

UNIVERSIDADE ESTADUAL PAULISTA

Faculdade de Ciências - Bauru

Bacharelado em Ciência da Computação

Marcelo de Almeida Franco

Abordagem técnica sobre os aspectos de uma
rede social colaborativa

UNESP

2014

Marcelo de Almeida Franco

Abordagem técnica sobre os aspectos de uma rede
social colaborativa

Monografia apresentada junto à disciplina Projeto e Implementação de Sistemas II, do curso de Bacharelado em Ciência da Computação, Faculdade de Ciências, Unesp, campus de Bauru, como parte do Trabalho de Conclusão de Curso.

Orientador: Prof. Dr. Marco Antônio Rahal Sacoman

Co-orientador: Felipe Teixeira Avelar

UNESP

2014

Marcelo de Almeida Franco

Abordagem técnica sobre os aspectos de uma rede social colaborativa

Monografia apresentada junto à disciplina Projeto e Implementação de Sistemas II, do curso de Bacharelado em Ciência da Computação, Faculdade de Ciências, Unesp, campus de Bauru, como parte do Trabalho de Conclusão de Curso.

BANCA EXAMINADORA

Marco Antônio Rahal Sacoman
Professor Doutor
DCo – FC – UNESP – Bauru
Orientador

Simone das Graças Domingues Prado
Professora Doutora
DCo – FC – UNESP – Bauru

Andréa Carla Gonçalves Vianna
Professora Doutora
DCo – FC – UNESP – Bauru

Bauru, 23 de Janeiro de 2014.

Resumo

Com o avanço da indústria de jogos no mercado, mostrou-se evidente a necessidade de profissionais atuando nessa área. Tendo isso em vista, esse trabalho tenta criar um meio onde os aspirantes e profissionais em criação de jogos possam interagir. Para chegar a esse ponto foi construído um sistema baseado nas tecnologias atuais, e nos paradigmas da Engenharia de Software.

Assim ao longo dessa monografia é estudado o processo de desenvolvimento do sistema, fazendo uso de diferentes tecnologias, e mostrando como poderia funcionar a troca de informação e conhecimento entre as pessoas amantes da criação de jogos.

Palavras-chave: *Crowdsourcing*, Redes sociais, Jogos *indie*.

Abstract

With the advancement of the gaming industry in the market, became evident the need for professionals working in this area. With this in mind, this paper attempts to create an environment where aspirants and professionals in creating games can interact. To get to this point, was built one system based on current technologies, and paradigms of software engineering system.

So throughout this monograph the process of the system development is studied, using different technologies, and showing how it can work, the exchange of information and knowledge among the games creation lovers.

Keywords: Crowdsourcing, Social Networks, Indie Games.

LISTA DE FIGURAS

Figura 1: Imagem do Commodore 64.	12
Figura 2: Página inicial do site artistshare.....	14
Figura 3: Página inicial do site kickstarter.	15
Figura 4: Imagem do jogo Broken Age.	16
Figura 5: Interface mais recente do programa CompuServe.	17
Figura 6: Interface da rede social Friendster.	18
Figura 7: Fluxo de dados feito pelo Django (MVC).....	23
Figura 8: Página padrão da área administrativa.....	24
Figura 9: Interface MySQL Workbench.....	25
Figura 10: Interface do StarUML.....	26
Figura 11: Interface do Balsamiq Mockups.	27
Figura 12: Diagrama de caso de uso.	27
Figura 13: Modelo do banco de dados.	29
Figura 14: Página home do protótipo.....	30
Figura 15: Página do projeto do protótipo.	30
Figura 16: Página do usuário do protótipo.	31
Figura 17: Classe CustomUser criada para modificação do usuário padrão.	32
Figura 18: Classe de gerenciamento do CustomUser.....	33
Figura 19: Parte da classe Projeto mostrando os dois identificadores.....	34
Figura 20: Parte da classe UsuProj mostrando identificadores e o tipo de usuário.	34
Figura 21: Página de cadastro de usuário.	37
Figura 22: Página de cadastro do currículo.	37
Figura 23: Página de visualização do usuário.	38
Figura 24: Página de busca do usuário.	39
Figura 25: Primeira etapa do cadastro de projeto.....	40
Figura 26: Segunda etapa do projeto.....	40
Figura 27: Janela de preenchimento da vaga.	41
Figura 28: Página do projeto.	42
Figura 29: Página dos projetos do usuário.....	43
Figura 30: Página de busca dos projetos.	44
Figura 31: Página inicial.	45
Figura 32: Página do fórum.	45
Figura 33: Página do tópico.....	46
Figura 34: Gráfico de exaustão de testes.....	47
Figura 35: Diagrama de estado das ações dos usuários com os projetos.	48

LISTA DE TABELAS

Tabela 1: Cronograma proposto para desenvolvimento das atividades.	20
---	----

SUMÁRIO

1	Introdução	9
1.1	Problema	10
1.2	Justificativa	10
1.3	Objetivos	11
1.3.1	Objetivos geral	11
1.3.2	Objetivos específicos	11
2	Referencial teórico	12
2.1	Jogos independentes	12
2.2	Crowdfundind	13
2.2.1	História	13
2.2.2	Influência nos jogos independentes	15
2.2.3	Crowdsourcing	16
2.3	Redes sociais	17
3	Projeto e desenvolvimento	19
3.1	Pré-projeto	19
3.1.1	Idealização	20
3.1.2	Definições	21
3.1.2.1	Paradigmas	21
3.1.2.2	Ferramentas	22
3.1.3	Diagrama de caso de uso	27
3.2	Desenvolvimento	28
3.2.1	Planejamento	28
3.2.2	Programação	31
3.2.2.1	Modelos	31
3.2.2.1.1	Modelos de usuário	32
3.2.2.1.2	Modelos de projeto	33
3.2.2.1.3	Modelos gerais	35
3.2.2.2	Páginas	35
3.2.2.2.1	Páginas de usuário	36
3.2.2.2.2	Páginas de projeto	39
3.2.2.2.3	Páginas gerais	44
3.2.3	Testes	46
3.3	Pós-projeto	47
4	Conclusões	49
4.1	Problemas e soluções	49
4.2	Trabalhos futuros	49
4.3	Considerações finais	50
	Referências bibliográficas	51
	Bibliografia complementar	52

Capítulo 1

1 Introdução

Atualmente um dos maiores meios de comunicação é a internet, sendo capaz de realizar a troca de informações entre diversas pessoas em diferentes lugares do mundo. Sendo que, essa troca de informações, muitas vezes é realizada através das redes sociais.

No mês de janeiro de 2013 estimou-se que passaram 322.319.612 usuários pelo *Facebook* – rede social baseada na internet, sendo o terceiro *website* mais acessado do mundo -, e que o tempo médio gasto por seus usuários foi de 6 horas e 17 minutos durante o mês (Nielsen, 2013). Assim revelando que as redes sociais são capazes de agrupar um grande número de pessoas, que por meio dela trocam ideias, informações, conhecimento e sociabilizam.

Além das redes sociais, outra aplicação *web* que, atualmente reúne uma grande quantidade de usuários ativos são os *websites* baseados em sistema de oportunidade. Sendo o mais conhecido o *Kickstarter*, um portal *online* onde o usuário cadastra seu projeto e prediz quanto de dinheiro é necessário para torna-lo realidade, assim, as pessoas que gostarem da concepção ajudam o idealizador a conseguir esse montante.

Apesar de parecer uma ideia que não atrairia muitos usuários colaboradores, dados mostram que: 2.241.475 pessoas chegaram a contribuir com US\$319.786.629 em 18.109 projetos que foram concluídos com sucesso (Kickstater, 2012). Através da ideia das redes sociais e dos sistemas de oportunidade, constrói-se um sistema capaz de agrupar pessoas para colaborarem entre si, com o intuito de concluir um projeto. Este sistema de colaboração funciona no modelo de *crowdsourcing* – utilização de inteligência e conhecimento coletivo e voluntário -, assim realizando o agrupamento de usuários conforme seus interesses.

Os jogos digitais atraíram grande interesse da população nos últimos anos, prova disso é o aumento significativo no faturamento desse setor da indústria, hoje maior que a indústria do cinema. Como consequência desse aumento, fica evidente o crescimento na produção de jogos, uma atividade que era dominada por grandes empresas. Porém muitas pessoas aspiram a criar os seus próprios jogos e, com a popularização da tecnologia isso se tornou cada vez mais real.

Ao investir na criação de jogos independentes, um novo ramo de produção surgiu e ficou conhecido como jogos *indie* – jogos criados por uma pessoa ou pequenas equipes sem qualquer apoio das publicadoras de jogos. Essa nova divisão cresceu e criou jogos que atingiram milhões de pessoas por todo o mundo como, por exemplo, *Braid* (2008), *Minecraft* (2011) ou *Fez* (2012), grandes inspiradores para o crescimento da produção da indústria independente.

Visando esse ramo da produção de jogos independente e utilizando a ideia proposta anteriormente de *crowdsourcing*. Esse trabalho tem como objetivo construir um sistema capaz de realizar a interação e colaboração entre vários usuários para a construção de jogos eletrônicos, funcionando como uma rede social que agrupa os usuários conforme seus interesses e afinidades com um projeto para o desenvolvimento de jogos.

1.1 Problema

Embora atualmente o mercado de jogos *indie* seja grande e influente, ainda existem situações em que os desenvolvedores encontram dificuldade na criação de seus jogos. De acordo com Sampaio (2012), encontrar mão de obra qualificada no Brasil é uma das principais dificuldades na produção de jogos.

Apesar de possuir um grande número de profissionais formados em áreas derivadas da Tecnologia da Informação, o Brasil ainda não é autossuficiente para preencher as vagas disponíveis. Além da questão quantitativa, a maioria dos cursos superiores da área são de baixa qualidade formando, assim, profissionais incapacitados, conforme a reportagem publicada na revista EXAME (2012).

Assim, apesar de pessoas, físicas ou jurídicas, serem capazes de idealizar um jogo, na maioria das vezes, esse projeto se torna inviável pela falta de recursos.

Faz-se necessário que haja um meio de encontrar e comunicar profissionais e amantes interessados na área de jogos para que, assim, seja viável a difusão do conhecimento, a troca de ideias e a formação de equipes para o desenvolvimento de jogos digitais.

1.2 Justificativa

A ideia de *crowdsourcing* não é nova.

Existem algumas ferramentas que proporcionam aos profissionais o uso do esforço coletivo e voluntário, uma das mais utilizadas são as aplicações *web*, que agrupam diferentes profissionais em projetos cadastrados pelos próprios usuários. Muitos desses profissionais gostam de acessar essas aplicações em seu tempo ocioso e participar de projetos, assim, adquirindo para si novos conhecimentos e experiências, além de contribuir para o desenvolvimento de uma ferramenta completamente nova, que possa auxiliar uma grande quantidade de pessoas no futuro.

Porém a maior parte dos projetos cadastrados nas aplicações *web* existentes não tem como objetivo a produção de jogos. Vislumbrando o potencial crescente do mercado nessa área, uma aplicação *web* que reunisse, em projetos, profissionais interessados na criação de jogos, resultaria no aumento da qualidade e quantidade de produções.

Pensando no problema de agrupamento, a solução que utiliza a internet de forma mais interessante, é um *website*.

1.3 Objetivos

1.3.1 Objetivo geral

Desenvolver um sistema capaz de realizar o cadastro de usuários e projetos, e criar uma interface para a comunicação e reunir os profissionais interessados no desenvolvimento de jogos, além de possibilitar a divulgação de seus trabalhos.

1.3.2 Objetivos específicos

- Estruturar o sistema a partir da análise de requisitos, diagramas UML e modelagem de banco de dados;
- Desenvolver um método capaz de identificar os principais interesses do usuário;
- Criar uma interface que siga os padrões de aplicações já implementadas para torná-la mais simples para o usuário;
- Construir um sistema capaz de reunir e realizar a comunicação entre usuários, observando as preferências de cada usuário.

Capítulo 2

2 Referencial teórico

2.1 Jogos independentes

A história dos jogos digitais não é antiga, alguns historiadores consideram seu início em 1958. Sendo que, no princípio das produções, os jogos eram criados de forma independente por não possuir um mercado próprio. Porém desde o início da produção em larga escala a maioria dos jogos vem sendo criada por grandes empresas, capazes de realizar a distribuição desses jogos.

Apenas em meados dos anos 2000, que através da distribuição de programas de computadores *sharewares*- programa de computador gratuito, porém com algum tipo de limitação-, os jogos independentes foram capazes de expandir seu mercado, gerando assim títulos que encantaram muitos jogadores.

Além dos *sharewares*, as ferramentas para o auxílio da produção de jogos também foram importantes no crescimento de criações independentes. Desde as primeiras ferramentas que auxiliavam o desenvolvimento de jogos para Commodore 64 - computador doméstico lançado pela Commodore em 1982 (Figura 1) - até as mais modernas como o *Game Maker* - *software* simples para criação de jogos de diferentes plataformas-.



Figura 1: Imagem do Commodore 64.
Fonte: Imagem retirada do site *oldcomputers*¹

¹ Endereço eletrônico da imagem: <http://oldcomputers.net/pics/C64combo.jpg>. Acesso em 19 out. 2013

O maior crescimento da indústria de jogos independentes ocorreu na segunda metade dos anos 2000, quando a internet permitiu a distribuição *online*, mudando o mercado da distribuição física para a digital. Antes o criador do jogo deveria ser capaz de produzir e distribuir os jogos em CDs e DVDs, enquanto, atualmente ele pode utilizar a internet para distribuir seu jogo digitalmente, sendo até mais eficiente do que a distribuição física da mídia.

