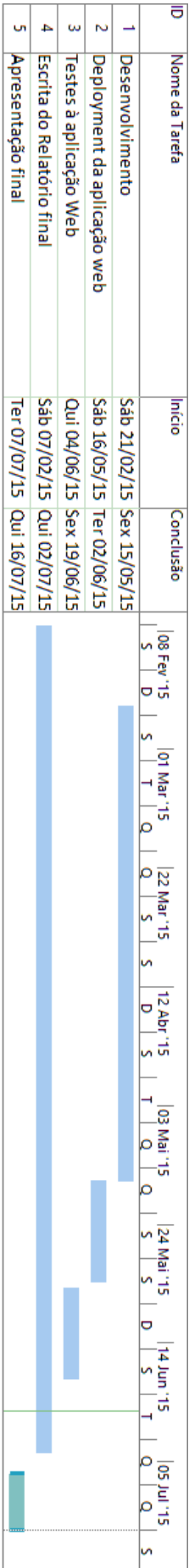


Figura 4.2: Diagrama de GANTT do 2º Semestre



## Datas planeadas

**Tabela 4.6:** Tarefas de desenvolvimento de integração dos módulos de software - Datas planeadas.

1. Desenvolvimento	Descrição	Data inicial	Data final
1.1. Desenho de um novo template.	Desenvolvimento, de raiz, de um template para o website.	21-02-2015	25-02-2015
1.2. Desenvolvimento do módulo de gestão de utilizadores.	Desenvolvimento de todo o módulo que gere os utilizadores, disponível na secção ??.	24-02-2015	4-3-2015
1.3. Testes do módulo de gestão de utilizadores.	Testes unitários e funcionais do módulo de gestão de utilizadores.	24-02-2015	5-4-2015
1.4. Desenvolvimento do módulo de monitorização de facturas e anuidades.	Desenvolvimento de todo o módulo que monitoriza de facturas e anuidades, disponível na secção ??	6-4-2015	20-4-2015
1.5. Testes do módulo de monitorização de facturas e anuidades.	Testes unitários e funcionais do módulo de monitorização de facturas e anuidades. 6-4-2015		20-4-2015
1.6. Desenvolvimento do módulo de gestão de paypal.	Desenvolvimento de todo o módulo que gere os pagamentos por paypal, disponível na secção ??	21-4-2015	3-5-2015
1.7. Testes de gestão de paypal.	Testes unitários e funcionais do módulo de gestão de pagamentos por paypal.	21-4-2015	5-5-2015
1.8. Integração dos módulos de software.	Processo de integração de todos os módulos para a aplicação web final.	5-5-2015	15-5-2015

**Tabela 4.7:** Tarefas de deployment da aplicação web - Datas planeadas

<b>2. Deployment da aplicação web</b>	<b>Descrição</b>	<b>Data inicial</b>	<b>Data final</b>
2.1. Configuração de servidor temporário.	Processo de configuração das diversas ferramentas necessárias à execução da aplicação web.	16-5-2015	20-5-2015
2.2. Configuração da ferramenta capistrano.	Configuração da ferramenta que facilita o deployment.	21-5-2015	23-5-2015
2.3. Deployment no servidor temporário.	Deploy de toda a aplicação web no servidor Digital Ocean registado pelo autor.	24-5-2015	29-5-2015
2.4. Configuração da ferramenta New Relic.	Configuração da ferramenta que permite consultar dados acerca do servidor e da aplicação.	30-5-2015	2-6-2015
2.5. Configuração da ferramenta Sentry.	Configuração da ferramenta que monitoriza erros no servidor e notifica o autor.	30-5-2015	2-6-2015

**Tabela 4.8:** Tarefas de testes à aplicação web - Datas planeadas.

<b>3. Testes à aplicação web</b>	<b>Descrição</b>	<b>Data inicial</b>	<b>Data final</b>
3.1. Testes após implementação dos módulos.	Testes funcionais levados a cabo após a implementação de todos os módulos.	4-6-2015	8-6-2015
3.2. Testes de performance.	Processo de teste de performance da aplicação web e servidor.	8-6-2015	14-6-2015
3.3. Testes de Segurança.	Processo de testar a segurança dos acessos à aplicação web.	14-6-2015	17-6-2015
3.4. Estudo para testes de usabilidade.	Processo de estudo e preparação dos testes de usabilidade.	17-6-2015	19-6-2015

**Tabela 4.9:** Escrita do relatório final da dissertação de mestrado - Datas planeadas.

<b>3. Escrita final</b>	<b>Descrição</b>	<b>Data inicial</b>	<b>Data final</b>
4.1. Correções ao relatório intermédio	Correcção de vários capítulos da dissertação de mestrado, seguindo recomendações ouvidas na defesa intermédia.	7-2-2015	18-02-2015
4.2. Escrita do relatório final	Escrita dos restantes capítulos da dissertação de mestrado.	25-2-2015	2-7-2015
4.3. Criação e preparação da apresentação final da dissertação de mestrado.	Criação, preparação e treino da apresentação final da dissertação de mestrado.	7-7-2015	16-7-2015

## Datas reais

**Tabela 4.10:** Tarefas de desenvolvimento de integração dos módulos de software - Datas reais.

<b>1. Desenvolvimento</b>	<b>Data inicial</b>	<b>Data final</b>	<b>Diferença</b>
1.1. Desenho de um novo template.	21-02-2015	26-02-2015	+1 dia
1.2. Desenvolvimento do módulo de gestão de utilizadores.	25-02-2015	10-3-2015	+6 dias
1.3. Testes do módulo de gestão de utilizadores.	25-02-2015	15-4-2015	+11dias
1.4. Desenvolvimento do módulo de monitorização de facturas e anuidades.	15-4-2015	2-5-2015	+12dias
1.5. Testes do módulo de monitorização de facturas e anuidades.	15-4-2015	7-5-2015	+17dias
1.6. Desenvolvimento do módulo de gestão de paypal.	4-5-2015	16-5-2015	+11dias
1.7. Testes de gestão de paypal.	4-5-2015	18-5-2015	+13dias
1.8. Integração dos módulos de software.	18-5-2015	26-5-2015	+11dias

**Tabela 4.11:** Tarefas de deployment da aplicação web - Datas reais.

<b>2. Deployment da aplicação web</b>	<b>Data inicial</b>	<b>Data final</b>	<b>Diferença</b>
2.1. Configuração de servidor temporário.	27-5-2015	31-05-2015	+11dias
2.2. Configuração da ferramenta capistrano.	31-5-2015	1-6-2015	+9dias
2.3. Deployment no servidor temporário.	1-6-2015	2-6-2015	+7dias
2.4. Configuração da ferramenta New Relic.	3-6-2015	4-6-2015	+2dias
2.5. Configuração da ferramenta Sentry.	3-6-2015	4-6-2015	+2dias

**Tabela 4.12:** Tarefas de testes à aplicação web - Datas reais.

<b>3. Testes à aplicação web</b>	<b>Data inicial</b>	<b>Data final</b>	<b>Diferença</b>
3.1. Testes após implementação dos módulos.	4-6-2015	8-6-2015	+0dias
3.2. Testes de performance.	8-6-2015	14-6-2015	+0dias
3.3. Testes de Segurança.	14-6-2015	18-6-2015	+1dia
3.4. Estudo para testes de usabilidade.	18-6-2015	20-6-2015	+1dia

**Tabela 4.13:** Escrita do relatório final da dissertação de mestrado - Datas reais.

<b>3. Escrita final</b>	<b>Descrição</b>	<b>Data inicial</b>	<b>Data final</b>
4.1. Correções ao relatório intermédio.	7-2-2015	20-2-2015	+2dias
4.2. Escrita do relatório final	25-2-2015	2-7-2015	+0dias
4.3. Criação e preparação da apresentação final da dissertação de mestrado.	7-7-2015	16-7-2015	+0dias

## 4.2 Ferramentas utilizadas

Neste capítulo irão ser abordadas as ferramentas utilizadas ao longo da dissertação de mestrado do autor. Estas ferramentas foram utilizadas no apoio ao projecto, durante o planeamento e também durante o desenvolvimento.

### 4.2.1 Git

Esta ferramenta foi utilizada para controlo de versões ao longo do desenvolvimento do projecto. Este é um sistema que funciona com base em repositórios de código, para este projecto de dissertação de mestrado foi utilizado o host Bitbucket<sup>1</sup>. Para toda a dissertação de mestrado foram utilizados dois repositórios, um para os ficheiros LaTeX que dizem respeito ao presente documento, e um outro para a aplicação web, com diversos commits para que as versões estejam armazenadas e categorizadas num local seguro. O segundo pode ser visto na figura 4.3.

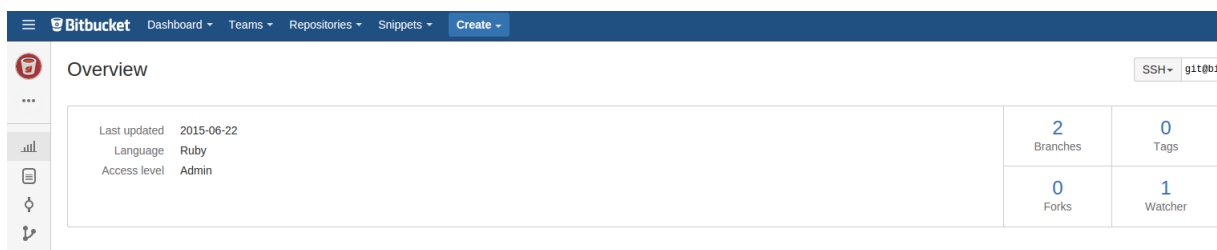


Figura 4.3: Repositório Git

### 4.2.2 Trello

O Trello é uma ferramenta que facilita a organização de projectos através de quadros de tarefas, que pode ser gerido por grupos, ou individualmente. Cada quadro de tarefas pode conter listas, e cada lista pode conter um ou mais cartões, cada cartão representa uma tarefa a ser efectuada. Cada cartão pode ainda conter uma checklist, por exemplo o cartão “Escrever secção das Ferramentas Utilizadas” contem uma lista com os seguintes pontos: Git; Trello; Google Docs; Visual Paradigm; Ngrok e Digital Ocean, ao escrever cada um destes pontos o autor coloca um check para o item correspondente. Para a dissertação de mestrado foram criadas várias listas de cartões por completar, e uma final chamada “Done” para que possa ser mantido um registo de tudo o que foi efectuada até à data. Na imagem 4.4 é possível ver um exemplo de quadro de tarefas.

