Relatório Final do Projeto MatchDog

Disciplina: Engenharia de Software

Professor: Vinicius Cardoso Garcia

Alunos:

Artur de Carvalho Alves

Plinio Antunes Garcia

Tiago Heineck

JULHO DE 2015

1 Identificação

Projeto: MatchDog

Área do Projeto: Engenharia de Software

2 Introdução

O surgimento do MatchDog está ligado à necessidade dos amantes de cachorros em achar cães da mesma raça para "cruzar", mas muito além disso, essas pessoas possuem ligações afetivas com seus animais de estimação e querem vê-lo feliz. Dessa forma, mesmo sem ter a intenção de procriação, muitos procuram outros donos de cães para passear e conversar.

Segundo dados de 2013 do IBGE, no Brasil a cada 10 casas, 4 possuem cachorros, sendo uma população de animais estimada em mais de 52 milhões (IBGE, 2013).

Um ponto importante do MatchDog é que pode facilitar o encontro não somente para os cachorros, mas também de seus donos, que podem conhecer alguém interessante utilizando o sistema. Desta forma, o MatchDog pode facilitar a aproximação das pessoas, visto que, os próprios cachorros já são uma afinidade entre os donos.

Para atender a esse nicho de mercado o sistema propõe um organizador de encontros, por meio de um ambiente facilitado e intuitivo para que sem muitas dificuldades possam realizar um cadastro e montar um álbum. Possibilitando o gerenciamento de quantos perfis de cachorros o dono desejar.

Além do cadastro é possível fazer buscas por raça, sexo do dono, sexo do cachorro e filtrar os interesses, sendo que ao ocorrer interesse mútuo - chamado de "match" - entre os dois usuários, o sistema habilita um serviço de chat para que possam se conhecer melhor.

O projeto MatchDog espera possibilitar alegria aos donos de cachorros e aos seus respectivos animais de estimação, motivando as pessoas a se relacionar em uma experiência divertida incluindo seus respectivos animais de estimação.

3 Justificativa

O mercado ao qual o MatchDog se propõe ocupar possui alguns sistemas similares, na tabela 1, apresentamos uma comparação com possíveis concorrentes.

O comparativo mostra que as principais funcionalidades do MatchDog estão presentes em poucos concorrentes, como a opção de cadastrar vários cachorros em um perfil de usuário, escolher passear ou cruzar e abrir chat quando houver interesse mútuo entre os proprietários de cães. A funcionalidade que permite visualizar a foto do dono do cachorro é de exclusividade do MatchDog.

PLANILHA DE RASTREABILIDADE - MAPA DE PRODUTOS					
Principais Requisitos	DogsApp	DogDate	CruzaPet	MyDoggy	**MatchDog**
Cadastrar vários cachorros em um perfil de usuário		X	X		X
Escolher passear ou cruzar		X			X
Permitir ver a foto do dono do cachorro					X
Abrir chat quando houver interesse de ambos os lados	X				X
Busca baseada em cidades	X	X	X	X	X
Login via Facebook	X	X		X	X
Versão site			X		X
Versão aplicativo mobile	X	X		X	

Tabela 1- Comparativo do MatchDog com possíveis concorrentes

4 Processo de Desenvolvimento

Antes de iniciar qualquer tarefa relacionada à análise ou desenvolvimento do sistema, foi decidido em uma reunião inicial, com a participação de toda a equipe, o processo de desenvolvimento do software. Onde basicamente, definimos Atividades e Artefatos.

4.1 Definição das Atividades

As quatro atividades definidas tomam como referência o modelo do Processo Unificado (IBM, 2003). E, basicamente representam as atividades principais das fases do

Processo Unificado, sendo elas: Elicitação de Requisitos, Análise de Requisitos, Implementação e Testes, e Implantação e Treinamento.

4.2 Definição dos Artefatos

Para cada uma das quatro atividades propostas foram definidos quais artefatos seriam gerados:

4.2.1 Atividade de Elicitação de Requisitos

A proposta dessa atividade é levantar todos os requisitos do sistema, Para isso, como não temos um cliente definido, tomamos como ponto de partida um *brainstorming* feito entre os membros da equipe, dando origem aos dois primeiros artefatos do projeto:

- Desenho: um diagrama visual, que contém esboço do fluxo principal do sistema e wireframes de algumas funcionalidades.
- Brainstorm: uma lista com as principais ideias citadas na reunião de *brainstorming*.

Durante a Elicitação de Requisitos, com base no *brainstorming*, e na troca de ideias com outros membros da equipe, foram gerados os seguintes artefatos

- User Story Cards: cartões que descrevem sucintamente qual a expectativa do usuário. Na frente desse cartão o analista basicamente responde a três perguntas (Quem quer? O que quer? Para quê quer?) e no verso do cartão lista sucintamente alguns casos de teste.
- Atributos de Qualidade; Descrição dos atributos de qualidade que foram definidos como importantes para o Projeto. Assim como a motivação que levou a definí-lo como importante.
- Planilha de Rastreabilidade (Mapa de Produtos): Planilha que lista as principais funcionalidades do sistema, e informa quais delas já são implementadas pelos atuais projetos já existentes no mercado.