Outro fator que ajudou no crescimento da produção independente de jogos foi o aumento do número de pessoas que jogam através das redes sociais, assim, abrindo um novo nicho de jogadores casuais. Pessoas que antes não jogavam por falta de tempo, começaram a jogar casualmente nas redes sociais.

A maioria dos jogos independentes, pelo fato de suas produtoras não possuírem muito recursos, são criados com capacidade gráfica inferior aos jogos de grandes produtoras. Sendo assim, a indústria independente tem que oferecer uma experiência inovadora aos seus jogadores, por essa razão grande parte dos jogos independentes surgem de ideias criativas, muitas das vezes chamados de *art games*, por experimentarem novos elementos que os jogos comuns não o fazem.

Através das inúmeras iniciativas para o aumento da produção independente que ocorreram nos últimos anos, e através do aumento de interesse dos jogadores, o mercado foi capaz de se estabilizar e produzir diversos títulos que atraem muitos jogadores.

2.2 Crowdfunding

2.2.1 História

Michael Sullivan (2006) define *crowdfunding*, inspirado pelo *crowdsourcing* - modelo de utilização de inteligência e conhecimento de forma coletiva e voluntária-, descreve a cooperação coletiva, atenção e confiança de pessoas que pela rede se reúnem e levantam fundos, normalmente pela internet, a fim de apoiar os esforços iniciados por outras pessoas ou organizações.

O *crowdfunding* possui uma longa história na humanidade, desde meados de 1700 já existiam estudos que demonstravam os seus conceitos, porém só nas últimas décadas a humanidade foi capaz de modelar o *crowdfunding* e torna-lo aplicável a sociedade moderna.

O primeiro sucesso de uso do *crowdfunding* foi registrado em 1997, quando a banda britânica Marillion conseguiu levantar dinheiro para reunir os integrantes e realizar uma turnê, somente através de doações realizadas pelos fãs através da internet. Motivados por esse novo método, muitos empresários investiram na criação de plataformas que utilizassem os conceitos do *crowdfunding*, sendo a primeira delas o *ArtistShare* (Figura 2).



Figura 2: Página inicial do site *artistshare*.

Fonte: Imagem retirada do site sellyourband.com/artistshare/²

Com o tempo o potencial de arrecadação de fundos pela a internet começou a ficar visível, e assim em 2000 surgiu o *JustGiving*, um site onde as pessoas poderiam entrar e doar dinheiro para a caridade, esse site tornou-se rapidamente um sucesso no Reino Unido e foi capaz de arrecadar mais de £700 milhões no total.

No entanto foi apenas em 2005 que surgiu a primeira plataforma na qual empresários eram capazes de emprestar fundos para iniciativas de projetos, essa plataforma chamava-se Kiva e apresentou-se como uma forma extremamente eficiente para a arrecadação de fundo, sendo capaz de levantar mais de US\$65 milhões para realização de iniciativas de projetos, e alcançado uma taxa de reembolso de 98,83% aos investidores.

Algumas plataformas para empréstimo de dinheiro surgiram depois da Kiva, mas não foram bem sucedidas inicialmente por utilizar conceitos muito novos, os quais a sociedade não estava acostumada.

Em 2009, uma plataforma que tornaria possível a arrecadação de fundos para iniciativas foi lançada, chamada *Kickstarter* (Figura 3). Ela, que se tornou a maior das plataformas para arrecadação de fundos trouxe uma novidade: os contribuintes das iniciativas não receberiam seu dinheiro de volta, entretanto, dependendo do nível de arrecadação alcançada e do montante doado, estes poderiam receber uma recompensa definida pelo dono do projeto como retorno.

² Endereço eletrônico da imagem: <http://sellyourband.com/wp-content/uploads/2012/04/Screen-Shot-2012-04-04-at-8.00.24-PM-e1333595506414.png>. Acesso em 19 out. 2013.

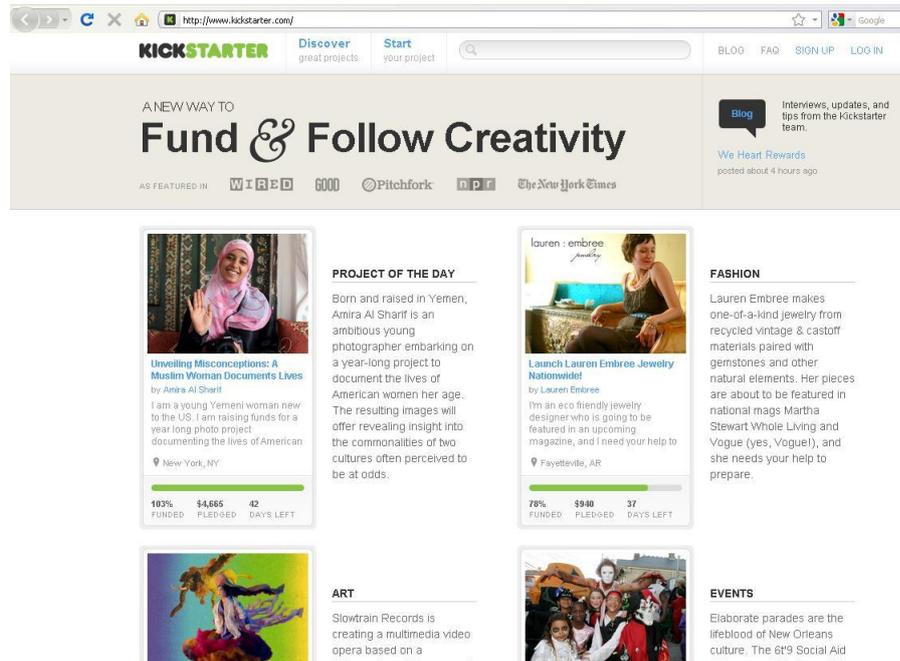


Figura 3: Página inicial do site *kickstarter*.
 Fonte: Imagem retirada do site *havaspr.com*³

Atualmente o sucesso de plataformas que utilizam o *crowdfunding* é bastante visível, possibilitando o financiamento de diversos projetos. Muitos deles que antes seriam inviáveis, vislumbram uma nova perspectiva, e tornam-se realidade.

2.2.2 Influência nos jogos independentes

Os jogos digitais são produtos que necessitam de inúmeros recursos para serem desenvolvidos, e normalmente, os produtores independentes enfrentam a dificuldade de obter esses recursos.

Assim, desde o surgimento das plataformas de *crowdfunding*, muitos produtores começaram a cadastrar seus projetos de jogos nessas plataformas, com o intuito de obter fundos e adquirir os recursos necessários para a produção de seus jogos.

O uso das plataformas *crowdfunding* com o intuito de angariar fundos para o desenvolvimento de jogos é uma prática recente. Até o ano de 2011, apenas jogos que precisavam de poucos recursos eram colocados nessas plataformas. Porém, em fevereiro de 2012, o jogo *Broken Age* (Figura 4) foi cadastrado no *Kickstarter* e chegou a obter um fundo de US\$3 milhões, sendo um dos maiores fundos obtidos até aquele momento.

³ Endereço eletrônico da imagem: <http://havaspr.com/us/wp-content/uploads/2010/10/kickstarter.jpg>. Acesso em 19 out. 2013.



Figura 4: Imagem do jogo *Broken Age*.
 Fonte: Imagem retirada do site pxlbyte.com⁴

Após se apresentar como um modelo eficiente, muitos outros projetos passaram a ser cadastrados nessas plataformas, permitindo que vários jogos fossem concluídos.

2.2.3 Crowdsourcing

O *crowdsourcing* tem o mesmo formato do *crowdfunding*, porém ao invés de procurar a obtenção de fundos, esse método procura a obtenção de mão-de-obra disposta a oferecer seu conhecimento, inteligência e experiência para pessoas ou organizações desenvolverem um projeto. Outra forma de definir *crowdsourcing* seria o processo de conseguir trabalho e recursos de um grupo de pessoas, segundo o artigo “What is Crowdsourcing?” exposto no *website Dailycrowdsource*⁵.

Um grande exemplo de *crowdsourcing* atualmente é a *Wikipedia*, enciclopédia que ao invés de contratar escritores e editores para escrever o seu conteúdo, deixou que a própria audiência criasse a informação. O resultado disso é a enciclopédia mais abrangente que existe no mundo.

O princípio do *crowdsourcing* é que várias pessoas juntas são capazes de produzir algo melhor do que uma sozinha. Isso ocorre, pois na diversidade do grupo, o número de ideias e habilidades diferentes é maior, assim aumentando as chances de um resultado final superior comparado ao de uma única pessoa.

⁴ Endereço eletrônico da imagem: <http://pxlbyte.com/wp-content/uploads/2013/05/Double-Fine-Bundle-Broken-Age.jpg>. Acesso em 19 out. 2013.

⁵ Autor desconhecido. **What is Crowdsourcing?**. Dailycrowdsource, 2012. Disponível em <<http://dailycrowdsource.com/crowdsourcing-basics/what-is-crowdsourcing>>. Acesso em 20 abr. 2013.

Como o *crowdsourcing* não se baseia na arrecadação de fundos, ele acaba sendo utilizado muito para iniciativas abertas a todos, assim possibilitando que todo tipo de pessoa colabore com a iniciativa. Tornando-se assim um eficiente meio de troca de informações entre diferentes áreas ou setores de negócios e entre diferentes níveis de conhecimento.

2.3 Redes sociais

Os primeiros registros de serviços capazes de sociabilizar dados surgiram em 1969 com a tecnologia *dial-up* e o lançamento do *CompuServe* (Figura 5) – serviço de conexão à internet em nível internacional. Alguns anos depois aconteceu o envio do primeiro email.



Figura 5: Interface mais recente do programa *CompuServe*.

Fonte: Imagem retirada do site [gsbrown.org](http://www.gsbrown.org)⁶

Porém o primeiro modelo de rede social surgiu com a criação do *Bulletin Board System*, um sistema onde o usuário poderia convidar seus amigos para eventos e realizar anúncios pessoais. Essa tecnologia utilizava linhas telefônicas e um *modem* para transmitir dados.

Em 1985 a *America Online* começou a oferecer ferramentas para que as pessoas criassem perfis, onde elas se descreveriam e poderiam criar grupos com o intuito de realizar a troca de informações e discussões sobre algum tópico. A partir deste momento a rede social, como ela é conhecida hoje, começou a ser modelada.

⁶ Endereço eletrônico da imagem: <http://www.gsbrown.org/compuserve/the-newest-compuserve-1996-07/>. Acesso em 21 out. 2013.

Outra inovação marcante foi o lançamento do *GeoCities*, este serviço oferecia recursos para que seus usuários fossem capazes de criar suas páginas pessoais, categorizada de acordo com sua região.

Na mesma época no formato de compartilhar conhecimento, o *Classmate* surgiu com o objetivo de reunir grupos de antigos colegas de escola e faculdade, para trocar conhecimento ou marcar reencontros.

Em meados de 2000 o primeiro serviço a receber o status de “rede social” foi lançado, chamava-se *Friendster* (Figura 6), suas funções permitiam ao usuário transportar as suas amizades da vida real para a virtual.

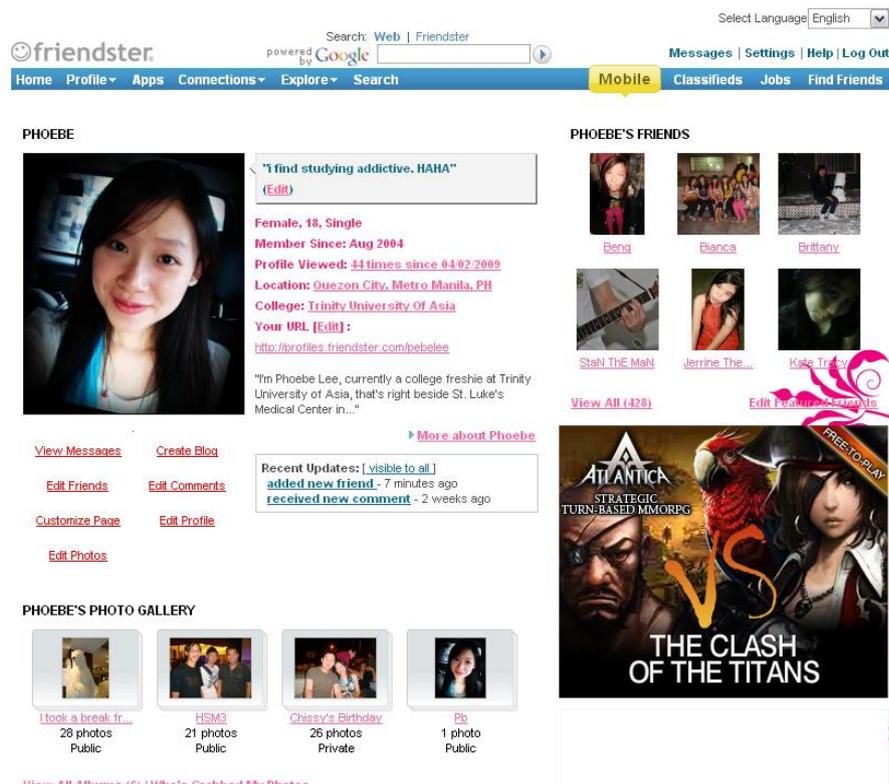


Figura 6: Interface da rede social *Friendster*.

Fonte: Imagem retirada do site dualipamedia.myblog.arts.ac.uk⁷

Desde 2004, quando a rede social *Facebook* foi criada, a sociedade observa um crescimento na quantidade de pessoas que utilizam redes sociais. Um dos principais benefícios que isso traz é a rápida troca de informações entre um grupo grande de pessoas. Apesar de conter muito conteúdo banal, as redes sociais mostram-se um grande meio de comunicação.