### 4.2.3 Google Docs

O Google Docs foi a ferramenta utilizada para a escrita das primeiras versões dos capítulos da dissertação de mestrado, permitindo uma mais fácil revisão dos mesmos, para que possam depois ser incorporados no documento já na sua forma final. Foi criada uma pasta geral, subdividida em capítulos da dissertação de mestrado. Esta pasta pode ser vista na figura 4.5.

<sup>1</sup>Mais informações em: [www.bitbucket.org/](http://www.bitbucket.org/)

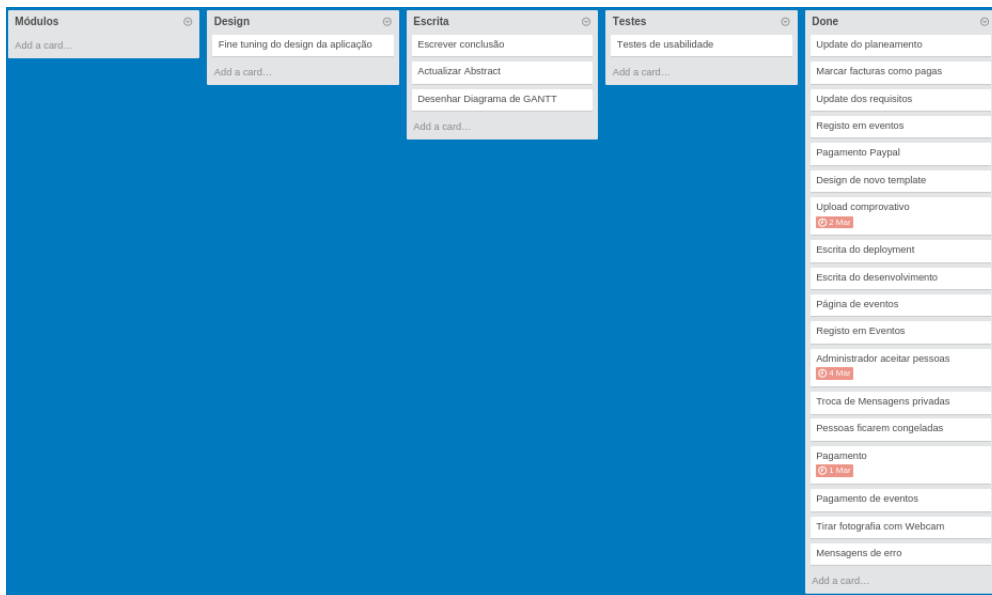


Figura 4.4: Quadro de tarefas do Trello

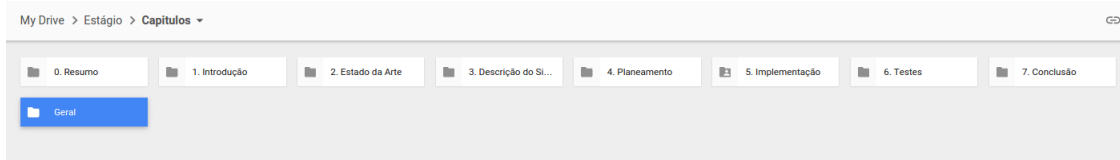


Figura 4.5: Pasta geral da dissertação de mestrado no Google Docs

## 4.2.4 Visual Paradigm

Foi utilizada uma versão do Visual Paradigm para desenhar os diferentes diagramas presentes nesta dissertação de mestrado, como o modelo de entidades e relacionamentos, o de Componentes ou o diagrama que representa a arquitectura Model View Controller. O *workspace* do *Visual Paradigm* encontra-se representado na figura 4.6.

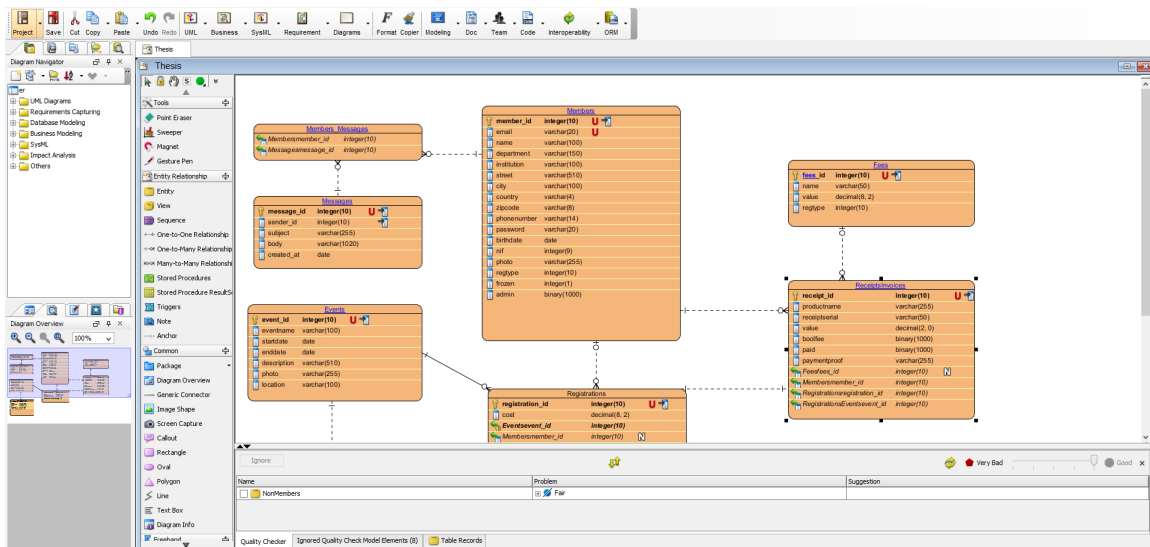


Figura 4.6: Área de trabalho do Visual Paradigm

### 4.2.5 Ngrok

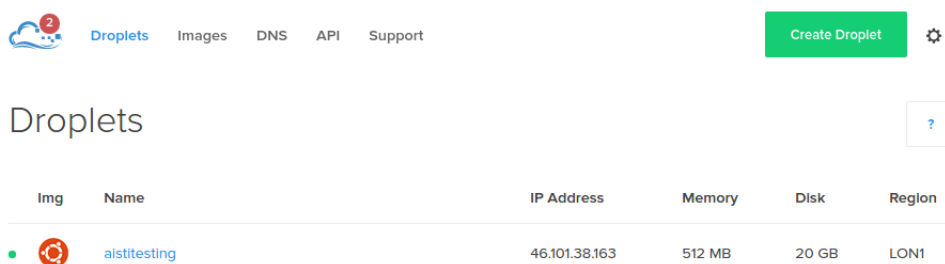
Ferramenta utilizada para a criação de túneis de ligação remota ao endereço local da máquina do autor onde estava a ser desenvolvida a aplicação web, correndo a partir de um terminal como pode ser constatado através da figura 4.7. Esta ferramenta permitiu assim testar remotamente algumas funcionalidades e ainda testar os pagamentos através do Paypal.

```
ngrok (Ctrl+C to quit)
Tunnel Status      online
Version            1.6/1.7
Forwarding         http://5a3ae50a.ngrok.com -> 127.0.0.1:3000
Forwarding         https://5a3ae50a.ngrok.com -> 127.0.0.1:3000
Web Interface      My Drive 127.0.0.1:4040
# Conn            0
Avg Conn Time     0.00ms
```

Figura 4.7: Processo de tunneling do Ngrok

### 4.2.6 Digital Ocean

Esta aplicação online oferece servidores na cloud, e foi utilizada para efectuar um early deploy da aplicação web, para testes por parte do Autor e do seu orientador, o Prof. Álvaro Rocha. Através do Digital Ocean é possível criar um servidor num local seleccionado pelo cliente e aceder-lhe através de sftp e ssh, para que seja possível efectuar diversos tipos de configurações. A este servidor é chamado *Droplet*, e a página de *overview* pode ser vista na figura 4.8.




Img	Name	IP Address	Memory	Disk	Region
	aisttesting	46.101.38.163	512 MB	20 GB	LON1

Figura 4.8: Droplet do Digital Ocean





# Capítulo 5

## Desenvolvimento

Neste capítulo serão descritas as diferentes metodologias de desenvolvimento para cada um dos módulos desenvolvidos: o sistema de gestão de associados, de monitorização de facturas e anuidades e o de gestão de pagamentos online. Estes módulos foram já falados previamente no capítulo Arquitectura de Sistema 3.3.

Em primeiro lugar será explicado o processo de inicio de desenvolvimento e de fluxo do sistema, mais tarde será demonstrado de que forma se desenvolveu, e qual o seu propósito e ainda o módulo de monitorização de facturas e anuidades. Por fim será descrita a integração do módulo de pagamentos por Paypal com o paypal em si.

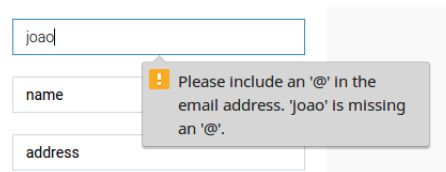
### 5.1 Módulo de gestão de utilizadores

Este módulo engloba todas as funcionalidades necessárias para que os associados possam interagir com a aplicação. As diferentes funcionalidades são: registo de um novo associado, e respectiva edição de perfil, troca de mensagens entre associados, visualização de eventos e registo nos mesmos. Cada uma destas funcionalidades foi desenvolvida em separado e foram depois integradas no projecto de forma a que a aplicação web funcione como um todo.

Inicialmente foram desenvolvidas as funcionalidades relacionadas com as sessões de utilizador: o registo, o login e a edição do perfil do utilizador. Há que ter atenção a diversos cuidados que devem ser tomados na hora de manusear os dados mais sensíveis de utilizadores, como por exemplo a sua password.

Para que este manuseamento fosse feito de forma segura a password é encriptada através de bcrypt [39]. Este algoritmo cria uma hash a partir da *password*, do *salt* e do *cost*. Esta password é um hash que demora aproximadamente 100ms a ser calculado, tornando a tarefa de a desencriptar mais complicada para um alguém com intenções malignas.