4.2.2 Atividade de Análise de Requisitos

Nessa atividade é feito todo o detalhamento necessário para o entendimento dos requisitos e implementação das funcionalidades. Essa atividade usa como artefatos de entrada aqueles que foram produzidos na etapa anterior, e como saída gera os seguintes:

- Lista de Requisitos: Planilha associando cada história de usuário aos respectivos requisitos funcionais identificados.
- Protótipos: Esboço das telas principais do sistema.
- Diagramas: foi definido que nessa atividade deve ser desenvolvido um diagrama de classes, e diagramas de atividades para os requisitos mais complexos.
- Issues (User Story Card detalhada): As Issues são as tarefas que serão implementadas, contendo referência para o User Story Card, assim como para seus detalhamentos, que são os protótipos e diagramas.

4.2.3 Atividade de Implementação e Testes

Essa é a atividade mais simples do processo de desenvolvimento, apenas um artefato de entrada, que são as Issues, e um artefato de saída, que é o código-fonte do projeto.

4.2.4 Atividade de Implantação

Com o código-fonte “em mãos”, nessa atividade o analista se propõe a implantar o sistema no servidor de produção. A cada nova versão implantada deve ser produzido o Release Notes, que basicamente devem conter o que há de novidade na nova versão implantada. Além disso, nessa atividade são produzidos e/ou modificados o manual de implantação e manual do usuário.

5 Gerência de Configuração e Ambiente

As ferramentas e tecnologias utilizadas foram escolhidas cuidadosamente para propiciar um processo simples e ágil, priorizando sempre o foco no trabalho colaborativo e distríbuido.

Em todas as fases do projeto, quando necessário criar ou editar arquivos de textos, elaborar planilhas e desenhos simples, fez-se o uso das ferramentas Documentos, Planilhas e Desenhos do Google, respectivamente. Da mesma forma todas as reuniões foram realizadas por conferências web via Hangout do Google. O aplicativo Whatsapp foi utilizado para marcar reuniões.

A wiki do GitHub foi utilizada para organizar os artefatos e informações a respeito do projeto, também fazendo uso das Issues para gestão das tarefas e do sistema de versionamento (Git) para controlar o projeto.

Como ambiente de codificação, foi utilizado o serviço Cloud9 por permitir o desenvolvimento colaborativo, possuir um servidor de Ruby de fácil configuração, ter integração com GitHub, portanto, traz os benefícios de um IaaS (Infrastructure as a Service).

Os demais recursos estão descritos a seguir conforme a atividade do processo de desenvolvimento definido anteriormente. Cada atividade corresponde a um *milestone* no GitHub.

5.1 Arquitetura

O modelo arquitetural utilizado neste projeto seguiu o modelo proposto pelo Rails, com responsabilidades bem definidas distribuídas entre o Modelo, Visão e Controle. Para auxiliar as visões e layout foram criadas algumas helpers. Também foi utilizado o princípio de convenção sobre configuração do próprio framework.

Por opção, não foram utilizados geradores de código *scaffolding*, também foram utilizados poucos geradores de *tags* de HTML nas visões, desta forma o responsável por implementar as telas não precisou se preocupar com *taglibs* do Rails.

5.2 Elicitação de Requisitos

- User Stories: artefato de desenvolvimento comum em projetos Scrum e Extreme Programming, que possui alto nível de definição de um requisito (AGILE MODELING, 2015). Para anexá-los à documentação foram utilizadas páginas na wiki do GitHub seguindo um modelo definido pela equipe. https://github.com/plinio/matchDog/wiki/Template-Story-Card

5.3 Análise de Requisitos

- Pencil para prototipação de telas. http://pencil.evolus.vn/
- Astah para diagramas de classes e atividades. http://astah.net/editions/community
- Adobe Illustrator para criação de logomarcas e imagens http://www.adobe.com/products/illustrator.html

5.4 Implementação e Testes

- Ruby como linguagem de programação. https://www.ruby-lang.org/pt/
- Ruby On Rails 4.2 como framework de programação ágil. http://rubyonrails.org/
- Omniauth com omniauth-facebook para prover autenticação. https://github.com/intridea/omniauth
- SQLite como banco de dados de desenvolvimento. https://www.sqlite.org/
- Bootstrap 3 para estilização de layout. http://getbootstrap.com/
- Cloud9 como ambiente de desenvolvimento. http://c9.io/
- GitHub para armazenar e gerenciar código-fonte. http://github.com/
- HTML, CSS e Javascript para *front-end* da aplicação

5.5 Implantação

- Heroku / para rodar o ambiente de produção do protótipo. http://heroku.com
- PostgreSQL para banco de dados de produção. http://www.postgresql.org/

6 Requisitos

O processo de elicitação de requisitos começou com um *brainstorming* com o objetivo de verificar quais seriam as ideias em torno do sistema, em seguida foi utilizado o recurso de Histórias de Usuários com elaboração de *User Story Cards*. Esta atividade permitiu gerar os artefatos dos cartões e também a identificação dos atributos de qualidade do sistema.