⁷ Endereço eletrônico da imagem: <http://dualipamedia.myblog.arts.ac.uk/files/2013/03/friendster2.png>. Acesso em 21 out. 2013.

Capítulo 3

3 Projeto e desenvolvimento

O projeto foi dividido em três etapas: pré-projeto, desenvolvimento e pós-desenvolvimento.

Na etapa de planejamento foram definidos todos os requisitos do projeto, organizando-os de modo adequado. Definiu-se a ideia, objetivo e motivação para desenvolvê-lo, foram escolhidas as ferramentas a serem utilizadas, e um estudo de cada uma delas foi realizado.

Toda documentação feita nessa etapa é necessária para evitar possíveis erros futuros e agilizar a correção no processo de desenvolvimento do sistema. Essa documentação também serve para dar uma noção geral de como o sistema deverá funcionar ao final do desenvolvimento. Para isso foi utilizado um diagrama UML de caso de uso.

Com todos requisitos analisados e ferramentas definidas foi iniciada a etapa de desenvolvimento. Primeiramente nessa etapa foi feito um planejamento de como seria estruturado o sistema, assim, definindo o modelo do banco de dados e um protótipo de *design* para auxiliar na criação das páginas.

Nessa etapa também foram aplicados os testes, após ser comprovado que o sistema não apresentava erros críticos avançou-se para a etapa final do projeto. Nessa última etapa foi feita uma documentação de tudo que foi produzido, visando facilitar a manutenção e evolução do sistema. Sendo que essa documentação final é mais abrangente do que as outras feitas durante o pré-projeto e desenvolvimento.

3.1 Pré-projeto

Na fase de pré-projeto foram definidos os objetivos do sistema, os elementos da Engenharia de Software capazes de melhorar e otimizar o progresso do projeto, e as ferramentas utilizadas. Todas essas definições auxiliaram na estruturação do sistema, para que, ao final do projeto o sistema estivesse funcional e adequado ao propósito inicial.

O texto a seguir aborda todas as etapas de definições e estruturação do sistema, dentre seus detalhes e suas características básicas. Assim, facilitando o entendimento do processo de criação e desenvolvimento de cada parte do projeto.

Inicialmente, para melhorar os processos do projeto como um todo, foi essencial a criação de um cronograma que mostrasse o tempo a ser gasto em cada tarefa, visando obedecer o tempo máximo de entrega (*deadline*).

O cronograma sofreu algumas alterações desde sua proposta inicial, isso ocorreu devido a realocação de tempo das tarefas, e pode ser visto resumidamente na Tabela 1.

Tabela 1: Cronograma para desenvolvimento das atividades.

Mês	1	2	3	4	5	6	7	8	9	10	11
Levantamento Bibliográfico	X	X	X	X							
Elaboração da Proposta	X	X									
Elaboração da Fundamentação Teórica		X	X	X							
Estudo das plataformas semelhantes			X								
Estudos de sistemas de feedbacking			X								
Análise de requisitos			X	X							
Diagramas UML				X							
Modelagem do banco de dados				X							
Desenvolvimento do protótipo					X	X	X	X			
Testes do protótipo						X	X	X			
Desenvolvimento da interface							X	X			
Adequação da interface ao protótipo							X	X	X		
Testes de funcionalidades e desempenho								X	X		
Elaboração da monografia								X	X	X	X

Após a definição do cronograma, os objetivos do projeto se tornaram mais claros. Além de tornar o processo de desenvolvimento mais ágil e simples.

3.1.1 Idealização

A ideia de criar um sistema para servir de ponte às pessoas interessadas na criação de jogos mostrou-se muito atraente. Pois, além de proporcionar um novo modo de divulgação entre os amantes de jogos, esse sistema possibilita as pessoas que não possuem muito tempo livre disponível ou não detêm o conhecimento técnico necessário para o desenvolvimento de um jogo, contribuir das mais variadas formas para um determinado projeto.

Assim, foi definido que o sistema deve estimular as pessoas através da possibilidade de cadastrar projetos de jogos. Ou seja, cada pessoa que deseja apresentar sua ideia como um jogo para outras pessoas, pode realizar o cadastro de um projeto, desde que essa pessoa seja cadastrada no sistema.

Portanto, existem dois tipos de cadastro no sistema, sendo eles: cadastro de usuário e de projetos.

Para o cadastro de usuários foi definido que a pessoa deveria inserir dados pessoais para futuras consultas, e um usuário e senha para autenticação.

Definiu-se também que, para mensurar o conhecimento, a pessoa deve, ao se cadastrar, escrever um currículo que servirá como método avaliativo no momento de aderir a um projeto.

Já no cadastro de projeto foi estabelecido que todos projetos têm como objetivo a criação de um jogo eletrônico, e deve ter suas características explicitas pelo criador através das ferramentas oferecidas pelo sistema.

Após criar o projeto, o mesmo é disponibilizado para busca, assim, o criador pode procurar pessoas para juntarem-se ao projeto e vice-versa. Para facilitar essa busca por parte dos usuários, uma das ferramentas adicionadas ao cadastro de projetos são vagas disponíveis, que faz o criador do projeto descrever as vagas existentes como, por exemplo, para uma vaga de programador, ele pode apresentar a linguagem de desenvolvimento, sistemas que deverá ser compatível, entre outras características.

Ao se unir a um projeto uma pessoa se torna membro do projeto, e preenche uma vaga de acordo com sua área de atuação. Dessa forma, através dos projetos, o relacionamento entre os usuários do sistema se forma.

Na parte de comunicação foram definidos que dois tipos de fóruns. Um deles seria parte de cada projeto cadastrado, ou seja, cada projeto tem um fórum próprio. O outro é o fórum geral, onde os usuários podem trocar informações e ideias sem ser necessário a relação com algum projeto que esteja em desenvolvimento ou acerca de tópicos gerais que não se adequam a um único projeto.

Após essa formulação do sistema, tornou-se possível a concepção de um documento que apresentasse seu funcionamento. Esse documento também serviu para definições dos elementos de Engenharia de Software envolvidos no projeto. Todas as funções do sistema serão apresentadas de forma mais clara e detalhada na seção 3.2.

3.1.2 Definições

Após apresentar a idealização do projeto foram explicitados os paradigmas de desenvolvimento, e também foi feito um estudo para abordar as melhores ferramentas a serem utilizadas.

Nas seções abaixo é apresentado como cada um desses elementos foram concebidos.

3.1.2.1 Paradigmas

Os paradigmas definidos para o projeto foram apenas os de Engenharia de Software. Depois de realizar algumas análises nos modelos de desenvolvimento, foi definido que o modelo utilizado seria o cascata ou sequencial. Este modelo foi escolhido pois ele oferece uma melhor estruturação do processo de

desenvolvimento, já que, uma tarefa só pode ser iniciada após o término de outra. O modelo cascata pode ser definido como:

O modelo cascata, algumas vezes chamado de ciclo de vida clássico, sugere uma abordagem sequencial e sistemática para o desenvolvimento de software, começando com o levantamento de necessidades por parte do cliente, avançando pelas fases de planejamento, modelagem, construção, emprego e culminando no suporte contínuo do software concluído. (PRESSMAN, 2011)

Também foram determinados os diagramas UML necessários para o desenvolvimento. Optou-se apenas pelo diagrama de caso de uso, como uma prévia do que seria o sistema.

Ainda definiu-se como seria o padrão de documentação. Sendo que ela foi redigida inteiramente em português, para que fosse possível a reutilização de elementos do desenvolvimento em projetos nacionais.

Ainda nessa etapa definiu-se os testes que seriam usados. Os testes escolhidos foram os testes de unidade, integração, funcional e interface. Teste de unidade é o teste mais básico dentre os tipos de testes; ele verifica a funcionalidade de um bloco de código específico. Já o teste de integração verifica se blocos reunidos funcionam da forma esperada. O teste de interface verifica se a navegabilidade e os objetivos da tela funcionam como especificados e se atendem da melhor forma ao usuário. Já o teste funcional testa as funcionalidades, requerimentos, e regras de negócio presentes na documentação (BASTOS et al., 2007).

3.1.2.2 Ferramentas

Após definir os paradigmas, e ter a documentação escrita, foram definidas as ferramentas que seriam utilizadas.

Inicialmente foram definidas as ferramentas para auxiliar na programação, para depois definir as ferramentas para documentação. A seguir é realizado um estudo acerca de cada uma dessas ferramentas, abordando os motivos para escolha delas.

Python e o framework Django

O *Django* é um *web framework*, escrito em *Python*, de alto nível e de código aberto, que segue o modelo *model-view-controller (MVC)*, e é utilizado para agilizar e melhorar o desenvolvimento *web*.

O *MVC* é um modelo de arquitetura de software que separa as informações e regras de negócio da interface com a qual o usuário interage. No *Django* esse modelo funciona seguindo os seguintes passos:

1. O usuário faz uma requisição (*HttpRequest*) que contém uma *url*;
2. O *handler* organiza a *url* e manda para os *middlewares*;

3. Os *middlewares* analisam a requisição e extrai diversas informações como, por exemplo, o usuário que está autenticado no momento;
4. Após os *middlewares* a requisição vai para o *url dispatcher*, que por sua vez analisa para qual *view* será acionada;
5. A *view* recebe todas as informações vindas da requisição e trata elas, fazendo buscas no banco de dados e/ou verificações, para então retornar uma resposta (HttpResponse);
6. Os *templates* recebem essa resposta da *view* e mostram na página as informações;
7. Por fim os modelos são as definições dos dados, ou seja, como os dados devem ser armazenados e tratados.

A figura 7 demonstra o fluxo de dados realizado pelo *Django*.

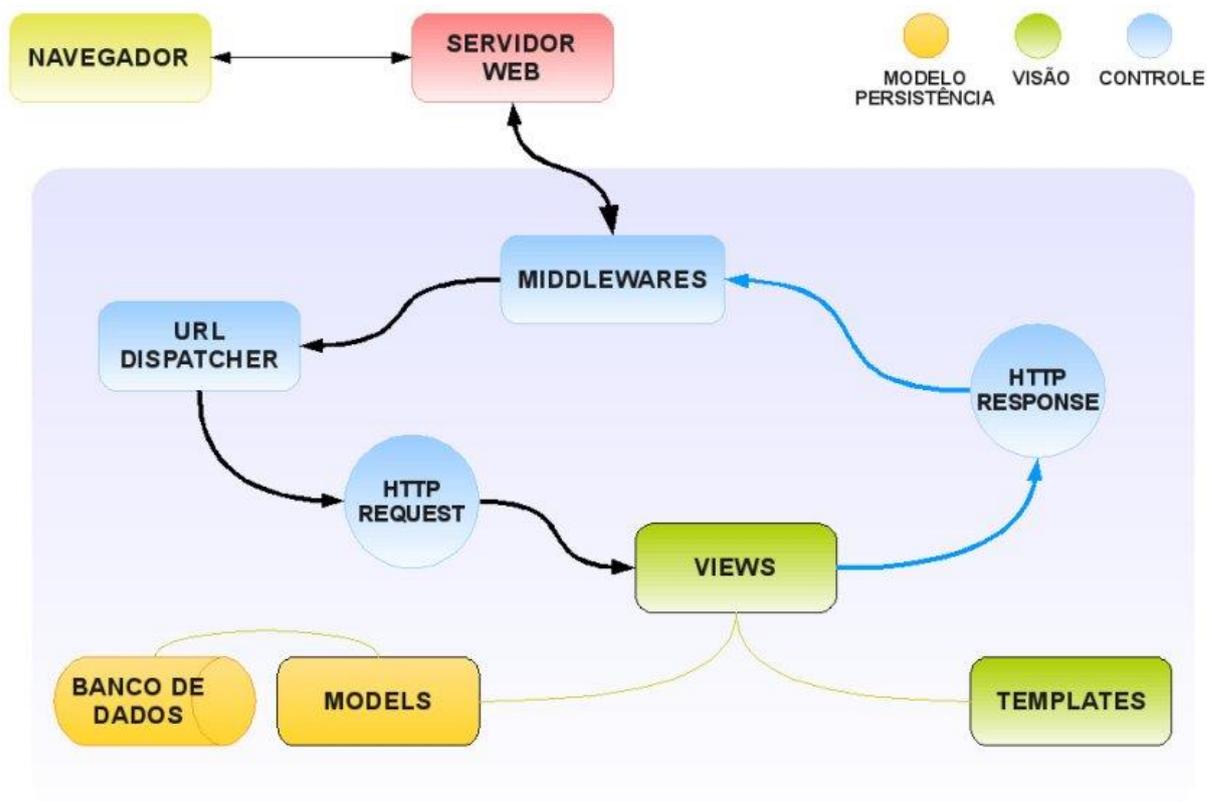


Figura 7: Fluxo de dados feito pelo *Django* (MVC).

Fonte: Imagem retirada do site aprendendodjango.com⁸

Django foi criado em 2003, por Adrian Holovaty e Simon Willison, ambos programadores *web* no *Lawrence Journal-World* que se interessavam por *Python*. O fato de trabalharem na produção e manutenção de *websites* de jornais, que precisava de atualizações extremamente rápidas e com prazos de entrega apertados, foi o que os levou a procurar uma forma eficiente e rápida de realizar a atualização dos *websites*.

⁸ Endereço eletrônico da imagem: <http://www.aprendendodjango.com/entendendo-como-o-django-trabalha/>. Acesso em 15 nov. 2013.