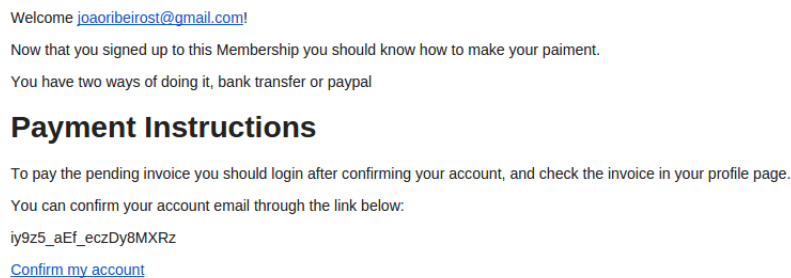
Ao efectuar o registo, o e-mail e a password são validados no momento, para que sejam cumpridos certos requisitos. Na imagem 5.1 é possível visualizar o erro gerado quando não é introduzido um e-mail válido.



**Figura 5.1:** Erro de validação de e-mail

No caso de um utilizador se esquecer de qual a sua password poderá ainda efectuar a recuperação da mesma. Esta recuperação processa-se fazendo reset à password actual do utilizador e enviando-lhe um e-mail com instruções para que possa escolher uma nova password. Ao efectuar login o utilizador pode ainda seleccionar a opção “*Remember me*” para que a aplicação se lembre do utilizador com o login efectuado.

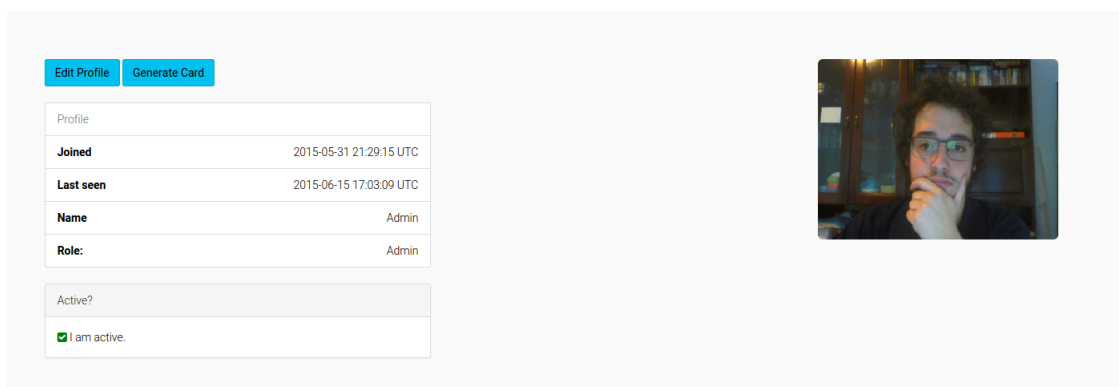
Para que o utilizador possa utilizar a aplicação web como membro registado terá que, antes de efectuar login, confirmar a sua conta a partir de um e-mail que lhe é enviado, que pode ser visto na imagem 5.2. Ao seguir o *url* que é apresentado no e-mail a aplicação irá reconhecer o token gerado para aquele utilizador e confirmar o seu registo na base de dados.



**Figura 5.2:** E-mail de confirmação de registo

Sempre que é efectuado um login serão armazenadas algumas informações na base de dados que permitem acompanhar a actividade do utilizador como: o número de logins efectuadas até ao momento; a data do ultimo login e ainda o endereço IP com que este se registou.

O utilizador poderá editar o seu perfil, tendo a possibilidade de adicionar uma foto sua, como é possível verificar na imagem 5.3. Esta poderá ser adicionada tanto a partir do seu computador como tirando na altura a partir da webcam. As imagens em .jpg e .png são optimizadas e comprimidas de forma a que não interfiram com a performance da aplicação web, sendo armazenado na base de dados o path onde esta se encontra. O utilizador pode ainda editar os restantes campos, para tal apenas necessita de introduzir a sua password actual, para que possa ser validado.



**Figura 5.3:** Fotografia de perfil do utilizador

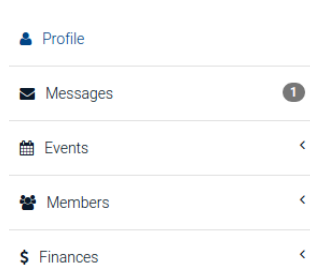
Para além da edição do perfil os associados podem ainda gerar o seu cartão de sócio online, sendo-lhes apresentado num documento pdf gerado no momento pela aplicação web. O cartão tem uma disposição de elementos idêntica à que se pode ver na figura 5.4



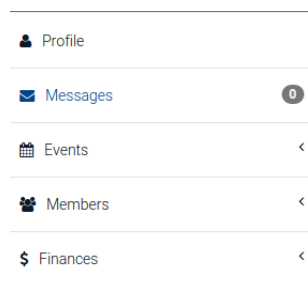
**Figura 5.4:** Cartão de associado

Relativamente à troca de mensagens os utilizadores poderão criar conversas com outros utilizadores. Em cada conversa poderão trocar mensagens, podendo depois gerir cada conversa que têm na sua inbox, eliminando-as ou marcando-as como lidas ou não. Os utilizadores podem ainda consultar a sua caixa de mensagens enviadas e o lixo de onde poderão recuperar uma conversa eliminada por engano.

Na eventualidade de um utilizador receber mensagens, ser-lhe-á mostrada a contagem de em quantas conversas este têm mensagens por ler, como na figura 5.5. Caso o utilizador não tenha nenhuma mensagens para ler ser-lhe-á apresentado o contador de mensagens com o número 0, como na figura 5.6.



**Figura 5.5:** Indicador de novas mensagens com novas mensagens



**Figura 5.6:** Indicador de novas mensagens a zero

As outras funcionalidades importantes deste módulo para os utilizadores são as que se encontram relacionadas com os Eventos. Neste caso a visualização dos mesmo e respectivo registo.

Ao consultar a lista de eventos existentes ao utilizador são apresentadas algumas informações como: a data inicial, a imagem que o evento tem, o título, a descrição e a localidade onde este irá decorrer.

Assim que o utilizador entre na página do evento poderá consultar ainda as opções de registo que pode seleccionar. Estas opções encontram-se divididas em obrigatórias e opcionais.

Após consultar o evento é possível efectuar o registo no mesmo, sendo gerada uma factura que indica ao utilizador os dados que deve utilizar para efectuar o pagamento da mesma, seja por paypal ou por transferência bancária.

O administrador tem também ele várias funções relacionadas com este módulo, elas são: visualização de todos os associados, possibilidade de eliminar associados, possibilidade de alterar o estado dos associados, possibilidade de gerar etiquetas para envio de cartas e a criação das próprias cartas, criação, edição e eliminação de eventos.

Falando das duas primeiras funcionalidades referidas, ao consultar a lista de utilizadores, como na figura 5.7, é apresentada ao administrador a opção de remover um determinado utilizador, se efectivamente for eliminado esta acção não é reversível e o utilizador terá que se registar novamente na aplicação web.

Listing Users				
ID	Name	Email	Status	
2		eddecuhus7176@gmail.com	admin	Show
4		admin@admin.com	admin	Show
8	João	joaoribeirost@gmail.com	pending	Show

Displaying all 3 User

**Figura 5.7:** Lista de membros registados na aplicação web

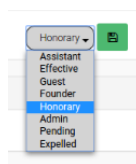
Ao consultar o perfil de um associado é ainda possível consultar: a data do seu ultimo login; o dia do registo; a lista de facturas pagas, ou não, do associado; e é ainda possível ao administrador alterar o estado do utilizador, como na figura 5.8, podendo seleccionar de entre os seguintes:

- *Assistant* - Representativo dos associados auxiliares.
- *Effective* - Para todos os associados que se encontram já como efectivos.

- *Guest* - Engloba todos os associados convidados.
- *Founder* - Utilizado para representar os associados fundadores da AISTI.
- *Honorary* - Para todos aqueles que se encontram ligados à associação de forma honorária.
- *Admin* - Os associados que pertencem a esta categoria são os administradores da aplicação, tendo mais privilégios que o normal.
- *Pending* - Representa todos os associados que não efectuaram ainda o pagamento da anuidade, encontrando-se pendentes.
- *Expelled* - No caso de um associado ser expulso da associação poderá ser seleccionado este estado para o utilizador.

Existem ainda outros dois estados não representados, pois a sua alteração é efectuada de forma automática, estes são:

- *Frozen* - Representa os associados que já se encontraram vinculados à associação mas que não efectuaram a renovação da sua anuidade.
- *Event Goer* - Para os não membros que apenas se registem em eventos, para que possam consultar as suas facturas e efectuar o pagamento das mesmas.



**Figura 5.8:** Menu de alteração de estado

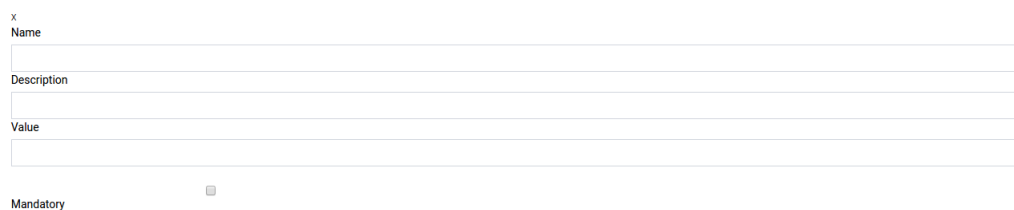
Ainda em relação às funcionalidades possíveis de gestão de associados o administrador poderá gerar um documento de etiquetas com os dados de cada um dos utilizadores, de forma a que seja apenas necessário colar em envelopes. Na zona administrativa pode ainda ser criada uma carta que será exportada em pdf para impressão. Há ainda a opção de enviar por email para todos os associados esta carta. O administrador pode ainda gerir os eventos, começando pela criação de um novo evento. Para que este passo seja possível é necessário preencher várias informações acerca do mesmo: fotografia de capa; nome; datas inicial e final; uma pequena descrição do evento; a localização, que será associada a um local num mapa através da Google Maps API<sup>1</sup>. Para completar a criação de um novo evento devem ser adicionadas as opções de registo, deve existir pelo menos uma opção obrigatória para que os associados se possam registar. Todas as não obrigatórias são consideradas extras, com preços a acumular com as das opções obrigatórias. Para cada opção de registo, extra ou não, devem ser preenchidos os seguintes campos:

- *Nome* - O nome da opção que irá ser adicionada ao evento.
- *Descrição* - Breve texto que descreva qual a natureza do evento.

<sup>1</sup>Mais informações em: <https://developers.google.com/maps/>

- *Preço* - O valor que a opção irá custar ao utilizador que se pretende registar.
- *Obrigatória* - Define o tipo de opção a ser adicionada: se for obrigatória apenas uma pode ser seleccionada pelo associado, por exemplo “todos os dias do evento” ou “apenas sábado”; no caso de ser opcional poderá ser, ou não, seleccionada, por exemplo “com jantar incluído” ou “levarei acompanhante”.

Na figura 5.9 é possível ver os campos de preenchimento do tipo de registo em evento.



O formulário contém os seguintes campos:

- Nome: campo de texto com um cursor 'x' no início.
- Description: campo de texto.
- Value: campo de texto.
- Mandatory: uma caixa de seleção desactivada.

**Figura 5.9:** Formulário de tipo de registo em evento

O administrador pode ainda editar os dados de um evento, ou eliminar o mesmo de forma não reversível.

## 5.2 Módulo de gestão e monitorização de facturas e anuidades

O módulo de gestão e monitorização de facturas e anuidades será apresentado de seguida, bem como as funcionalidades para utilizadores e para administradores.

As funções deste módulo passam por monitorizar diariamente as facturas que se encontram por pagar e a anuidade dos associados. De forma a que tal seja possível é corrida uma thread todos os dias à meia noite que irá verificar vários factores acerca das facturas não pagas e das anuidades não renovadas, dividindo-se em duas categorias:

- *Facturas de Eventos* - A verificação de facturas de eventos é efectuada da seguinte forma:
  - Numa primeira instância, quando já decorreram 75 dias desde que o evento ocorreu. Neste caso será enviado um e-mail de aviso ao associado que se encontra em dívida perante a associação para o relembrar do pagamento que necessita ser efectuado.
  - Numa segunda instância irá verificar se já decorreram 90 dias e a factura não tiver sido paga será então enviado um e-mail ao associado, avisando-o do seu incumprimento, e ao administrador para que este possa tomar medidas para com o associado em dívida.
- *Anuidades* - No controlo das anuidades o sistema irá agir em duas alturas:
  - Primeiro irá verificar se já passaram 340 dias desde o pagamento da anuidade actual, enviando um e-mail ao associado a avisar que deverá

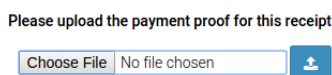
proceder à renovação da sua quota, caso contrário irá ser colocado no estado “Frozen” não podendo aceder às regalias que a associação lhe oferece.

- Caso se passem 366 dias desde a ultima anuidade e esta não tenha sido renovada o sistema irá então congelar o associado, notificando-o do seu incumprimento. O Administrador será também avisado através de e-mail que um dos associados não renovou as suas quotas.

No caso da gestão de anuidades deverá ser adicionado um dia às verificações no caso de se tratar de um ano bissexto.

Para além das funções internas do sistema há ainda algumas que são geridas por este módulo, mas que no entanto são efectuadas pelos utilizadores que poderão: consultar as suas facturas, consultar quanto tempo falta até terminar a sua anuidade, e efectuar o upload de comprovativos de pagamento das suas facturas, de anuidade, ou não.

Tudo isto poderá ser efectuado a partir da área de utilizador sendo que a consulta do tempo restante até terminar a sua anuidade e as suas facturas podem ser consultadas na sua página de perfil. Ao consultar uma factura em particular irá deparar-se com os dados desse documento, incluindo os dados da AISTI e do próprio associado. Caso não tenha ainda pago os valores em dívida poderá fazê-lo de duas formas: imediatamente, através de paypal ou através de transferência bancária. Se o associado optar pela segunda opção terá que, após efectuar o pagamento, fazer um upload do comprovativo do pagamento através do campo da figura 5.10, para que o administrador possa actuar sobre ele mais tarde. Assim que o upload é efectuado são enviados dois e-mails, um ao associado para o notificar de que o upload foi completado com sucesso, e um ao administrador a notificar da existência de um novo comprovativo para validar.



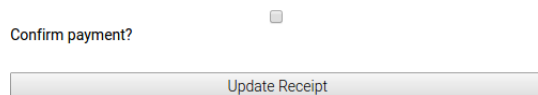
**Figura 5.10:** Campo para upload de comprovativo

O associado pode ainda gerar um pdf da factura que se encontra a consultar, para que possa imprimir ou simplesmente armazenar no seu computador de forma digital.

Chegamos por fim às funcionalidades que este módulo coloca à disponibilidade dos administradores. Os administradores terão a possibilidade de consultar todas as facturas pagas e por pagar, sejam elas relativas a eventos ou a anuidades de associados. Caso seja necessário poderá também editar as quotas (fees) já inseridas no sistema, alterando o seu valor e nome, deverá existir uma quota para cada tipo de registo:

- *Single*
- *Scholar*
- *Company*

Uma das funções mais importantes a que o administrador tem acesso neste módulo é a de validação de pagamentos, ao consultar uma factura por pagar, caso já tenha sido efectuado o upload de um comprovativo o administrador poderá então consultar esse documento e, caso seja válido, confirmar o pagamento da factura através dos campos da figura 5.11.



**Figura 5.11:** Confirmação de pagamento da factura

### 5.3 Módulo de Gestão de Paypal

Neste módulo são geridas as ligações ao site Paypal, que permitem gerir os pagamentos. Cada vez que o utilizador selecciona pagamento por paypal é redireccionado para a página do paypal, representada na figura 5.12, com um pedido de criação de novo pagamento no paypal. O utilizador poderá então proceder ao pagamento através de cartão de crédito ou do dinheiro que tem armazenado na sua conta paypal.

Após efectuar o pagamento o utilizador é automaticamente redireccionado para a aplicação web de gestão de associados da AISTI, onde a factura será marcada como paga e, no caso de ser um pagamento de anuidade, o estado do utilizador será alterado automaticamente de Pendente para Auxiliar.

Para testar este módulo foram utilizadas contas de comprador e vendedor na sandbox do paypal, estas são contas virtuais com dinheiro em caixa também ele virtual e informações falseadas, para que se possam testar os pagamentos.

Resumo do seu pedido	
Descrições	Valor:
Anual fee Single created on 2015-06...	€120,50
Preço do artigo: €120,50	
Quantidade: 1	
<b>Total de artigos</b>	<b>€120,50</b>
	<b>Total €120,50 EUR</b>

**Figura 5.12:** Página de pagamento do Paypal

Um dos problemas encontrados ao testar os pagamentos de paypal com a aplicação web a ser executada localmente era o facto de o paypal não ter um host para o qual retornar, não enviando quaisquer informações acerca do estado do pagamento e do cliente. No entanto foi utilizada uma ferramenta chamada ngrok<sup>2</sup> que permite efectuar uma ligação externa ao servidor local, através de um url temporário.

Para que seja possível aceitar pagamentos através do paypal a AISTI terá que ter uma conta de vendedor certificada e activar nas suas definições a opção que permite o auto-redireccionamento do utilizador após o pagamento da factura.

<sup>2</sup>Mais informações em: <https://ngrok.com/>



## 5.4 Deployment

Nesta secção irá ser inicialmente apresentada a forma como todo o processo de *deployment* foi realizado e ainda as ferramentas utilizadas para um melhor apoio à manutenção do mesmo.

Para que o *deployment* fosse efectuado com sucesso foi criado um droplet através da ferramenta de cloud Digital Ocean. A máquina seleccionada possui 512Mb e 20Gb de espaço num disco SSD e corre o sistema operativo *Ubuntu 14.04*, configurado para Ruby on Rails.

O Digital Ocean permite criar *snapshots* do servidor. Estas são uma espécie de backup do estado do servidor, podendo voltar a um dos *snapshots* posteriormente caso existam problemas nas execuções mais recentes.

Na página do *droplet* é ainda possível visualizar algumas estatísticas, como a largura de banda utilizada, a utilização de disco e de processador. Na figura 5.13 é possível verificar essas estatísticas para os últimos 30 dias.



**Figura 5.13:** Estatísticas do servidor Digital Ocean

Foi ainda configurado no projecto, e no servidor, o Capistrano<sup>3</sup>. Esta ferramenta automatiza o processo de *deploy* para um servidor remoto, facilitando o processo ao programador que pretende colocar a sua aplicação web online. Inicialmente é necessário configurá-lo tendo atenção a vários factores, sendo necessário resolver alguns problemas iniciais da configuração. No entanto após tudo estar configurado da maneira correcta o processo é acelerado.

O Capistrano pode ser configurado de forma a que efectue *deploy* directamente de um repositório Git para o servidor. Quer isto dizer que ao utilizar o Git como sistema de controlo de versões o processo de *deployment* pode ser ainda mais facilitado, tendo apenas de garantir que o Capistrano se encontra configurado para o *branch* final.

Ao executar o comando `cap staging deploy` a ferramenta irá carregar os ficheiros da release mais recente para o servidor e efectuar vários passos automaticamente,

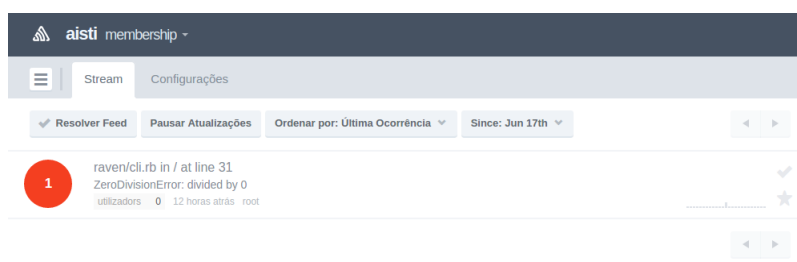
<sup>3</sup>Mais informações <http://capistranorb.com/>

como: pré-compilação do Sass para CSS, as *migrations* da base de dados e ainda todo o processo de *bundle* para gestão de gems.