Assim em 2005 ambos haviam desenvolvido tanto o *framework* que decidiram lançá-lo como um programa aberto. Atualmente o *Django* é um estável projeto aberto que conta com milhares de usuários e contribuidores ao redor do mundo.

Por ter nascido no ambiente onde jornais são produzidos, o *Django* acaba tendo muitos recursos que ajudam em *websites* dinâmicos. Além da vantagem de ser um programa aberto, aumentando a facilidade de encontrar soluções para possíveis problemas que podem ocorrer no desenvolvimento de um *website*.

Além do descrito acima, uma das ferramentas importantes que o *Django* fornece de forma automática é a criação de uma área administrativa para o *website*. Essa ferramenta é importante para agilizar o processo de inserção de conteúdo no site, e sua página padrão pode ser vista na figura 8.

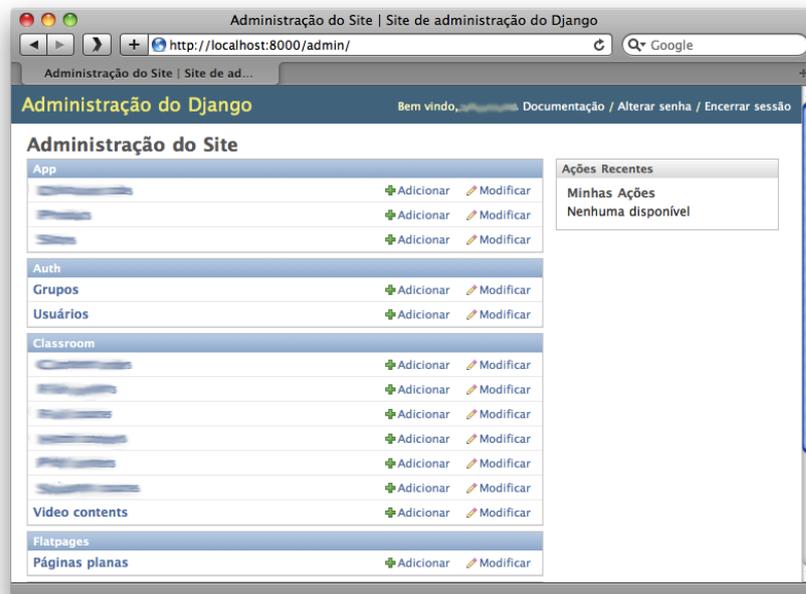


Figura 8: Página padrão da área administrativa.

Fonte: Imagem extraída do site osantana⁹

Também, duas importantes ferramentas genéricas fornecidas pelo *Django* para ajudar na criação de *websites*, são: “usuário” e “grupos”. Sendo que o usuário é basicamente um sistema de autenticação pronto, que pode ser customizado pelo programador. E o grupos um sistema de permissão, que restringe o acesso, bloqueando algumas ações de acordo com o grupo de acesso atrelado ao usuário. A partir disso, nota-se que o *Django*, em conjunto com o *Python*, destaca-se como melhor opção para o desenvolvimento de *websites*.

⁹ Endereço eletrônico da imagem: <http://osantana.me/2010/07/02/interface-admin-do-django-um-baita-quebra-galho/>. Acesso em 15 nov. 2013.

MySQL

O *MySQL* é o mais popular banco de dados de código aberto do mundo, possuindo mais de 100 milhões de cópias baixadas ou distribuída durante sua existência. Suas principais características são a ótima velocidade, confiabilidade e facilidade de uso, sendo assim uma boa escolha para aplicações *web*.

Muitas empresas optam por utilizar o *MySQL* para economizar e otimizar *websites* de negócios críticos e que necessitam de muito espaço. Pode ser um banco de dados de código aberto, suas documentações são gratuitas e milhares de programadores contribuem com correções ou acréscimos ao seu código original.

Na figura 9 pode ser vista a interface do *MySQL Workbench*, que é um software fornecido aos desenvolvedores para auxiliar na criação e administração do banco de dados, já que oferece uma interface mais amigável para controle e visualização.

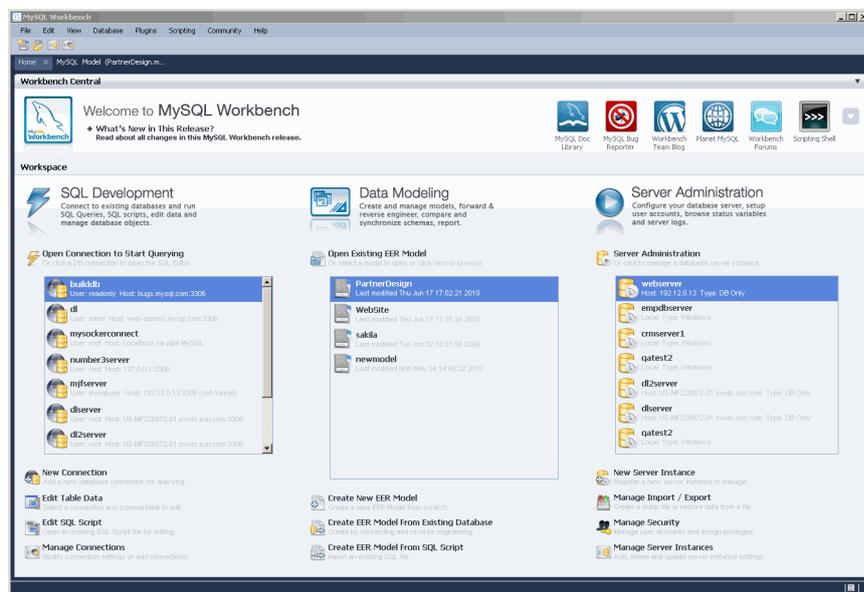


Figura 9: Interface *MySQL Workbench*.

Fonte: Imagem extraída do site [zwodnik](http://www.zwodnik.com)¹⁰

Pelas características citadas acima, o *MySQL* é considerado um dos melhores bancos de dados gratuitos para ser utilizado, já que é compatível com o *Django* e fornece o suporte mais adequado se comparado as outras opções gratuitas.

StarUML

Outra ferramenta de código aberto, com diversas características interessantes que foi utilizada, é o *StarUML*.

¹⁰ Endereço eletrônico da imagem: http://www.zwodnik.com/media/images/mysql-workbench_windows.gif. Acesso em 15 nov. 2013.

Além de possibilitar a criação de diagramas UML, esse programa permite ao usuário realizar engenharia direta (diagrama gera código) e engenharia reversa (código gera diagrama) para as linguagens Java, C++ e C#.

Os diagramas que são produzidos a partir dele seguem a padronização da UML 2.0 e fornece ao usuário uma interface simples que possibilita um trabalho rápido para o desenvolvimento da documentação UML (STARUML, 2012).

Essa ferramenta foi escolhida para criar o diagrama de caso de uso, assim, vislumbrando a maior parte das ações que usuário realizará dentro do sistema. E moldando como o sistema ficará em seus estágios finais.

Na figura 10 pode ser vista a interface do *StarUML*.

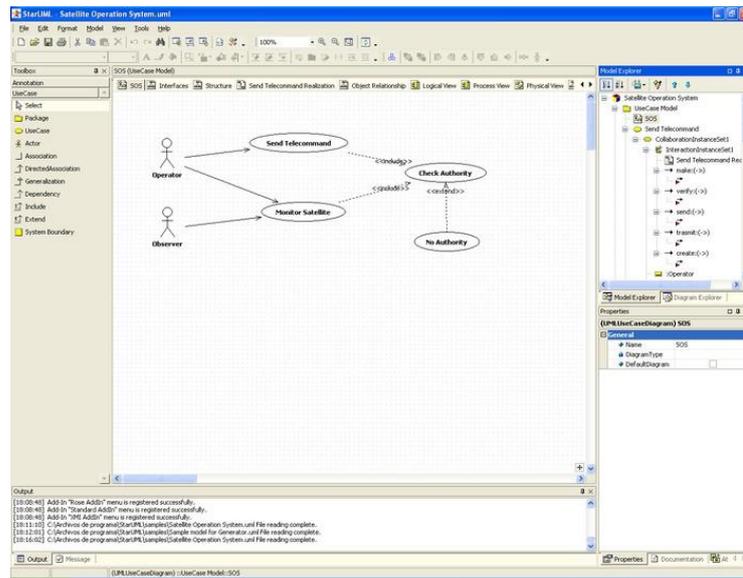


Figura 10: Interface do *StarUML*.

Fonte: Imagem extraída do site staruml.en.softonic.com¹¹

Balsamiq Mockups

O *Balsamiq Mockups* é uma ferramenta gráfica que auxilia no esboço de interfaces de usuário para *websites* e aplicações *web*, *desktop* e *mobile*. A ferramenta é focada na fase de idealização, assim, possuindo uma interface simples que permite ao usuário permanecer com foco na estrutura ao invés de ícones e cores.

Em alguns casos o *Balsamiq Mockups* chega a oferecer interatividade suficiente para substituir um protótipo. Assim, a análise retroativa do aplicativo desenvolvido pode ser iniciada antes de qualquer linha de código ser processada.

Optou-se pela utilização dessa ferramenta, para que fossem definidas as estruturas das páginas do *website*, facilitando posteriormente na programação e estilização. Também foi utilizado como forma de demonstrar o funcionamento do sistema em algumas apresentações.

¹¹ Endereço eletrônico da imagem: <http://screenshots.en.sftcdn.net/en/scrn/64000/64167/staruml-8.jpg>. Acesso em 15 nov. 2013.

Na figura 11 pode ser vista a interface do *Balsamiq Mockups*.

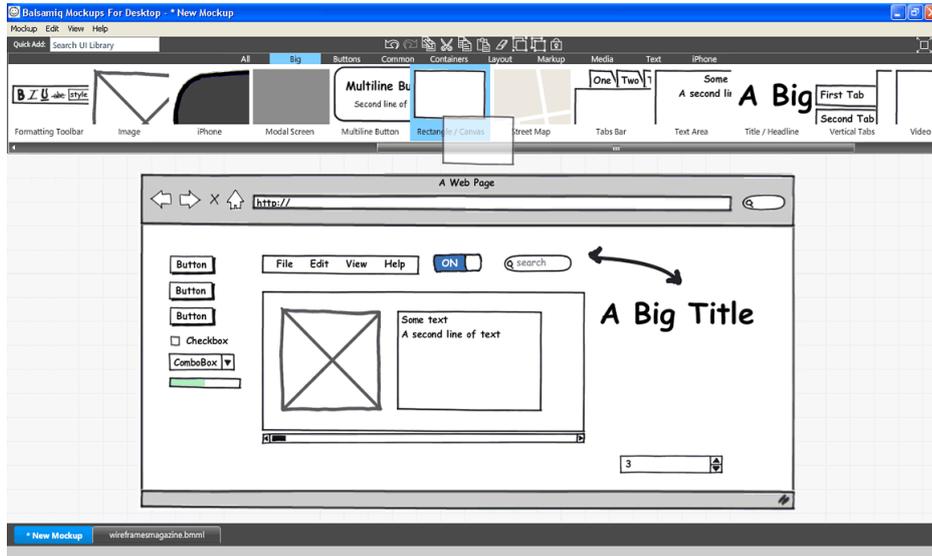


Figura 11: Interface do *Balsamiq Mockups*.

Fonte: Imagem retirada do site wireframes.linowski.ca¹²

3.1.3 Diagrama de caso de uso

Definido todos os aspectos de desenvolvimento do projeto, foi possível criar o diagrama UML de caso de uso, como pode ser visto na Figura 12.

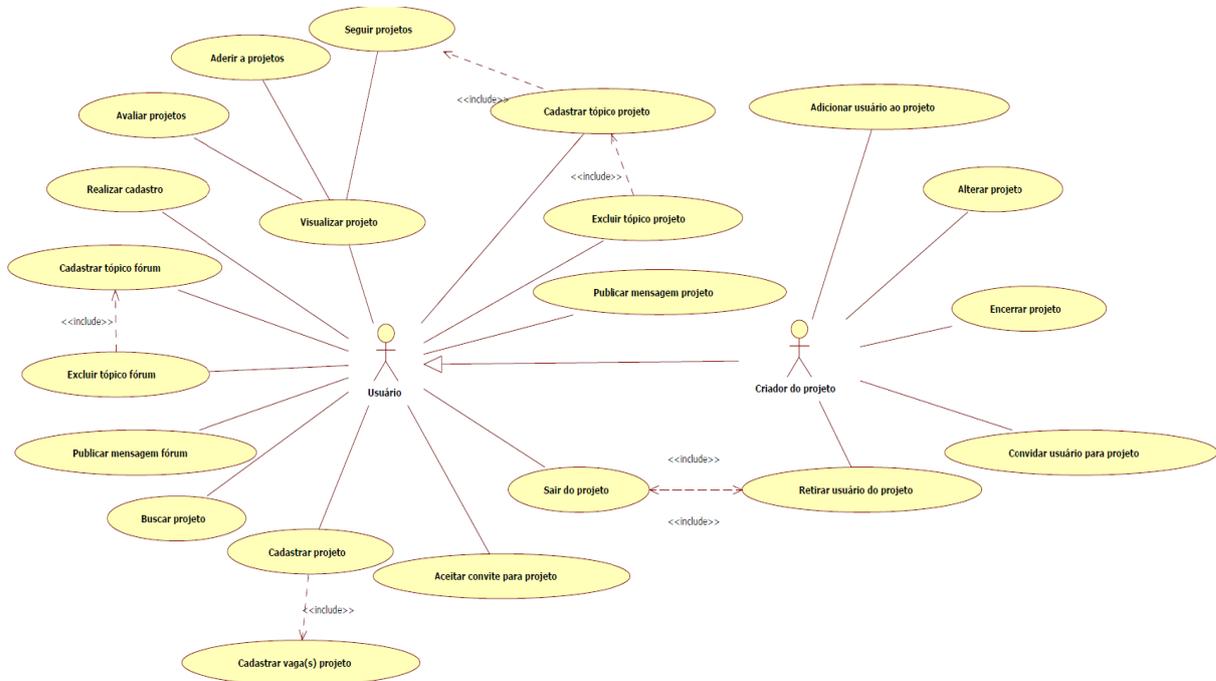


Figura 12: Diagrama de caso de uso.