Para melhor apoio a toda a manutenção da aplicação web foi ainda configurada uma outra ferramenta, o Sentry. Esta ferramenta é um logger em tempo real, especializado em monitorizar erros e extrair informações críticas para as suas resoluções.

É possível configurar esta ferramenta de forma a receber os erros no e-mail, numa conversa de Slack ou IRC, por exemplo, e permite até uma integração com o GitHub. Para o projecto de dissertação de mestrado foram configurados os alertas por e-mail.

Na figura 5.14 é possível visualizar uma mensagem de erro no painel de administração da ferramenta Sentry.



**Figura 5.14:** Erro da aplicação web

Actualmente a aplicação web encontra-se instalada num servidor temporário de Digital Ocean, estando a ser estudadas as opções de hosting para que esta possa ser migrada para o servidor final, onde ficará em utilização pela AISTI.

# Capítulo 6

## Testes

Neste capítulo serão apresentados os testes realizados à aplicação web realizada no âmbito do projecto de dissertação de mestrado. Inicialmente será exposto o plano de testes unitários e de integração para questões relacionadas com o desenvolvimento da aplicação web, como testes à base de dados e a preparação para testes de usabilidade à interface gráfica da aplicação web. Numa segunda instância serão apresentados os testes de sistema, que irão incidir sobre questões de usabilidade e performance.

Nesta secção serão apresentados os testes efectuados à base de dados e às funcionalidades da aplicação Web. No final será ainda descrito o plano para testes de usabilidade no futuro.

### 6.1 Testes à base de dados

Durante a execução dos testes à base de dados será necessário:

- Verificar que a informação é guardada correctamente na base de dados.
- Verificar que as informações editadas são correctamente alteradas na base de dados.
- Verificar que as informações removidas são correctamente eliminadas na base de dados.

#### 6.1.1 Estratégia

Na tabela 6.1 pode ser visualizada a estratégia escolhida para levar a cabo os testes à base de dados.

**Tabela 6.1:** Testes à Base de Dados

Objectivo do Teste:	Garantir que todos os acessos à base de dados funcionam da forma correcta e sem corrupção dos dados.
Técnica:	<ul style="list-style-type: none"> <li>• Testar cada pedido à base de dados.</li> <li>• Consultar a base de dados de modo a verificar se os dados estão a ser bem guardados.</li> </ul>
Garantia de finalização:	Todos os acessos à base de dados funcionam correctamente e sem corromper os dados.

## 6.2 Testes Funcionais

Para a execução dos testes funcionais é necessário:

- **Para visitantes**
  - Verificar que os visitantes se conseguem registar e efectuar o login.
- **Para utilizadores registados**
  - Verificar que os utilizadores registados conseguem consultar e pagar a sua factura.
  - Verificar que os utilizadores registados conseguem efectuar upload do comprovativo de pagamento.
  - Verificar que os utilizadores registados conseguem registar-se em eventos.
- **Administradores:**
  - Verificar que os administradores conseguem confirmar o pagamento de facturas.
  - Verificar que os administradores conseguem criar novos eventos. Verificar que os administradores conseguem alterar o estado dos utilizadores registados.

### 6.2.1 Estratégia

Na tabela 6.2 pode ser visualizada a estratégia escolhida para levar a cabo os testes funcionais.

**Tabela 6.2:** Testes funcionais

Objectivo do Teste:	Medir a qualidade funcional dos componentes do sistema, confrontando o resultado com o que é esperado. Inclui entrada de dados, processamento e resposta.
Técnica:	<p>Executar cada caso de uso, usando dados válidos e inválidos, para verificar o seguinte:</p> <ul style="list-style-type: none"> <li>• Quando são utilizados dados inválidos, são geradas mensagens de erro ou aviso.</li> <li>• Quando são utilizados dados válidos, os resultados são os esperados.</li> </ul>
Garantia de finalização:	Todos os acessos à base de dados funcionam correctamente e sem corromper os dados.

## 6.3 Testes de segurança

Para que os testes de segurança sejam bem sucedidos é necessário:

- Verificar que os utilizadores não conseguem aceder a funcionalidades para as quais não têm permissões.

### 6.3.1 Estratégia

Na tabela 6.3 pode ser visualizada a estratégia escolhida para levar a cabo os testes de segurança.

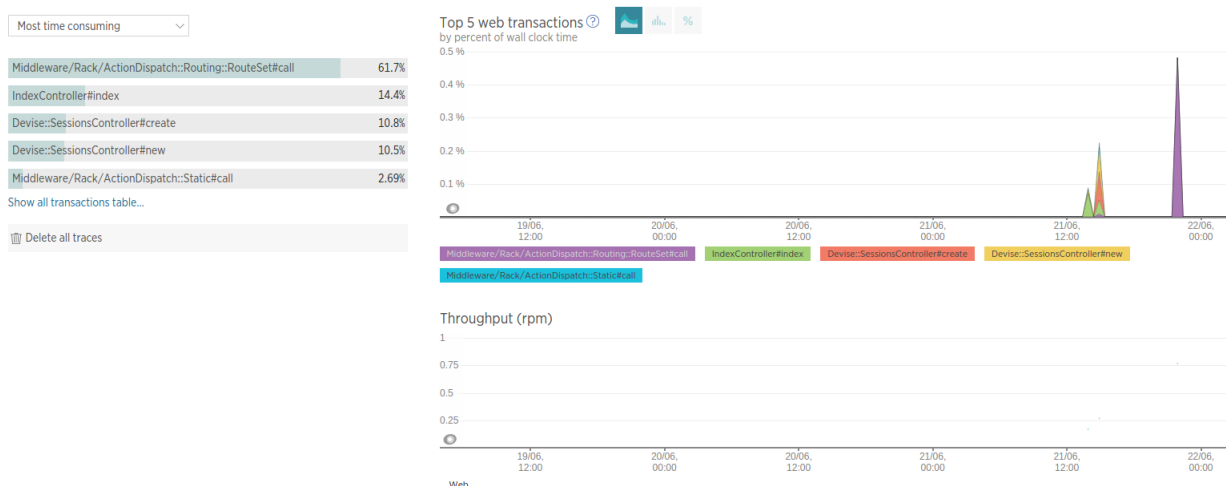
Outras questões de segurança foram descritas na secção 3.1.2 de requisitos não funcionais referentes à segurança.

**Tabela 6.3:** Testes de segurança

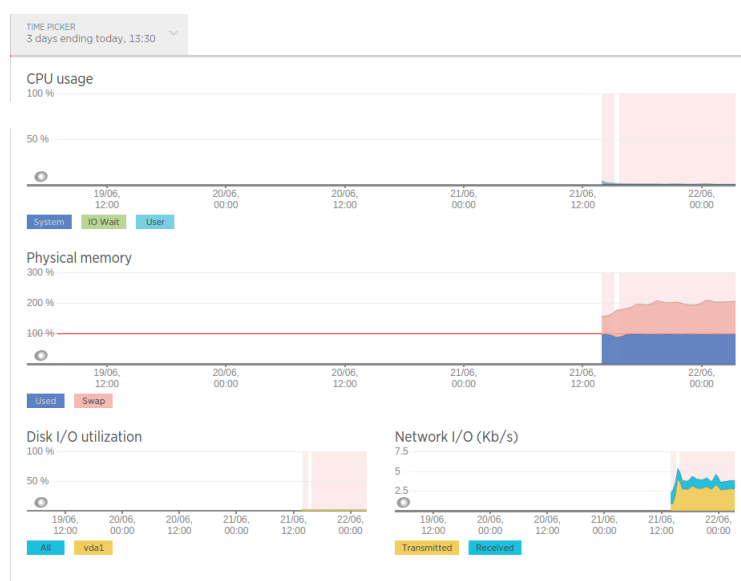
Objectivo do Teste:	Garantir que cada tipo de utilizador tem acesso apenas às funcionalidades devidas.
Técnica:	Testar, efectuando login como utilizador normal e como administrador e verificar se as permissões estão diferenciadas da forma correcta.
Garantia de finalização:	Após os testes constatar que as permissões estão definidas correctamente.

## 6.4 Testes de performance

Nesta secção são apresentados os testes à performance da aplicação web, para tal foram utilizadas duas ferramentas, a New Relic, que nos apresenta dados acerca da aplicação web e ainda do servidor, como é possível ver nas figuras 6.1 e 6.2, respectivamente.



**Figura 6.1:** Dados de transacções da aplicação web



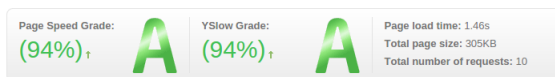
**Figura 6.2:** Dados de servidor

Como referido anteriormente não é possível nesta fase do projecto de dissertação de mestrado efectuar medições de confiança, pois o número de utilizadores não é elevado. No entanto foram efectuadas algumas medições que oferecem alguma segurança quanto à performance esperada ser atingida.

Na figura 6.3 é possível verificar ainda as medições da ferramenta GTMetrix<sup>1</sup> que permite efectuar medições à performance da página, dividindo as medições por

<sup>1</sup>Para mais informações consultar: <http://gtmetrix.com/>

categorias e avaliando a aplicação web com uma nota, neste caso foi conseguido a nota **A**, apresentando uma melhor performance do que sites como `www.uc.pt` e `www.portaldasfinancas.gov.pt/pt/home.action` que apresentam ratings de page speed de C e E, respectivamente. O relatório completo pode ser consultado no apêndice D Na ferramenta PageSpeed da Google<sup>2</sup> foi obtida uma pontuação de 89%, como pode ser visualizado na imagem 6.4, significando que a aplicação web desenvolvida conseguiu superar o que era pedido pelos requisitos não funcionais. 3.1.2



**Figura 6.3:** Rating da aplicação web na ferramenta GTMetrix

87 / 100 Resumo de sugestões

**Figura 6.4:** Rating da aplicação web na ferramenta PageSpeed

## 6.5 Testes de usabilidade

Os testes de usabilidade não faziam parte dos objectivos iniciais da dissertação de mestrado, no entanto foi efectuado um estudo e uma preparação que irá permitir que estes sejam efectuados durante os primeiros tempos de funcionamento da aplicação web.