¹² Endereço eletrônico da imagem: <http://wireframes.linowski.ca/wp-content/themes/darwin/images/full64.png>. Acesso em 15 nov. 2013.

Com essa documentação pronta foi possível iniciar a próxima fase do projeto: a fase de desenvolvimento.

3.2 Desenvolvimento

O desenvolvimento do projeto foi dividido em quatro partes: planejamento, programação, testes e documentação.

No planejamento foram concebidos alguns documentos de rascunhos das páginas do *website*, eles foram utilizados para criar estruturas de páginas pela ferramenta *Balsamiq Mockups*, descrita na seção 3.1.1.3. Essa documentação serviu de base para toda a codificação posterior que seria desenvolvida.

O período de programação foi a fase em que praticamente todo o sistema foi desenvolvido.

Inicialmente foi criada a base de dados a partir do modelo prescrito e, então, foram criadas as páginas do sistema. A partir dessa base e páginas, todos os eventos e métodos de funcionalidade do sistema foram criados.

Também nessa etapa foi desenvolvida a estruturação das páginas do website, ou seja, o *layout* das páginas foi organizado baseando-se no protótipo de *design* criado anteriormente. Todos os *layouts* foram criados perante análises de outros *websites* já conhecidos.

A fase de testes foi dividida em duas etapas, sendo elas uma antes e uma após a aplicação do *layout*. A primeira fase teve como objetivo testar os eventos e métodos criados. Já a segunda fase verificou todo o sistema após aplicação do *layout*. Essas fases são apresentadas de forma mais clara na seção indicada para esta etapa.

A documentação do desenvolvimento foi feita após sua conclusão, para que fosse explicado de forma mais coerente todas etapas do projeto.

3.2.1 Planejamento

Inicialmente no planejamento foi modelado o banco de dados. Para construir essa modelagem foi utilizada a ferramenta *MySQL Workbench*, descrita na seção 3.1.1.3.

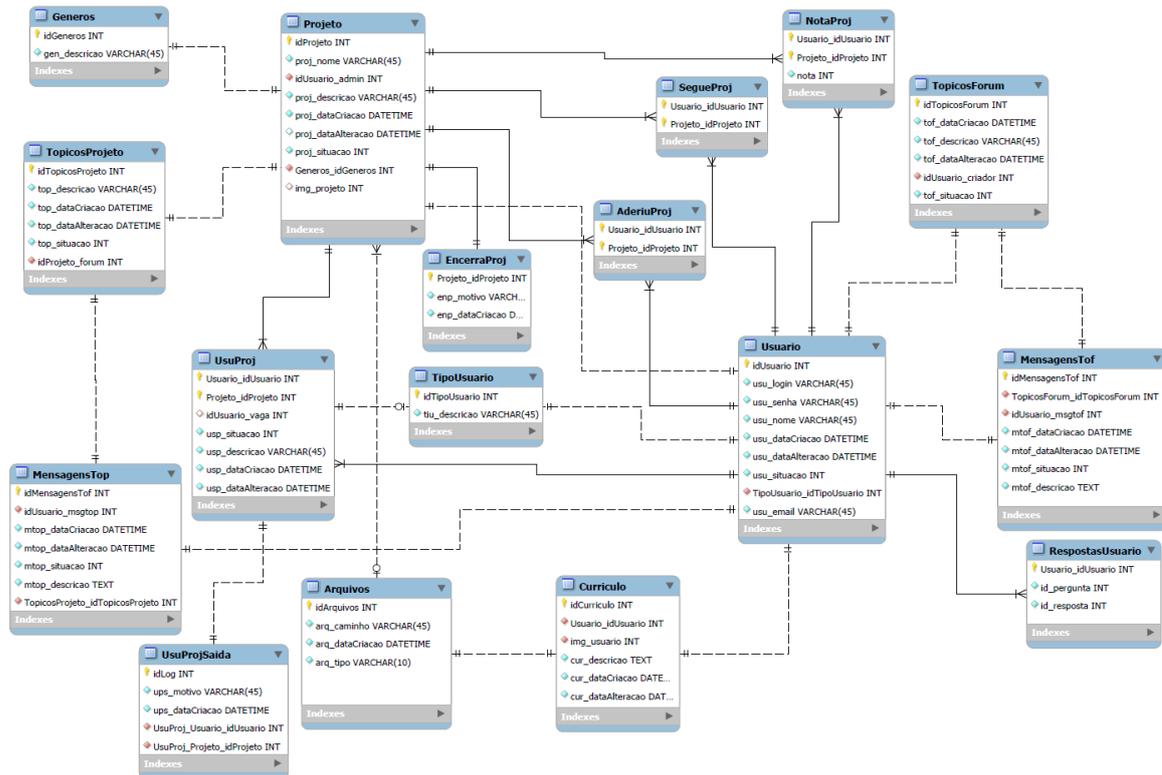


Figura 13: Modelo do banco de dados.

A figura 13 mostra a modelagem final do banco de dados. A única tabela que sofreu alterações desde sua modelagem foi a de *Usuario*, pois o *Django* já fornece um modelo de usuário. Assim, foi optado por descartar essa tabela de usuários e utilizar a nascente do *Django* com algumas modificações.

No desenvolvimento do sistema essa modelagem foi utilizada como base para gerar os modelos utilizados no projeto, que serão apresentados na seção 3.2.2.1.

Ainda antes de iniciar a programação do sistema, desenvolveu-se um protótipo de *design*. Ele contém todas as estruturas de telas existentes no sistema, sendo elas:

- Home;
- Projeto;
- Usuario;
- BuscaProjetos;
- BuscaUsuario;
- CadastroUsuario;
- CadastroProjeto;
- Forum;
- ForumMensagens;

Esse protótipo serviu de base para gerar o *html* das páginas do *website*, e também para orientar na criação dos eventos e métodos.

Nas figuras 14, 15 e 16 seguem esquetes respectivamente das páginas: Home, Projeto e Usuario; produzidas através da ferramenta *Balsamiq Mockups* apresentada na seção 3.1.1.3.

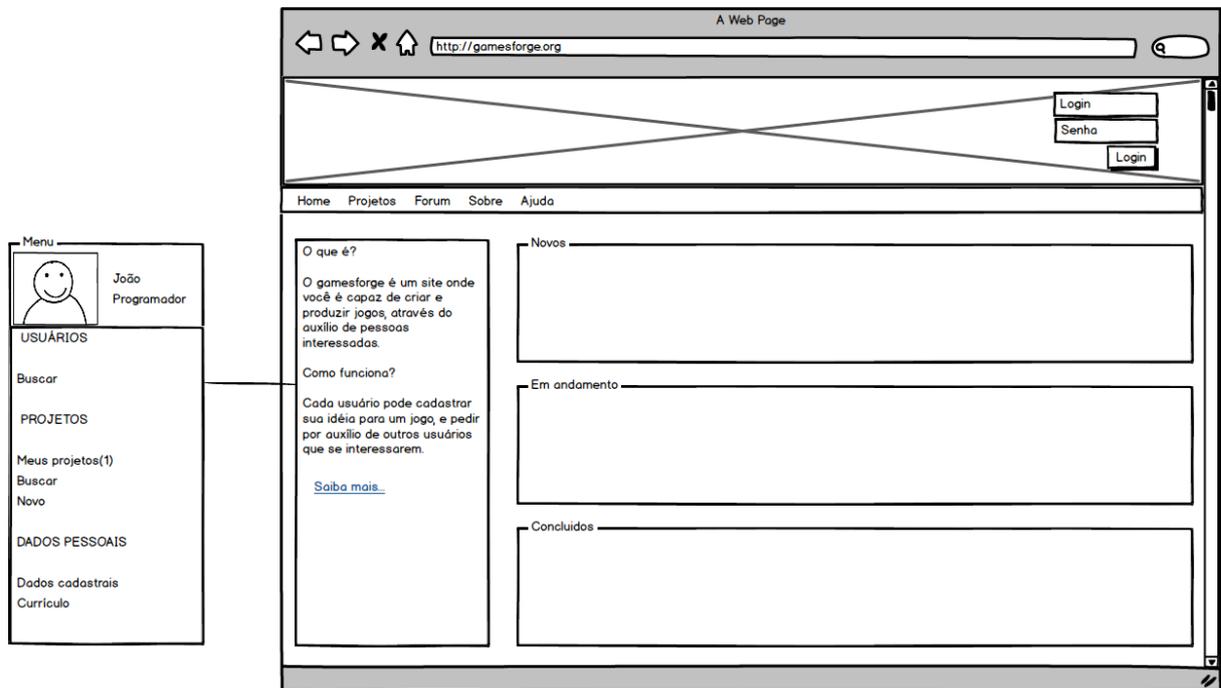


Figura 14: Página home do protótipo.

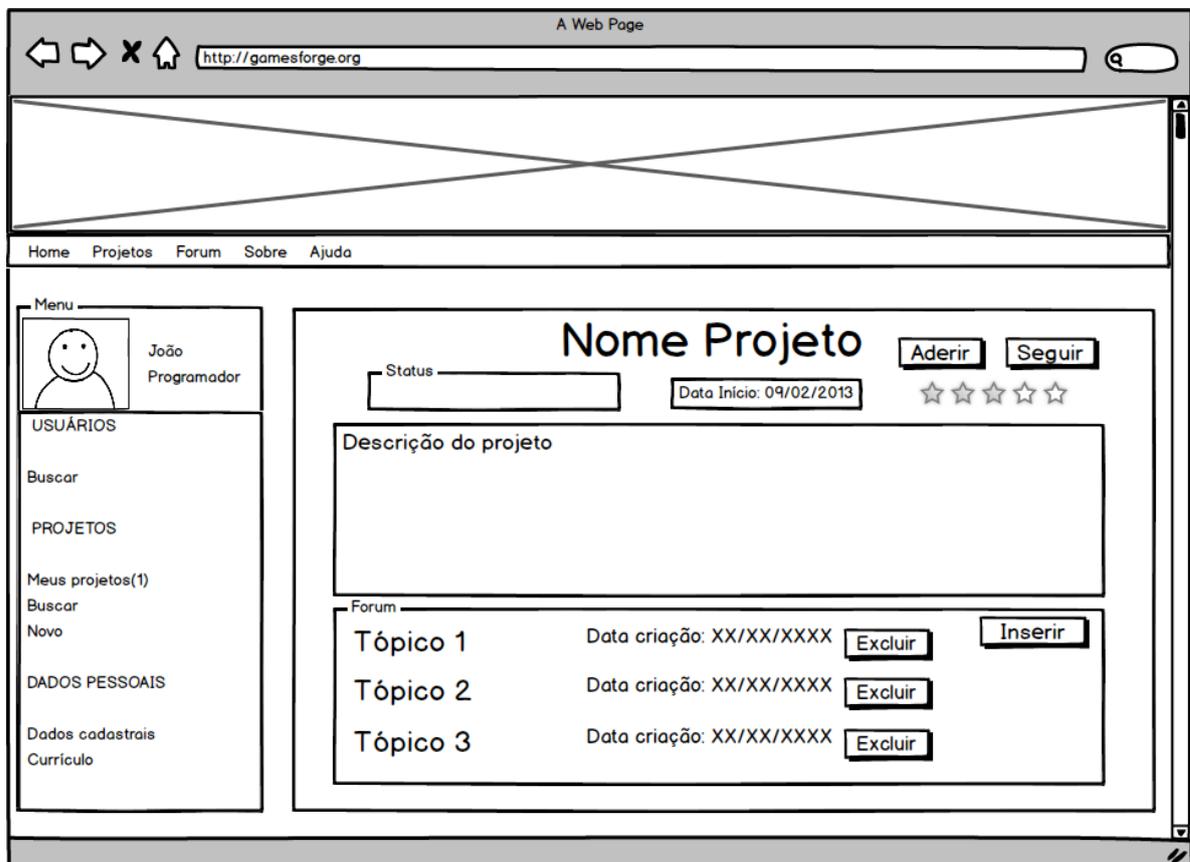


Figura 15: Página do projeto do protótipo.