Aos utilizadores seleccionados pela sua faixa etária e género será pedido que efectuem tarefas e, para cada uma serão anotadas características da forma como o utilizador a levou a cabo. As medições serão efectuadas assinalado "NC" no caso de o utilizador não ter conseguido completar a tarefa e com uma medida temporal que represente o tempo que o utilizador demorou até completar essa mesma tarefa. A tabela com as tarefas e características a serem medidas pode ser consultada no anexo A para os testes de utilização e no anexo B para as de administração.

Será ainda pedido feedback aos utilizadores, que deverá ser anotada numa outra tabela, que pode ser vista no anexo C. O Feedback será anotado sob a forma de concordância com afirmações pré-construídas que devem ser classificadas de 0 a 5, em que 0 significa "Não concordo" e 5 significa "Concordo plenamente".

Para os testes de usabilidade foram seleccionadas quatro faixas etárias, com 5 testes por cada faixa, pois é o suficiente para descobrir 75% dos erros [40] As faixas etárias seleccionadas após algum estudo daquelas que são mais comumente utilizadas para amostras mais pequenas [41] [42]:

- 15-24
- 25-34
- 35-44
- 45-65

<sup>2</sup>Para mais informações consultar: <https://developers.google.com/speed/pagespeed>





# Capítulo 7

## Conclusão

Neste capítulo é apresentada a conclusão, dividida em duas secções. Em primeiro lugar irá ser apresentado o trabalho a efectuar no futuro, e de seguida serão apresentadas as considerações finais acerca da dissertação de mestrado.

### 7.1 Trabalho futuro

Este é um projecto cuja dimensão obriga a que o trabalho não se restrinja àquele que se realizou durante a duração da dissertação de mestrado. Assim, nesta secção será apresentado o trabalho que irá ser efectuado com vista a melhorar alguns aspectos da aplicação web de gestão de associados desenvolvida no âmbito da dissertação de mestrado.

#### 7.1.1 Testes de usabilidade

No futuro serão efectuados testes de usabilidade com grupos de pessoas seleccionados a partir da sua faixa etária e género. A metodologia com que estes testes serão efectuados foi descrita na secção ?? e será efectuada após o deployment da aplicação web no servidor final.

#### 7.1.2 Deployment

O deployment efectuado durante o decorrer do projecto de dissertação de mestrado foi realizado num servidor temporário registado pelo autor no serviço oferecido pelo Digital Ocean. No futuro a aplicação será colocada num dos servidores oficiais da AISTI, preparado para o efeito, onde deverão ser efectuadas as configurações necessárias para que tudo funcione sem percalços.

#### 7.1.3 Manutenção

O autor oferecerá apoio na manutenção da aplicação web até que este se torne quase autónoma. Desta forma a AISTI não terá que se preocupar com potenciais problemas da aplicação web de gestão de associados.

## 7.2 Considerações finais

No desenvolvimento deste trabalho foi necessário passar pelas diferentes fases de um processo de engenharia de software.

Inicialmente foi levada a cabo uma análise de concorrência onde se analisaram sete aplicações de gestão de associados, seis das quais online. Desta análise nasceu uma lista de funcionalidades que poderiam ser implementadas no projecto, para além daquelas já definidas no plano desta dissertação de mestrado.

Após esta fase foram efectuados estudos de forma a definir a arquitectura do sistema. Toda a análise de requisitos foi efectuada tendo como base uma reunião com o representante da AISTI, em que foram também discutidos que funcionalidades poderiam ser incluídas na solução final. Os requisitos levaram a uma descrição dos casos de uso, onde se torna possível perceber a forma como os utilizadores irão interagir com o sistema.

Após a definição dos casos de uso foram desenhados os diagramas necessários à descrição da arquitectura do sistema, tais como o diagrama de entidades e relacionamentos, que descreve a arquitectura da base de dados que foi implementada durante o projecto; os diagramas representativos do modelo de arquitectura MVC; e por fim o diagrama de componentes onde são definidos os componentes do sistema e as suas funções;

Todos os módulos de software foram desenvolvidos e integrados, encontrando-se já implementados num servidor registado pelo autor. Durante o desenvolvimento foram efectuados testes unitários e funcionais às funcionalidades da aplicação web. No futuro serão ainda efectuados testes de usabilidade para certificar que a aplicação web está desenvolvida da melhor forma para utilização por parte dos associados da AISTI.

# Apêndice A

## Questionário de usabilidade para utilizadores

### A.1 Informação Pessoal

Nome:

Ocupação:

Idade:

Género:

### A.2 Tarefas

1. Efectuar o registo no site
2. Confirmar a sua conta
3. Efectuar login no site
4. Consultar factura por pagar
5. Simular pagamento online
6. Upload de comprovativo de pagamento
7. Ver lista de eventos
8. Registar em evento
9. Criar nova conversa com outro utilizador
10. Gerar cartão de sócio

### A.3 Tabela de questionário



# Apêndice B

## Questionário de usabilidade para administradores

### B.1 Informação Pessoal

Nome:

Ocupação:

Idade:

Género:

### B.2 Tarefas

1. Confirmar factura como paga
2. Alterar o estado de um utilizador
3. Gerar etiquetas para envio de cartas
4. Gerar uma carta PDF
5. Eliminar utilizador
6. Criar novo evento

### B.3 Tabela de questionário

	A	B	C	D	E	F	G	H	I
1	<b>Testes de Administração</b>								
2	<b>Ocupação</b>	<b>Gênero</b>	<b>Idade</b>	<b>T1</b>	<b>T2</b>	<b>T3</b>	<b>T4</b>	<b>T5</b>	<b>T6</b>
3	A definir	F	15-24						
4	A definir	M	15-24						
5	A definir	F	15-24						
6	A definir	M	15-24						
7	A definir	F	15-24						
8	A definir	M	25-34						
9	A definir	F	25-34						
10	A definir	M	25-34						
11	A definir	F	25-34						
12	A definir	M	25-34						
13	A definir	F	35-44						
14	A definir	M	35-44						
15	A definir	F	35-44						
16	A definir	M	35-44						
17	A definir	F	35-44						
18	A definir	M	45-65						
19	A definir	F	45-65						
20	A definir	M	45-65						
21	A definir	F	45-65						
22	A definir	M	45-65						

# Apêndice C

## Questionário de Feedback

### C.1 Informação Pessoal

Nome:

Ocupação:

Idade:

Género:

### C.2 Tarefas

1. Os menus de interacção encontram-se bem construídos.
2. É fácil perceber como funciona a aplicação web.
3. O design da aplicação é simples e compreensível.
4. Consegui entender como navegar entre páginas.
5. Foi simples efectuar o registo na página.
6. Foi fácil pagar a quota anual.
7. Consegui facilmente seleccionar e registar-me num evento.
8. Compreendi as mensagens de erro.
9. O Conteúdo das páginas está de acordo com o esperado.
10. Os textos são legíveis e do tamanho ideal para leitura.

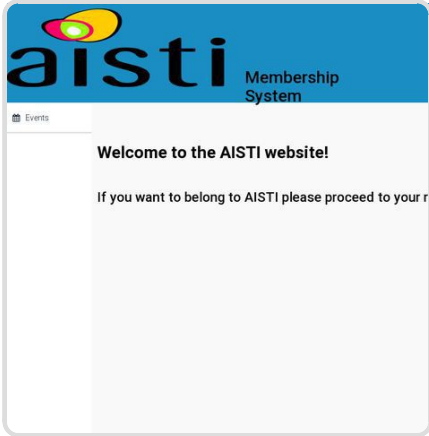
### C.3 Tabela de questionário





# Apêndice D

## Relatório de GTMetrix



## Performance Report for:

<http://46.101.38.163/>

Report generated: Monday, June 22, 2015, 4:38 PM -0700

Test Server Region: Vancouver, Canada

Using: Firefox (Desktop) 25.0.1, Page Speed 1.12.16, YSlow 3.1.8

Page Speed Grade:

**(94%)** ↑ Avg: 80%

**A**

YSlow Grade:

**(94%)** ↑ Avg: 78%

**A**

Page load time: 1.46s | Total page size: 305KB | Total number of requests: 10

### Priority Issues (Top 5)

Leverage browser caching	F (20)	↓ Avg Score: 62%	Server	High
Defer parsing of JavaScript	C (71)	↑ Avg Score: 57%	JS	High
Specify a cache validator	B (80)	↓ Avg Score: 93%	Server	High
Optimize the order of styles and scripts	B (85)	↓ Avg Score: 93%	CSS/JS	High
Minify HTML	A (97)	↑ Avg Score: 94%	Content	High

### How does this affect me?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, **Google has announced that they are using page speed in their ranking algorithm.**

### About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google Page Speed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

### About the Developer



GTmetrix is developed by the good folks at **Gossamer Threads**, a Vancouver-based company with over 16 years experience in web technology. [www.gossamer-threads.com](http://www.gossamer-threads.com)

### What do these grades mean?

This report is an analysis of your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The **grades you see represent** how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

### What's in this report?

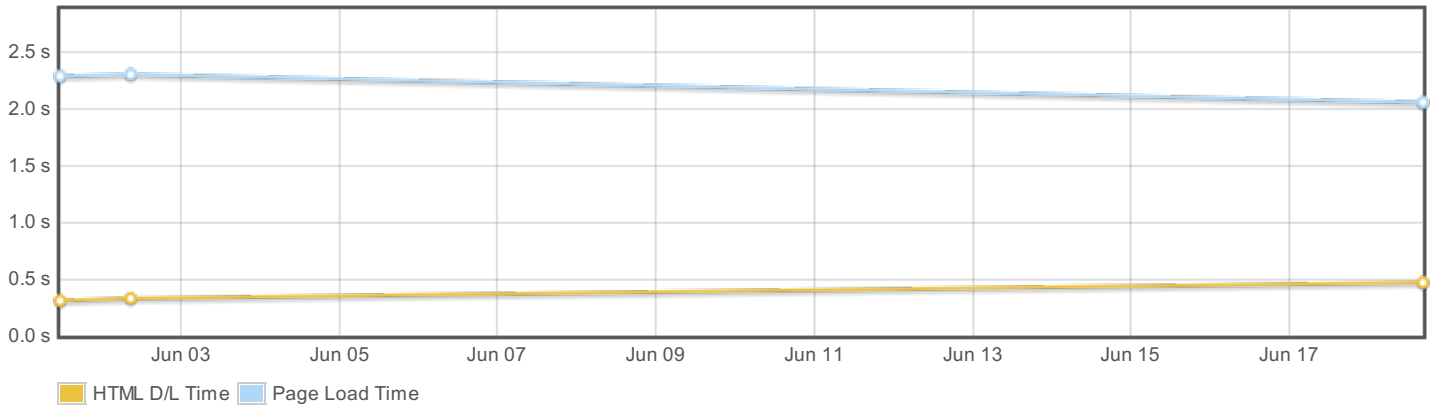
This report covers basic to technical analyses on your page. It is categorized under many headings:

- **Executive:** Overall score information and Priority Issues
- **History:** Graphed history of past performance
- **Waterfall:** Graph of your site's loading timeline
- **Technical:** In-depth Page Speed & YSlow information

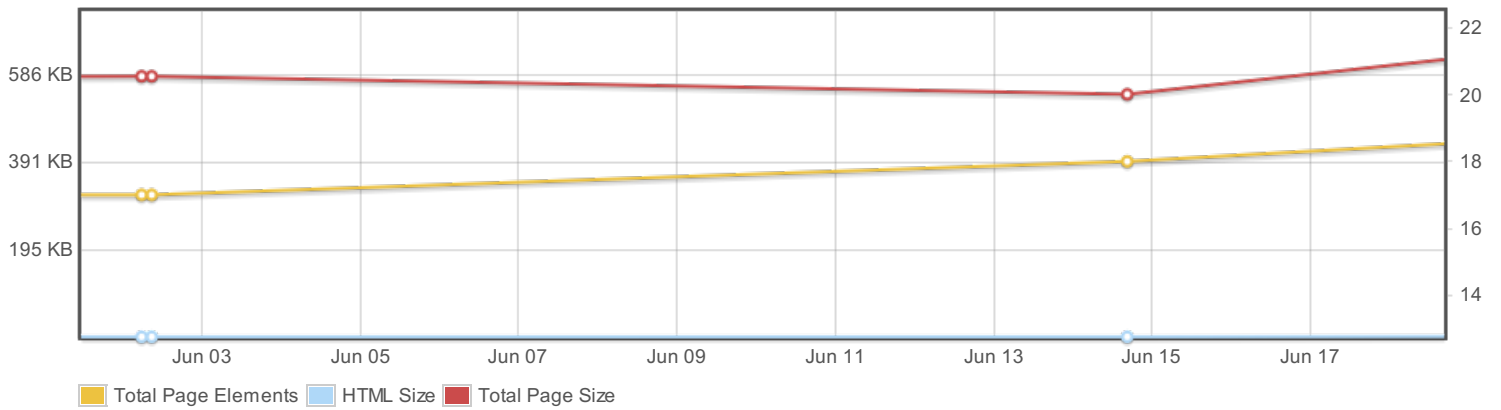
These will provide you with a snapshot of your performance.

## History

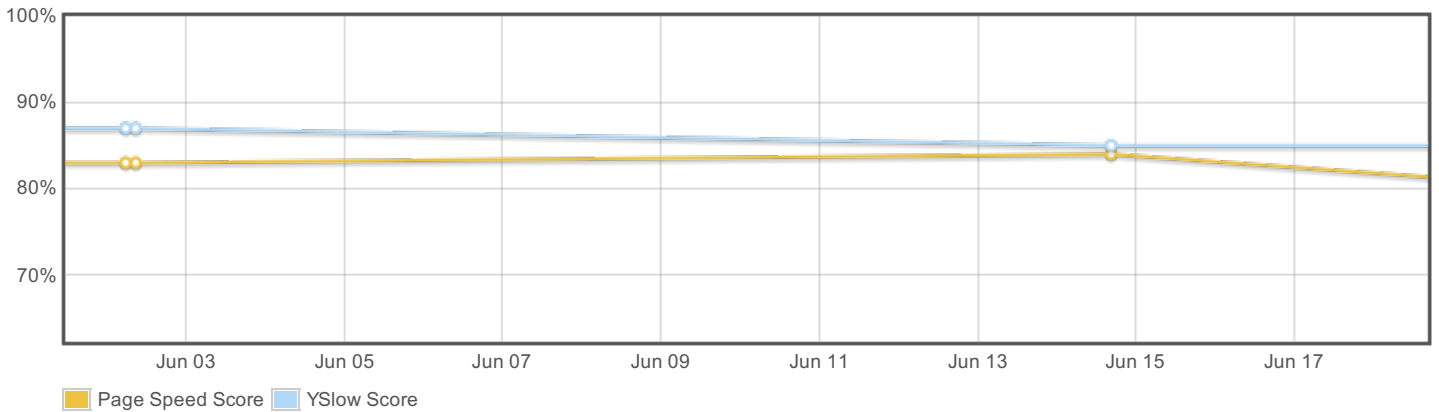
### Page load times



### Page sizes and request counts

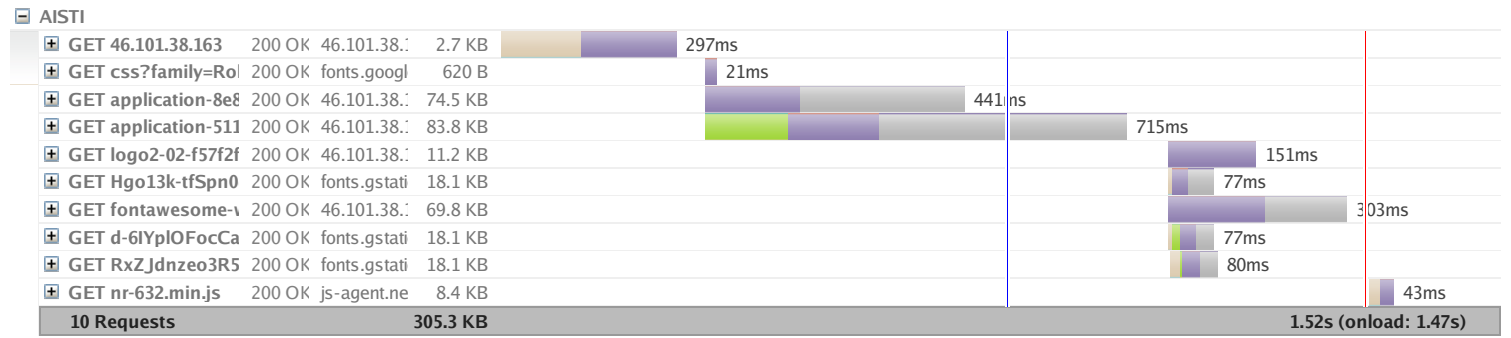


### Page Speed and YSlow scores



## Waterfall

The waterfall graph displays the loading behaviour of your site in Firefox. It can be used to discover simple issues such as 404's or more complex issues such as external resources blocking page rendering.



## Page Speed Recommendations

RECOMMENDATION	GRADE	RELATIVE	TYPE	PRIORITY
Leverage browser caching	F (20)	↓ Avg Score: 62%	Server	High
Defer parsing of JavaScript	C (71)	↑ Avg Score: 57%	JS	High
Specify a cache validator	B (80)	↓ Avg Score: 93%	Server	High
Optimize the order of styles and scripts	B (85)	↓ Avg Score: 93%	CSS/JS	High
Minify HTML	A (97)	↔ Avg Score: 94%	Content	High
Minify CSS	A (99)	↑ Avg Score: 88%	CSS	High
Minify JavaScript	A (99)	↑ Avg Score: 91%	JS	High
Avoid bad requests	A (100)	↔ Avg Score: 98%	Content	High
Avoid a character set in the meta tag	A (100)	↑ Avg Score: 94%	Content	High
Avoid landing page redirects	A (100)	↔ Avg Score: 97%	Server	High
Enable gzip compression	A (100)	↑ Avg Score: 84%	Server	High
Enable Keep-Alive	A (100)	↔ Avg Score: 97%	Server	High
Inline small CSS	A (100)	↔ Avg Score: 95%	CSS	High
Inline small JavaScript	A (100)	↔ Avg Score: 97%	JS	High
Minimize redirects	A (100)	↑ Avg Score: 90%	Content	High
Minimize request size	A (100)	↔ Avg Score: 99%	Content	High
Optimize images	A (100)	↑ Avg Score: 78%	Images	High
Put CSS in the document head	A (100)	↔ Avg Score: 100%	CSS	High
Remove query strings from static resources	A (100)	↑ Avg Score: 86%	Content	High
Serve resources from a consistent URL	A (100)	↔ Avg Score: 95%	Content	High
Serve scaled images	A (100)	↑ Avg Score: 81%	Images	High
Specify a Vary: Accept-Encoding header	A (100)	↑ Avg Score: 91%	Server	High
Specify a character set early	A (100)	↔ Avg Score: 97%	Content	High
Specify image dimensions	A (100)	↑ Avg Score: 48%	Images	High
Avoid CSS @import	A (100)	↔ Avg Score: 96%	CSS	Medium
Combine images using CSS sprites	A (100)	↑ Avg Score: 82%	Images	Medium
Prefer asynchronous resources	A (100)	↔ Avg Score: 96%	JS	Medium