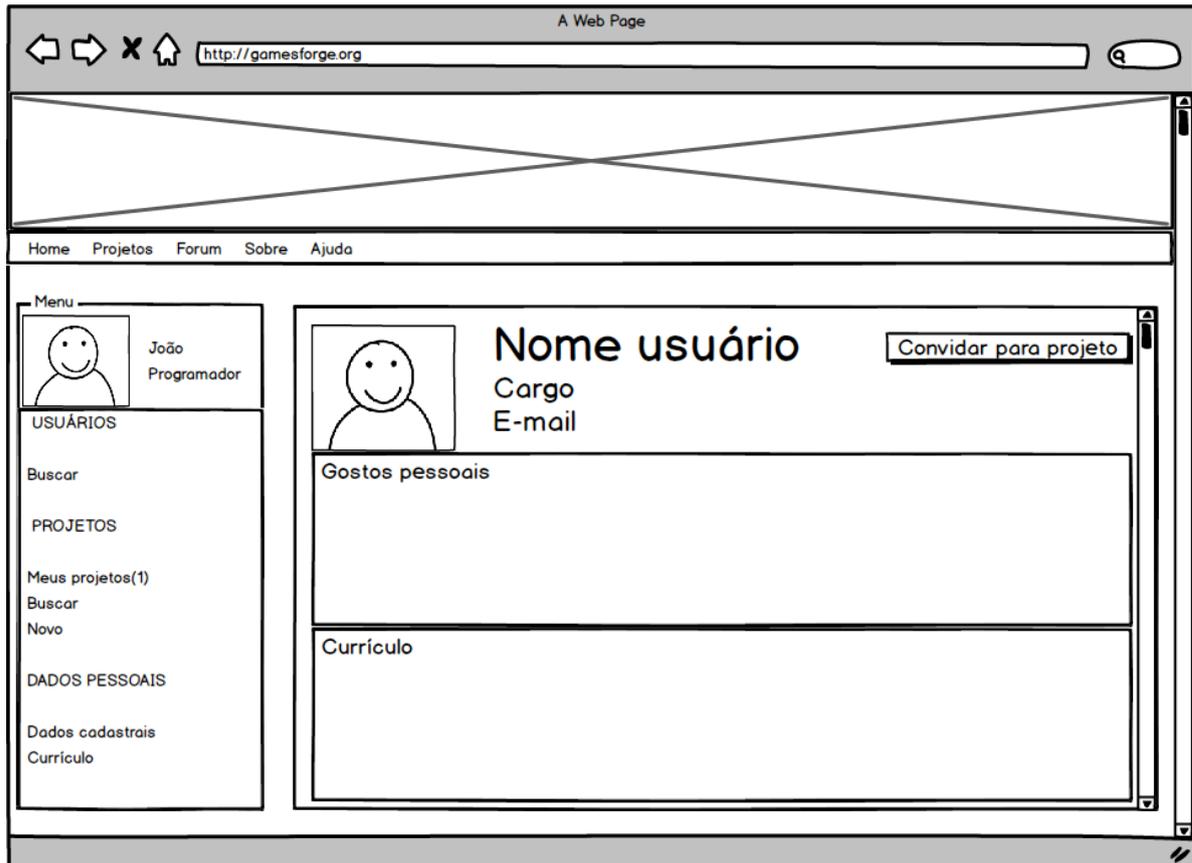


Figura 16: Página do usuário do protótipo.

3.2.2 Programação

Como apresentado na seção 3.1.1.3, o *Django* será o *framework* adotado para desenvolver a codificação do sistema. Esse *framework* segue o modelo *MVC*, para simplificar a compreensão dessa fase do projeto. Definiu-se que o texto será dividido em cinco etapas: modelos, visões, *templates*, controles e estilos.

Os modelos são a base do sistema e com eles foram definidas todas tabelas do banco de dados. A criação da base de dados é feita automaticamente pelo *Django*, desde que todos os modelos sigam a estrutura corretamente. Para um projeto funcionar nesse *framework* é necessário definir, primeiramente, os modelos.

Após criar os modelos, o *Django* não restringe a ordem de desenvolvimento do projeto. Portanto, para demonstrar melhor cada uma das páginas importantes do *website* serão intercaladas as partes de programação entre: *templates*, estilos, controles e visões.

3.2.2.1 Modelos

Um modelo é uma fonte única e definitiva de informação sobre os dados do sistema. Nele estão contidos os atributos e comportamentos de um determinado

dado armazenado e, geralmente, cada modelo é atrelado a uma única tabela no banco de dados.

No projeto foram criados 15 modelos de dados, os quais são apresentados a seguir:

- CustomUser;
- Curriculo;
- Genero;
- Arquivos;
- Projeto;
- UsuProj;
- NotaProj;
- SegueProj;
- AderiuProj;
- TopicosForum;
- MensagensForum;
- TopicosProjeto;
- MensagensProjeto;
- LogSaida;
- LogProjeto;

Cada um desses modelos será explicado posteriormente juntamente com os motivos para sua criação.

3.2.2.1.1 Modelos de Usuario

Dentro desses modelos estão contidos dois dos citados anteriormente: CustomUser e Curriculo.

Quando um projeto *Django* é criado, o próprio *framework* gera um modelo de usuário para o sistema. Assim, o desenvolvedor pode utilizar esse modelo para realizar o cadastro e autenticação dos usuários no banco.

Porém o modelo genérico não possuía todos atributos necessários para o projeto, portanto foram acrescentados novos campos no modelo usuário padrão.

As figuras 17 e 18 mostram as modificações aplicadas no modelo usuário padrão.

```
class CustomUser(AbstractBaseUser):
    email = models.EmailField(verbose_name='email address',max_length=255, unique=True,db_index=True,)
    tipouser = models.IntegerField()
    is_active = models.BooleanField(default=True)
    is_staff = models.BooleanField(default=False)
    is_admin = models.BooleanField(default=False)

    objects = CustomUserManager()

    USERNAME_FIELD = 'email'
    REQUIRED_FIELDS = ['tipouser']
```

Figura 17: Classe CustomUser criada para modificação do usuário padrão.

```

class CustomUserManager(BaseUserManager):
    def create_user(self, email, tipouser, password=None):
        if not email:
            raise ValueError('Usuarios precisam ter um email.')

        user = self.model(
            email=CustomUserManager.normalize_email(email),
            tipouser=tipouser,
        )

        user.set_password(password)
        user.save(using=self._db)
        return user

    def create_superuser(self, email, tipouser, password):
        user = self.create_user(email,
            password=password,
            tipouser=tipouser
        )
        user.is_admin = True
        user.is_staff = True
        user.save(using=self._db)
        return user

```

Figura 18: Classe de gerenciamento do CustomUser.

Junto ao novo modelo de usuário foi incluído um modelo de currículo. Esse modelo contém todos dados do currículo do usuário, entre eles: descrição, imagens, data de criação e data de alteração. Além de possuir o número de identificação do usuário atrelado a ele.

Esses dois modelos, CustomUser e Currículo, contém todos dados pessoais de cadastro do usuário, além de contar com dados para autenticação também.

3.2.2.1.2 Modelos de projeto

Para os projetos foram definidos dez modelos: Genero, Projeto, UsuProj, NotaProj, SegueProj, AderiuProj, TopicosProjeto, MensagensProjeto, LogProjeto e LogSaida.

Todo projeto cadastrado no sistema deve pertencer a um gênero e para isso o modelo Genero foi criado, sendo que, este modelo contém a maior parte dos gêneros de jogos existentes para um projeto.

Para guardar os dados referentes ao projeto, como, por exemplo, o nome, foi criado um modelo com todos os atributos relacionados ao projeto. Sendo que dois desses atributos são derivados de outros modelos, sendo eles: o número de identificação do usuário e o número de identificação do gênero.

O identificador do usuário é utilizado para saber quem foi o criador do projeto, e o identificador do gênero para saber em qual gênero o projeto se enquadra. Os identificadores podem ser observados na figura 19.

```
class Projeto(models.Model):
    prj_admin = models.ForeignKey(settings.AUTH_USER_MODEL)
    prj_genero = models.ForeignKey(Genero)
```

Figura 19: Parte da classe Projeto mostrando os dois identificadores.

Além dos identificadores, o modelo Projetos contém um atributo que armazena o atual estado do projeto. Esse atributo pode assumir três valores, sendo eles:

1. Em andamento – Nesse estado é possível inserir usuários nas vagas abertas, e projeto aparece nas buscas.
2. Concluído – Não é possível inserir usuários nas vagas, e projeto aparece nas buscas.
3. Encerrado – Não é possível inserir usuários nas vagas, o projeto não aparece nas buscas e só pode ser visualizado pelo criador ou por participantes (usuários que aderiram ao projeto).

Já que no momento de criação do projeto o criador deve também inserir as vagas em aberto, tornou-se necessário criar um modelo que identificasse essas vagas. O modelo criado para essa atribuição se chama Usuproj, e contém todas informações da vaga, inclusive o atual ocupante dela.

Para que fosse possível o controle de estado da vaga foi criado um atributo *status*, esse atributo é capaz de assumir 3 situações:

1. Aberta – É possível inserir um usuário nessa vaga.
2. Em processamento – Algum usuário foi convidado a preencher essa vaga, porém o convite ainda não foi aceito.
3. Preenchida - Não é possível inserir um usuário nessa vaga, porém o usuário que a preenche pode ser retirado para então a vaga voltar ao estado de Aberta.
4. Encerrada – Não é possível fazer qualquer tipo de alteração nessa vaga, o ocupante dessa vaga continua nela após a vaga ser encerrada.

Além do estado, a vaga também possui o identificador de usuário, assim como o identificador de projeto, para que seja possível controlar a vaga pelo sistema.

Os identificadores e o tipo da vaga podem ser observados na figura 20.

```
class Usuproj(models.Model):
    usuario = models.ManyToManyField(settings.AUTH_USER_MODEL, blank=True)
    projeto = models.ManyToManyField(Projeto)
    up_usutipo = models.IntegerField()
```

Figura 20: Parte da classe Usuproj mostrando identificadores e o tipo de usuário.

Outros três módulos relacionados ao projeto são: NotaProj, SegueProj e AderiuProj. Todos têm em comum a presença dos identificadores de usuário e de projeto entre os atributos.

No NotaProj o atributo de nota do projeto foi criado, assim armazenando a nota que um usuário avaliador deu a um projeto.

No SegueProj os únicos atributos existentes são os identificadores, assim um usuário atribuído a um projeto na tabela desse módulo está seguindo o projeto.

O AderiuProj é semelhante ao SegueProj, contendo os mesmos atributos, porém o sistema entende que os usuários inseridos nesse modelo desejam ser membros do projeto, ou seja, preencher uma vaga disponível.

Para o fórum do projeto foram criados dois módulos: TopicosProjeto e MensagensProjeto.

O TopicosProjeto tem como atributos os identificadores do usuário e projeto, a descrição do tópico e elementos de validação – data de criação e data de alteração-.

O módulo MensagensProjeto contém todas as mensagens de determinado tópico, assim ele possui como atributos os identificadores de tópico e usuário, a descrição da mensagem e os mesmos elementos de validação do TopicosProjetos.

Por último, foi necessária a criação de modelos que armazenassem o histórico de encerramento de projetos e de retirada de usuários das vagas. Ou seja, quando um usuário encerra um projeto, o motivo do encerramento, a data e o identificador do projeto fica armazenado na tabela de histórico. O mesmo vale para quando o usuário sai de um projeto. Os modelos criados para essas duas ocasiões são respectivamente os modelos: LogProjeto e LogSaida.

3.2.2.1.3 Modelos gerais

Os modelos restantes são: TopicosForum, MensagensForum e Arquivos.

Os modelos TopicosForum e MensagensForum foram ambos definidos para o fórum geral. Eles contêm o identificador de usuário para controle dos tópicos e mensagens criadas no fórum. Eles também possuem uma descrição, data de criação e data de alteração.

No modelo Arquivo, estão todas as informações das imagens existentes no sistema. Como seu tamanho, data de criação, e o principal, o caminho onde essa imagem está armazenada.

3.2.2.2 Páginas

As páginas criadas no sistema seguem o padrão do *Django*, para cada uma das páginas existe um *template*, um estilo, e uma ou várias *views*.

Os *templates* são arquivos que contêm toda a estrutura das páginas do *website*. Os elementos dessa estrutura são sinalizados por *tags html*, e alguns dos

métodos de tratamento de conteúdo da página são escritas no próprio *template* utilizando a linguagem *Python*. Os *templates* existentes no sistema são:

- home.html
- projeto.html
- usuario.html
- buscaprojeto.html
- buscausuario.html
- Cadastro de usuário:
 - cadastrousuario.html
 - cadastrocurriculo.html
- cadastroprojeto.html
- forum.html
- topicoforum.html
- meusprojetos.html
- sobre.html
- ajuda.html

Todo *template* tem um *controller* atrelado a ele, sendo que esse serve como mediador entre a requisição e a resposta. Ou seja, quando um usuário faz uma requisição na página o *controller* a trata e manda um conjunto de informações para a *view*. Ao receber essas informações a *view* decide como proceder e gera uma resposta. Essa resposta normalmente contém o que será mostrado ou tratado no *template*.

Para fazer essa requisição é necessário que a *url* da página seja inserida no arquivos *urls.py*. Assim ao digitar um endereço na barra de navegação do *browser* o sistema é capaz de tratar a entrada fornecida pelo usuário, e redireciona-lo para a página desejada.

O estilo das páginas é desenvolvido totalmente em *css* – linguagem de estilo utilizada para definir a apresentação de documentos escritos -, e todos os arquivos de estilo estão dentro da pasta do projeto.

3.2.2.2.1 Páginas do usuário

As páginas referentes ao conteúdo do usuário são quatro, sendo elas: *usuario.html*, *castrousuario.html*, *cadastrocurriculo.html* e *buscausuario.html*.

O cadastro de usuário foi dividido em duas partes: o cadastro de dados pessoais e o cadastro de currículo.

No cadastro de dados pessoais (Figura 21), o usuário deve inserir dados que correspondem ao perfil do usuário, entre eles:

- Nome
- Login
- Senha
- Email

- Área de atuação
- Gosto pessoal – perguntas de *feedback* para determinar gostos do usuário

Figura 21: Página de cadastro de usuário.

Para assegurar que o usuário não pressione o botão “Voltar” sem querer, e assim perca todos os dados escritos, ao pressionar o botão “Voltar” uma janela se abre e pergunta ao usuário se ele realmente deseja cancelar o cadastro.

Se o usuário aceitar o cancelamento, o sistema o redirecionará para a página inicial. Se o usuário recusar, o sistema manterá o usuário na página de cadastro.

Ao clicar no botão “continuar” o usuário avança no cadastro sendo redirecionado para a página de cadastro do currículo. Como pode ser visto na figura 22.

Figura 22: Página de cadastro do currículo.

O currículo não é obrigatório, assim se o usuário pressionar o botão “Continuar” com o currículo em branco, uma janela abrirá e pedirá para o usuário confirmar sua decisão. Lembrando-o que ao deixar o currículo em branco uma parte de seu perfil não aparecerá. Se o usuário aceitar, o currículo será salvo no estado que estiver. Se o usuário recusar, o sistema manterá o usuário na página de cadastro de currículo.

Quando o usuário desejar editar seu perfil ele deve acessar a página Usuarios pelo menu de navegação, e então pressionar o botão “Editar perfil”. Dessa forma o usuário será redirecionado para a primeira página de cadastro, já com todos seus dados carregados nos campos do formulário.

Ao terminar todo o cadastro, o usuário é redirecionado para de perfil dele, essa página contém todas informações inseridas pelo usuário na realização do cadastro.

Figura 23: Página de visualização do usuário.

Na figura 23 a frase abaixo do nome do usuário é gerada pelo sistema. Sendo que ela é preenchida com dados fornecidos pelo usuário no seu cadastro.

O botão de “convidar para projeto” aparecerá apenas aos outros usuários do sistema, e somente se esses outros usuários tiverem algum projeto criado. Ao ser convidado para um projeto, o usuário convidado recebe um e-mail com as informações do projeto e um link de redirecionamento para página do projeto. Se o usuário convidado desejar participar do projeto, ele deve clicar no botão aderir que será mostrado no decorrer do texto na seção de Páginas de projeto.

A última página de usuário criada foi a de busca. Essa página tem a função de listar os usuários de acordo com os filtros escolhidos. Entre os filtros existentes estão: “Nome”, “Número de projetos em que participou” e “Área de atuação”. A ordem que a busca mostra os resultados é:

- 1º Nomes por ordem alfabética;
- 2º Número de projetos em que participou por ordem decrescente.



Figura 24: Página de busca do usuário.

Na página representada na figura 24 o usuário pode acessar a edição de perfil, como descrito anteriormente, ao pressionar o botão “Editar perfil”.

3.2.2.2 Páginas de projeto

Existem quatro páginas vinculadas aos projetos: projeto.html, buscaprojeto.html, cadastroprojeto.html e meusprojetos.html.

O cadastro de projeto (Figuras 25 e 26) foi dividido em duas etapas. Ambas fases estão contidas na mesma página, porém são mostradas ao usuário de maneira sequencial. Inicialmente o usuário insere dados de identificação do projeto, entre eles:

- Título – nome do projeto definido pelo criador;
- Ícone – imagem exibida na página do projeto, e também pode aparecer na página inicial do sistema;
- Status – esse campo só aparece se o usuário estiver na opção de editar o projeto, pois ao ser criado, é atribuído o status ‘Em Andamento’ automaticamente;
- Gênero(s) – gênero no qual o projeto se enquadra;
- Descrição – texto com o intuito de apresentar as características do projeto.

Figura 25: Primeira etapa do cadastro de projeto.

Na segunda etapa o usuário descreve todas as vagas que estarão disponíveis no projeto. Inicialmente deve-se definir o número de vagas existentes, a cada incremento no campo “Número de vagas”, mostrado na figura 26, um novo formulário de descrição de vaga será aberto. Nesse formulário haverá um campo para escolher o cargo e um campo para descrevê-lo.

Figura 26: Segunda etapa do projeto.

Para preencher uma vaga, o criador deve pressionar o botão “Convidar”, mostrado na figura 26. Após isso, uma janela se abrirá e dentro dela aparecerá uma busca de usuários. O resultado da busca conterá todos os usuários que se

enquadrem nos filtros e que tenham aderido ao projeto. Essa janela pode ser vista na figura 27.

Se o criador deseja convidar alguém para o projeto basta pressionar o botão “Convidar”, que a janela se fechará e o nome do usuário aparecerá na vaga, fazendo com que ela assuma o estado “Em processamento”.

Um convite será enviado para o usuário. Se ele recusar o convite, a vaga voltará ao estado “Aberta” e o nome do usuário convidado desaparece da vaga. Se o convidado aceitar, a vaga assume o estado “Preenchida” e o nome do usuário se mantém.

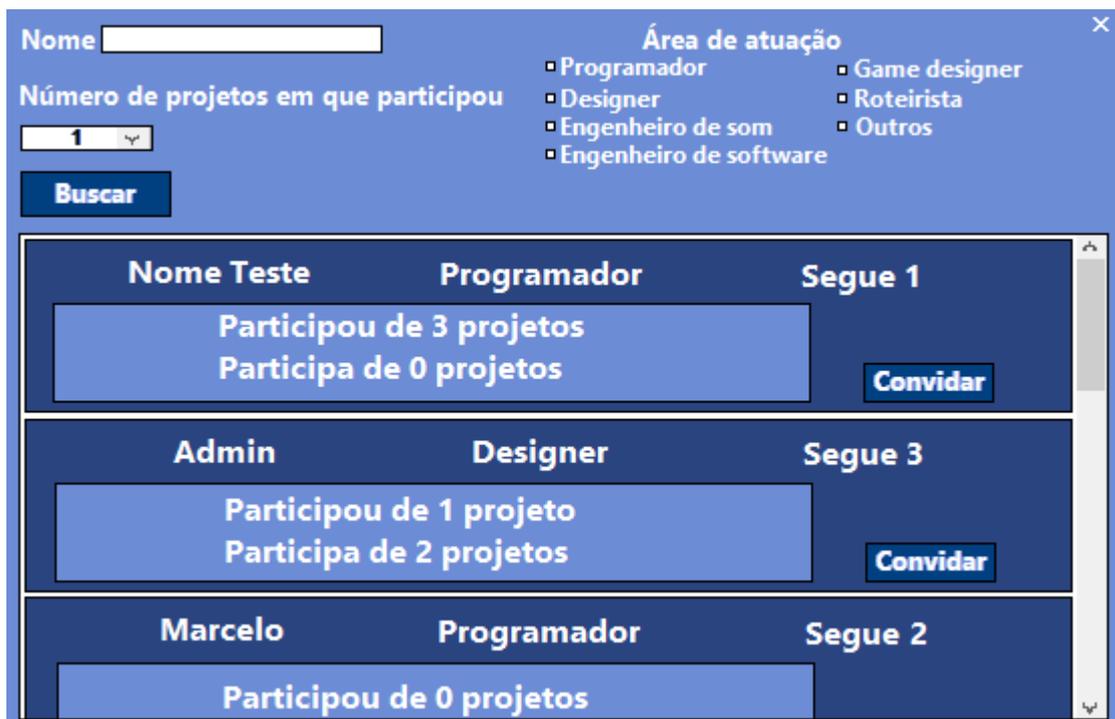


Figura 27: Janela de preenchimento da vaga.

Ao terminar a criação do projeto o usuário é redirecionado para a página do projeto (Figura 28), onde poderão ser observadas todas as informações cadastradas. Além de contar com o fórum do projeto, os botões “Aderir” e “Seguir”, e a nota do projeto.



Figura 28: Página do projeto.

Os botões “Aderir” e “Seguir” não aparecem para o criador do projeto. Se um usuário pressionar os botões de aderir ou seguir o projeto, o botão pressionado desaparece da página do projeto para esse usuário. A diferença entre os dois atos sugeridos por esses botões são:

- “Aderir” – o usuário se dispõe a preencher uma das vagas disponíveis no projeto;
- “Seguir” – o usuário recebe as atualizações da página do projeto, porém não pode preencher as vagas do projeto.

Quando todas as vagas do projeto forem preenchidas, o botão “Aderir” deixa de aparecer na página do projeto, e só aparecerá novamente se uma das vagas voltar ao estado “Aberta”.

Para avaliar o projeto o usuário deve pressionar na nota que ele deseja atribuir ao projeto. A escala das notas varia de um a cinco e é identificada por estrelas que “acendem” quando o usuário passa o mouse sobre elas.

Ao clicar no *link* “Mais tópicos” o usuário é redirecionado para o fórum do projeto. Essa página é idêntica ao fórum geral, porém, contém os tópicos referentes ao projeto. Uma descrição mais detalhada do fórum será apresentada na seção 3.2.2.2.3.

Para o usuário visualizar os projetos que está participando, ele precisa entrar na página meusprojetos.html. Nessa página estão todos os projetos que ele participa ou já participou, e é nela que acontece a maior parte das regras de negócios do sistema. Também nessa página o usuário pode cadastrar um novo projeto pressionando o botão “Novo projeto”, e então sendo redirecionado para a página de cadastro de projeto.

A página meusprojetos.html pode ser vista na figura 29.



Figura 29: Página dos projetos do usuário.

Cada formulário possui um resumo de um projeto do usuário. Além do resumo, nesses formulários aparecem três botões. Um deles é o botão “Abrir”, que ao ser pressionado redireciona o usuário para a página do projeto. Outro botão é o “Editar”, este aparece somente para o criador do projeto, e ao pressionado, redireciona-o para o cadastro de projeto, porém com os campos preenchidos com os dados do projeto. O último botão é capaz de assumir diversos valores dependendo da ligação do usuário com o projeto.

Os valores que esse segundo botão pode assumir são:

- “Aceitar” – se o usuário for convidado para o projeto;
- “Encerrar” – se o usuário for o criador do projeto;
- “Sair” – se ele estiver participando do projeto e não é seu criador.

Ao encerrar ou sair do projeto uma janela se abrirá questionando o motivo do usuário estar encerrando ou saindo do projeto, ao responder o usuário pode prosseguir. Essa informação ficará salva no histórico de saída ou encerramento do projeto.

A última página de projetos é a de busca (Figura 30). Essa página funciona de maneira semelhante a página de busca de usuário, porém com algumas modificações nos filtros. Os filtros da busca de projetos são: “Título”, “Número de seguidores”, “Número de participantes”, “Nota” e “Status”.

A exibição do resultado segue a ordem de prioridade a seguir: ordem alfabética do título, *status* (ordem de “Em andamento” depois “Concluído”), número de seguidores (ordem decrescente), nota (ordem decrescente), número de seguidores (ordem decrescente), sendo o primeiro o com a maior prioridade e o último com a menor.

Os formulários que aparecem como resultado da busca contém um resumo e um botão “Abrir” que redireciona para a página do projeto.



Figura 30: Página de busca dos projetos.

3.2.2.2.3 Páginas gerais

As páginas restantes são: forum.html, home.html, sobre.html e ajuda.html.

A página inicial do *website* é a home.html (Figura 31). Nessa página existem 3 divisões, sendo que cada uma mostra cinco projetos. As divisões são: os cinco projetos mais recentes do sistema, os cinco projetos em andamento com maior nota e os cinco projetos concluídos com maior nota.

Ao clicar sobre um dos projetos dentro dessas divisões o usuário é redirecionado para a página do projeto (Figura 28).

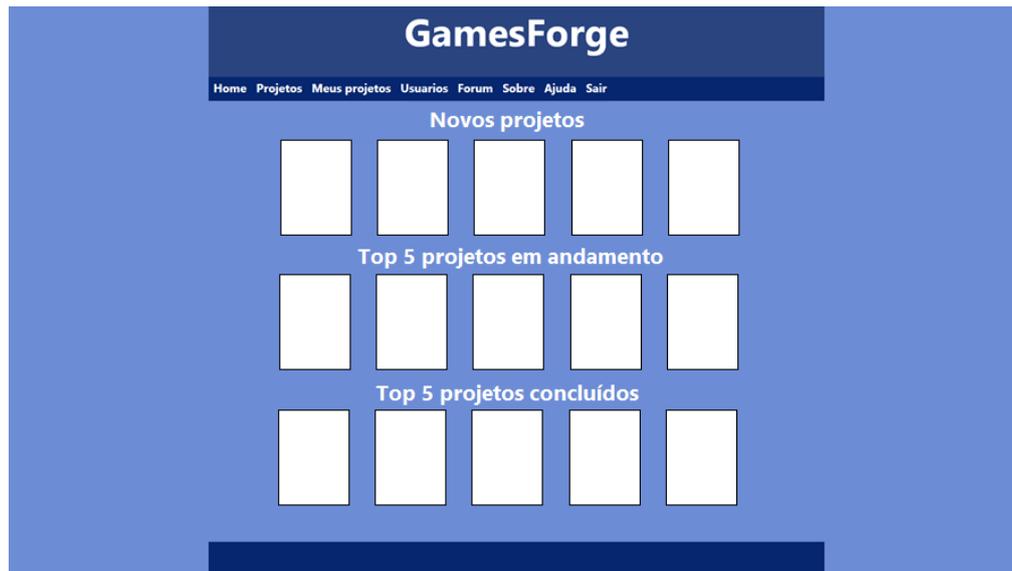


Figura 31: Página inicial.

O fórum do projeto e geral são exibidos na página `forum.html` (Figura 32). Dependendo da requisição do usuário, ele é redirecionado para o fórum de um projeto ou para o fórum geral. No fórum existem diversos *links* que abrem tópicos cadastrados.



Figura 32: Página de tópicos do fórum geral.

Para criar um tópico o usuário deve pressionar o botão “Novo tópico”, que abrirá uma janela com os campos para o título e descrição do tópico. Ao finalizar a criação do tópico um novo *link* será criado no fórum, e esse *link* ao ser pressionado redireciona o usuário para o tópico criado.