## YSlow Recommendations

RECOMMENDATION	GRADE	RELATIVE	TYPE	PRIORITY
Add Expires headers	E (56)	↑ Avg Score: 26%	Server	High
Use a Content Delivery Network (CDN)	C (70)	↑ Avg Score: 12%	Server	Medium
Use cookie-free domains	B (85)	↑ Avg Score: 46%	Cookie	Low
Avoid empty src or href	A (100)	⚡ Avg Score: 100%	Content	High
Make fewer HTTP requests	A (100)	↑ Avg Score: 37%	Content	High
Compress components with gzip	A (100)	↑ Avg Score: 72%	Server	High
Minify JavaScript and CSS	A (100)	↑ Avg Score: 72%	CSS/JS	Medium
Avoid URL redirects	A (100)	↑ Avg Score: 83%	Content	Medium
Make AJAX cacheable	A (100)	⚡ Avg Score: 100%	JS	Medium
Put CSS at the top	A (100)	⚡ Avg Score: 100%	CSS	Medium
Remove duplicate JavaScript and CSS	A (100)	⚡ Avg Score: 100%	CSS/JS	Medium
Put JavaScript at bottom	A (100)	⚡ Avg Score: 100%	JS	Medium
Avoid AlphasLoader filter	A (100)	⚡ Avg Score: 97%	CSS	Medium
Avoid HTTP 404 (Not Found) error	A (100)	⚡ Avg Score: 100%	Content	Medium
Reduce the number of DOM elements	A (100)	↑ Avg Score: 93%	Content	Low
Do not scale images in HTML	A (100)	⚡ Avg Score: 100%	Images	Low
Use GET for AJAX requests	A (100)	⚡ Avg Score: 100%	JS	Low
Avoid CSS expressions	A (100)	⚡ Avg Score: 98%	CSS	Low
Reduce DNS lookups	A (100)	↑ Avg Score: 72%	Content	Low
Reduce cookie size	A (100)	⚡ Avg Score: 100%	Cookie	Low
Make favicon small and cacheable	A (100)	⚡ Avg Score: 100%	Images	Low
Configure entity tags (ETags)	A (100)	↑ Avg Score: 71%	Server	Low
Make JavaScript and CSS external	(n/a)		CSS/JS	Medium

# Bibliografia

- [1] Alistair Cockburn. *Writing Effective Use Cases*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 2000.
- [2] Holly Arrow and Joseph E McGrath. Membership matters how member change and continuity affect small group structure, process, and performance. *Small group research*, 24(3):334–361, 1993.
- [3] Irwin Altman, Anne Vinsel, and Barbara B Brown. Dialectic conceptions in social psychology: An application to social penetration and privacy regulation. *Advances in experimental social psychology*, 14:107–160, 1981.
- [4] Robert C Ziller. Toward a theory of open and closed groups. *Psychological bulletin*, 64(3):164, 1965.
- [5] Civilai Terawatanavong and Cynthia M. Webster. Organisational members' commitment to professional associations. page 7, 2005.
- [6] Thomas W Gruen, John O Summers, and Frank Acito. Relationship marketing activities, commitment, and membership behaviors in professional associations. *Journal of marketing*, 64(3):34–49, 2000.
- [7] CB Bhattacharya. When customers are members: customer retention in paid membership contexts. *Journal of the academy of marketing science*, 26(1):31–44, 1998.
- [8] Regulamento interno da aisti. page 3, 2012.
- [9] Estatutos da aisti. page 6.
- [10] Adam Thomson. Accor seeks bigger slice of digital pie, 2014. Disponível online em [ft.com/intl/cms/s/0/dfaee666-55e8-11e4-a3c9-00144feab7de.html](http://ft.com/intl/cms/s/0/dfaee666-55e8-11e4-a3c9-00144feab7de.html) no dia 15 de Dezembro de 2014.
- [11] Cláudia Bancaleiro. Uber avaliada em 40 mil milhões de dólares após investimento recente, 2014. Disponível online em [publico.pt/tecnologia/noticia/uber-avaliada-em-40-mil-milhoes-de-dolares-apos-investimento-recente-1678553](http://publico.pt/tecnologia/noticia/uber-avaliada-em-40-mil-milhoes-de-dolares-apos-investimento-recente-1678553) no dia 15 de Dezembro de 2014.
- [12] Cláudia Bancaleiro. Uber pode ser a aplicação mais valiosa do mercado mas também é a mais polémica, 2014. Disponível online em [publico.pt/tecnologia/noticia/uber-pode-ser-a-aplicacao-mais-valiosa-do-mercado-mas-tambem-e-a-mais-polemica-1678958](http://publico.pt/tecnologia/noticia/uber-pode-ser-a-aplicacao-mais-valiosa-do-mercado-mas-tambem-e-a-mais-polemica-1678958) no dia 15 de Dezembro de 2014.

- [13] Dmitriy Bychkov. Desktop vs. web applications: A deeper look and comparison. 2013. Disponível online em [seguetech.com/blog/2013/06/07/desktop-vs-web-applications-deeper-comparison](http://seguetech.com/blog/2013/06/07/desktop-vs-web-applications-deeper-comparison) no dia 15 de Dezembro de 2014.
- [14] Agência Lusa. Metade dos utilizadores acede à internet com dispositivo móvel. Disponível online em [zap.aeiou.pt/metade-dos-utilizadores-acede-a-internet-com-dispositivo-movel-47760](http://zap.aeiou.pt/metade-dos-utilizadores-acede-a-internet-com-dispositivo-movel-47760) no dia 15 de Dezembro de 2014.
- [15] Jeff Smith. Desktop applications vs. web applications. Disponível online em [streetdirectory.com/travel\\_guide/114448/programming/desktop\\_applications\\_vs\\_web\\_applications.html](http://streetdirectory.com/travel_guide/114448/programming/desktop_applications_vs_web_applications.html) no dia 15 de Dezembro de 2014.
- [16] Piero Fraternali. Tools and approaches for developing data-intensive web applications: a survey. *ACM Computing Surveys (CSUR)*, 31(3):227–263, 1999.
- [17] Ethan Marcotte. *Responsive web design*. Editions Eyrolles, 2011.
- [18] Max Starkov and Mariana Safer. Adaptive vs. full responsive design: How to choose the right option for your property website., 2014. Disponível online em <http://4hoteliers.com/features/article/8741> no dia 15 de Dezembro de 2014.
- [19] The top 7 free and open source membership management software products, 2013. Disponível online em <http://blog.capterra.com/top-7-free-open-source-membership-management-software-products/>, consultado pela última vez em 10 de Fevereiro de 2015.
- [20] Leah Readings. Doing magic with membership management, 2013. Disponível online em <http://www.business2community.com/mobile-apps/doing-magic-with-membership-management-review-of-wild-apricot-0500449>, consultado pela última vez em 10 de Fevereiro de 2015.
- [21] 15 ideas to provide more value for your chamber of commerce members, 2013. Disponível online em <http://www.yourmembership.com/articles/149/15-Things-to-Do-For-Your-Chamber-of-Commerce-Members.htm>, consultado pela última vez em 10 de Fevereiro de 2015.
- [22] Elizabeth Hull, Ken Jackson, and Jeremy Dick. *Requirements Engineering*. Springer London, Boston, MA, USA, 2nd edition.
- [23] Ruby on rails security guide. Disponível online em <http://guides.rubyonrails.org/security.html>, consultado pela última vez em 17 de Junho de 2015.
- [24] Paulo Rupino da Cunha. Uml: Casos de uso em engenharia de software 1; dei; uc, 2005.
- [25] *OOSE.book*, chapter Modeling with UML. 2009.
- [26] Trygve Reenskau. Mvc. *XEROX PARC*, 1979. Disponível online em <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>, consultado pela última vez em 8 de Maio de 2015.
- [27] Donald Bell. Uml basics: The component diagram. *IBM Global Services*, 2004.



- [28] Ruby on rails: What it is and why we use it for web applications, 2014. Disponível online em <http://www.bitzesty.com/blog/2014/7/ruby-on-rails-what-it-is-and-why-we-use-it-for-web-applications>, consultado pela última vez em 15 de Fevereiro de 2015.
- [29] Kresimir Bojic. What are the benefits of ruby on rails? after two decades of programming, i use rails. Disponível online em <http://www.toptal.com/ruby-on-rails/after-two-decades-of-programming-i-use-rails>, consultado pela última vez em 15 de Fevereiro de 2015.
- [30] Five reasons why we use ruby on rails. Disponível online em <http://www.infront.com/blogs/the-infront-blog/2013/1/4/five-reasons-why-we-use-ruby-on-rails>, consultado pela última vez em 14 de Dezembro de 2014.
- [31] LindsayT. Ruby on rails vs django, 2014. Disponível online em <https://blog.udemy.com/ruby-on-rails-vs-django/>, consultado pela última vez em 15 de Fevereiro de 2015.
- [32] Dillon Buchanan. Play!framework vs ruby on rails, 2013. Disponível online em <http://dillonbuchanan.com/programming/playframework-vs-ruby-on-rails/>, consultado pela última vez em 15 de Fevereiro de 2015.
- [33] Rails vs django vs play: Battle of frameworks. Disponível online em <http://diogonunes.com/blog/rails-vs-django-vs-play-frameworks/> no dia 15 de Dezembro de 2014.
- [34] Sharon Machlis. Perl vs. php vs. ruby. Disponível online em <http://www.computerworld.com/article/2467421/application-development/perl-vs--php-vs--ruby.html> no dia 15 de Fevereiro de 2014.
- [35] Jason Chandler. What php programmers can learn from ruby on rails, 2013. Disponível online em <http://www.drdoobbs.com/web-development/what-php-programmers-can-learn-from-ruby/240161412>, consultado pela última vez em 15 de Fevereiro de 2015.
- [36] Dan Cederholm. Why sass?, 2013. Disponível online em <http://alistapart.com/article/why-sass>, consultado pela última vez em 17 de Fevereiro de 2015.
- [37] Christopher Gimmer. Top 5 reasons to use bootstrap, 2014. Disponível online em <https://bootstrapbay.com/blog/reasons-to-use-bootstrap/>, consultado pela última vez em 17 de Fevereiro de 2015.
- [38] Natalia David. 11 reasons to use twitter bootstrap, 2013. Disponível online em <http://www.sitepoint.com/11-reasons-to-use-twitter-bootstrap/>, consultado pela última vez em 17 de Fevereiro de 2015.
- [39] Dustin Boswell. Storing user passwords securely: hashing, salting, and bcrypt, 2012. Disponível online em <http://dustwell.com/how-to-handle-passwords-bcrypt.html>, consultado pela última vez em 16 de Junho de 2015.
- [40] Jakob Nielsen. Usability inspection methods. chapter Heuristic Evaluation, pages 25–62. John Wiley & Sons, Inc., New York, NY, USA, 1994.
- [41] Susan E. Wyse. 5 examples of survey demographic questions, 2012. Disponível online em <http://www.snapsurveys.com/blog/5-survey-demographic-question-examples/>, consultado pela última vez em 22 de Junho de 2015.

- [42] Standardized survey classifications - individuals. Disponível online em <http://www.pgagroup.com/standardized-survey-classifications.html>, consultado pela última vez em 22 de Junho de 2015.