A página de visualização do tópico pode ser vista na figura 33.



Figura 33: Página do tópico.

Na página do tópico estão todas mensagens cadastradas nele. A ordem de exibição das mensagens segue a data de criação das mesmas. Ou seja, as mensagens novas vão ficar posicionadas abaixo das mais antigas. Toda mensagem tem como informação: seu criador, a data de criação, o número de mensagens do usuário e a descrição.

Para cadastrar uma nova mensagem no tópico basta o usuário pressionar o botão “Nova mensagem”. Uma janela se abrirá, semelhante a janela de criação de tópico, ao preencher os campos dessa janela a mensagem será criada.

As últimas páginas do *website* são: ajuda.html e sobre.html. A página de ajuda foi criada para auxiliar o usuário na navegação do *website*. Ela apresenta algumas características e funcionalidades do sistema ao usuário, com o intuito de auxiliar na utilização do website, além de sanar as principais dúvidas relativas as características da ferramenta desenvolvida. Já a página sobre.html apresenta todos os desenvolvedores e colaboradores na construção do sistema.

Ambas as páginas tem conteúdo estático, ou seja, podem ser modificadas apenas através do código-fonte.

3.2.3 Testes

Foram realizados testes de unidade, integração, funcional e interface. Durante o desenvolvimento do projeto os testes de unidade e integração foram aplicados, pois ambos buscam sanar os problemas relacionados ao momento de produção, assim verificando problemas diretamente no código.

Já os testes de interface e funcional, foram realizados após todo o desenvolvimento, utilizando técnicas de teste do tipo caixa-preta¹³. A cada erro encontrado durante esses testes, voltava-se para etapa de desenvolvimento, corrigindo o erro e voltando a desenvolver as atividades relativas aos testes.

Com relação à quantidade de testes realizados, decidiu-se a partir do modelo proposto por Bastos et al. (2007), apresentado na Figura 34.

Esse modelo sugere que, confrontando-se o número de testes realizados com a quantidade de defeitos encontrados, existe um ponto em que o custo para encontrar um erro não é mais lucrativo, portanto esse é o ponto crítico no qual os testes passam a ter uma relação de custo/benefício negativa.

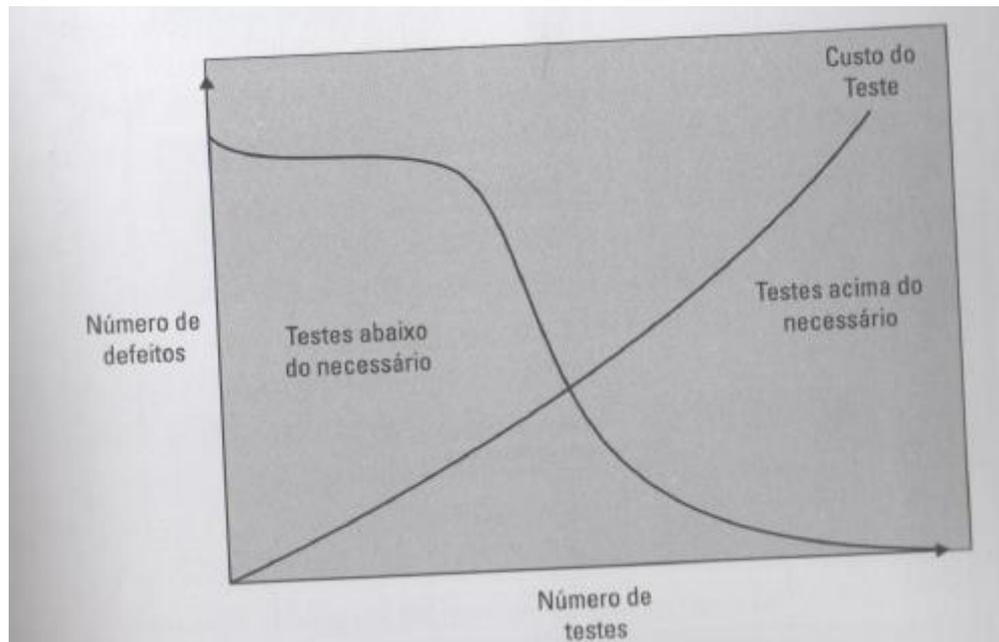


Figura 34: Gráfico de exaustão de testes.

Fonte: Bastos et al., 2007, p. 25.

O sistema foi aprovado em todos os testes aos quais foi submetido, não sendo encontrados novos erros até a etapa de homologação e documentação.

3.3 Pós-projeto

Após a etapa de desenvolvimento, viu-se a necessidade de gerar um documento relativo a todas etapas do projeto. Nesse documento foram apresentadas todas as definições e métodos de desenvolvimento utilizados.

Além desse documento, foi criado um diagrama de estado, onde estão descritas todas as ações do usuário com relação aos projetos. Esse diagrama tem como função facilitar a manutenção futura do sistema. Para gera-lo foi utilizado a ferramenta *StarUML*, apresentada na seção 3.1.1.3.

¹³ Os testes de caixa-preta analisam o comportamento de um objeto ignorando totalmente sua construção interna

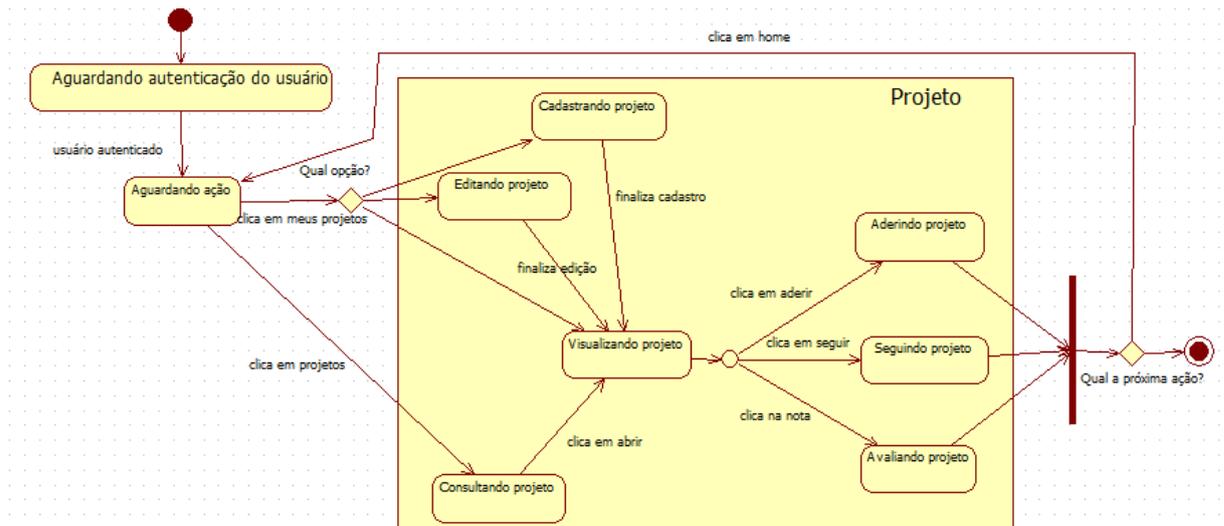


Figura 35: Diagrama de estado das ações dos usuários com os projetos.

Capítulo 4

4 Conclusões

4.1 Problemas e soluções

Devido ao conhecimento reduzido dos desenvolvedores sobre a parte de *design*, foram encontrados empecilhos na hora de produzir os protótipos de *layouts* na aplicação. Então apesar de ser completamente funcional, de acordo com os objetivos propostos, o sistema apresenta algumas falhas de usabilidade.

Para que esse problema seja solucionado, propõem-se que o projeto continue com um designer capaz de implementar os estilos às páginas, e adequá-las ao sistema.

Outros problemas encontrados durante o desenvolvimento, foram algumas restrições do *framework*. Apesar de agilizar o desenvolvimento em diversas situações, ele deixou a desejar quando se tratava do seu uso em conjunto com a linguagem *JavaScript*, e de suas bibliotecas derivadas.

Para resolver esse problema foi necessário reduzir o uso do *JavaScript* e utilizar outras tecnologias que fossem compatíveis com o *framework*. Pode-se listar como tecnologias retiradas, o *CKEditor* – editor de texto *html* – que estava em conflito com os formulários nativos do *Django*; e a biblioteca *JavaScript/CSS Bootstrap* que também estava conflitando com componentes do *Django*. Em contrapartida, foram adicionados elementos *html* e *JavaScript* equivalentes com menos recursos mas que cumpriam a mesma função satisfatoriamente.

4.2 Trabalhos futuros

Para prosseguir com o projeto, as próximas etapas seriam: a melhoria no design do site e a adequação do sistema para ser rodado em um servidor. O sistema foi completamente desenvolvido e testado localmente, assim para que funcione em um servidor são necessárias pequenas modificações.

Após efetuadas essas modificações torna-se possível o lançamento do sistema na *web*. Assim, as pessoas que acessam a internet poderão usar o sistema, e possibilitar uma análise de opinião, que pode acarretar em possíveis melhorias.

4.3 Considerações finais

O objetivo geral do projeto foi alcançado. Ao final do desenvolvimento foi entregue uma ferramenta capaz de facilitar o meio de comunicação entre os interessados na criação de jogos. Além de ampliar a gama de possibilidades para as pessoas que não possuem conhecimento técnico ou tempo disponível para desenvolver um jogo.

Ao final, o sistema foi testado em *host* local e funciona conforme especificado, portanto, se pequenos ajustes forem realizados, o *site* estará apto a rodar *online*.

Referências bibliográficas

BASTOS, A. et al. **Base de conhecimento em testes de software**. [S.l.]: Martins, 2007.

EXAME. Daniele, A.. **65 mil jovens serão formados em TI esse ano, diz estudo**. São Paulo, jul. 2012. Disponível em <<http://exame.abril.com.br/carreira/noticias/65-mil-jovens-serao-formados-em-ti-esse-ano-diz-estudo>>. Acesso em 7 mar.2013.

KICKSTARTER. **Best of Kickstarter 2012**. 2012. Disponível em <<http://www.kickstarter.com/year/2012>>. Acesso em 21 mar. 2013.

NIELSEN. **Top 10 Global Web Parent Companies**. Nova York, 2013. Disponível em <<http://www.nielsen.com/us/en/top10s.html>>. Acesso em 21 mar. 2013.

PRESSMAN, R. **Engenharia de software**. 7. ed. [S.l.]: MCGRAW HILL - ARTMED, 2011.

SAMPAIO, H.. **“Só produzimos menos jogos que os EUA”, diz organizador da GGJ**. Arena IG, fev. 2012. Disponível em <<http://arena.ig.com.br/so-produzimos-menos-jogos-que-os-eua-diz-organizador-da-ggj/n1597613164303.html>>. Acesso em 8 mar. 2013.

STARUML. **StarUML**, 2012. Disponível em: <<http://crocusmodeller.com/>>. Acesso em 21 out. 2013.

SULLIVAN, M.. **Crowdfunding**. 2013. Disponível em <<http://crowdfunding.pbworks.com/w/page/10402176/Crowdfunding>>. Acesso em 8 mar. 2013.

Bibliografia complementar

_____. NBR 10520: informação e documentação – citações em documentos - apresentação. Rio de Janeiro: ABNT, 2002.

Afonso, Arlei. **Conhecendo os Indie Games**. Gamesindiebr. Disponível em <<http://gamesindiebr.blogspot.com.br/p/historia.html>>. Acesso em 20 abr. 2013.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6023: informação e documentação - referências - elaboração. Rio de Janeiro: ABNT, 2002.

Autor desconhecido. **The History of Crowdfunding**. Fundable, 2012. Disponível em <<http://www.fundable.com/crowdfunding101/history-of-crowdfunding>>. Acesso em 20 abr. 2013.

Cox, Patrick. **Top 10 reasons to use HTML5 right now**. 2011. Disponível em <<http://tympanus.net/codrops/2011/11/24/top-10-reasons-to-use-html5-right-now/>>. Acesso 14 abr. 2013.

Daquino, Fernando. **A história das redes sociais: como tudo começou**. Tecmundo, nov. 2012. Disponível em <<http://www.tecmundo.com.br/redes-sociais/33036-a-historia-das-redes-sociais-como-tudo-comecou.htm>>. Acesso em 13 abr. 2013.

Holovaty, A.; Kaplan-Moss, J. et al. **The Django Book**. 2002. Disponível em <<http://www.djangobook.com/en/2.0/index.html>>. Acesso em 13 abr. 2013.

jQuery. **jQuery API**. 2013. Disponível em <<http://api.jquery.com/>>. Acesso em 14 abr. 2013.

Mozilla. **About JavaScript**. 2012. Disponível em <https://developer.mozilla.org/en-US/docs/JavaScript/About_JavaScript>. Acesso em 15 abr. 2013.

Mozilla. **CSS**. 2013. Disponível em <<https://developer.mozilla.org/en-US/docs/Web/CSS?redirectlocale=en-US&redirectslug=CSS>>. Acesso em 13 abr. 2013.

MySQL. **MySQL 5.7 Reference Manual**. 2013. Disponível em <<http://dev.mysql.com/doc/refman/5.7/en/>>. Acesso em 14 abr. 2013.

PRESSMAN, R. **Engenharia de software**. [S.l.]: MCGRAW HILL - ARTMED, 2006.

Reichert, Kate. **Top 5 Problems Faced By Indie Game Developers**. Gamasutra, jan. 2012. Disponível em <http://www.gamasutra.com/blogs/KateReichert/20121101/180767/Top_5_Problems_Faced_By_Indie_Game_Developers.php>. Acesso em 7 mar. 2013.