Outros artefatos gerados na elicitação de requisitos foram um pequeno mapa mental e *wireframes* que podem ser vistos na Figura 1. Tal artefato foi gerado durante o *brainstorm* por um analista observador que tratou de registrar as ideias.

Estes artefatos atuaram como norteadores para a extração dos requisitos funcionais do sistema que foram catalogados em uma planilha relacionando o cartão que originou ou que se relaciona com o requisito.

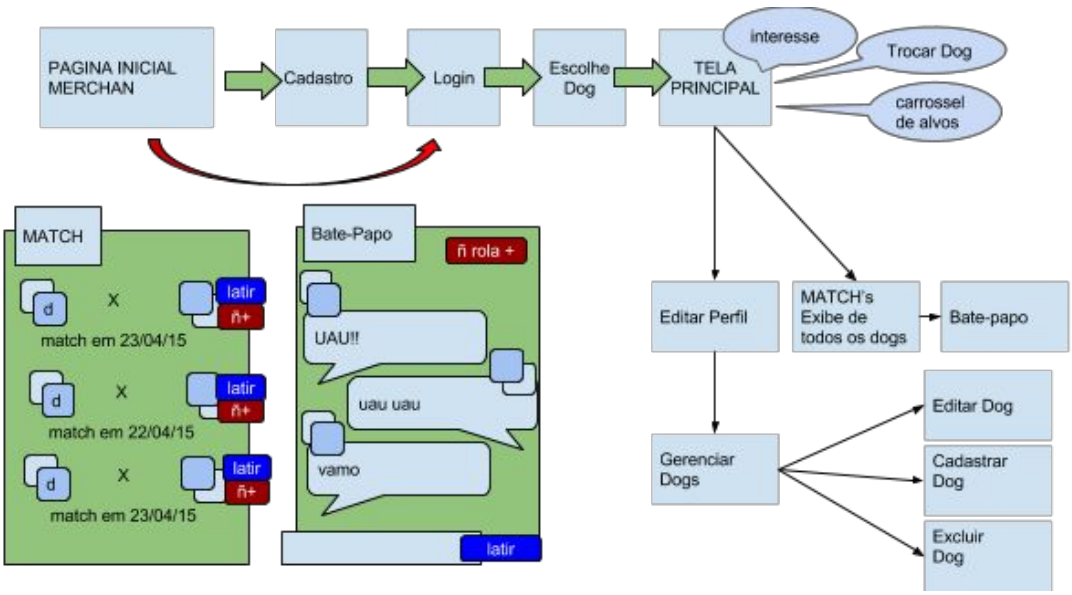

Figura 1 - Mapa mental e wireframes

6.1 Requisitos Funcionais

Os requisitos funcionais foram definidos em documento a parte, disponível na *wiki* do projeto no GitHub pelo link https://github.com/plinio/matchDog/wiki/Lista-de-Requisitos, como todos os outros artefatos produzidos. Este levantamento foi feito a partir da interpretação dos cartões de histórias dos usuários, sendo que na planilha são mencionados os cartões correspondentes com cada requisito funcional encontrado.

6.2 Atributos de Qualidade

Para auxiliar a equipe nos objetivos principais do sistema, foram definidos os atributos de qualidade a seguir.

i) Usabilidade

O sistema deve ser de fácil uso, permitindo realizar suas tarefas sem necessitar de grande esforço para entender o funcionamento desde o início. O MatchDog é um sistema que tem seu foco no encontro de cachorros e de seus donos, sendo assim, não pode ter curva de aprendizagem alta, devido ao fato de seus usuários (donos de cachorro) terem conhecimento variado em relação à tecnologia. **Prioridade:** alta.

ii) Confidencialidade

O sistema deve divulgar os dados do usuário somente no nível que for permitido. O usuário deve ter a possibilidade de escolher o quanto quer se expor no sistema, podendo publicar as fotos que ele achar mais conveniente, da mesma forma, poucos dados adicionais sobre ele devem ser informados a outro usuário caso não exista interesse mútuo entre os dois. As informações que o usuário coloca no site não devem ser expostas em outras redes sociais ou sites. **Prioridade**: alta.

iii) Escalabilidade

Tratando-se de um serviço web, o sistema e seus recursos devem possuir mecanismos que possibilitem seu crescimento de maneira rápida e fácil, de acordo com o aumento da demanda, seja em funcionalidades e módulos, bem como em recursos de hardware. **Prioridade**: média

7 Visão de Análise e Projetos

Os artefatos gerados na atividade de Elicitação de Requisitos, foram as principais entradas da atividade de Análise de Requisitos.

A partir das histórias de usuários, foram realizadas todas as revisões e verificação dos casos de teste, dando origem à lista de requisitos apresentada na seção 6.1. Esta lista traz uma descrição detalhada de cada requisito e a quais dos cartões está vinculada, além da prioridade do requisito.

7.1 Diagramas

Os diagramas definidos foram de Classes e Atividade. Apresentados a seguir.

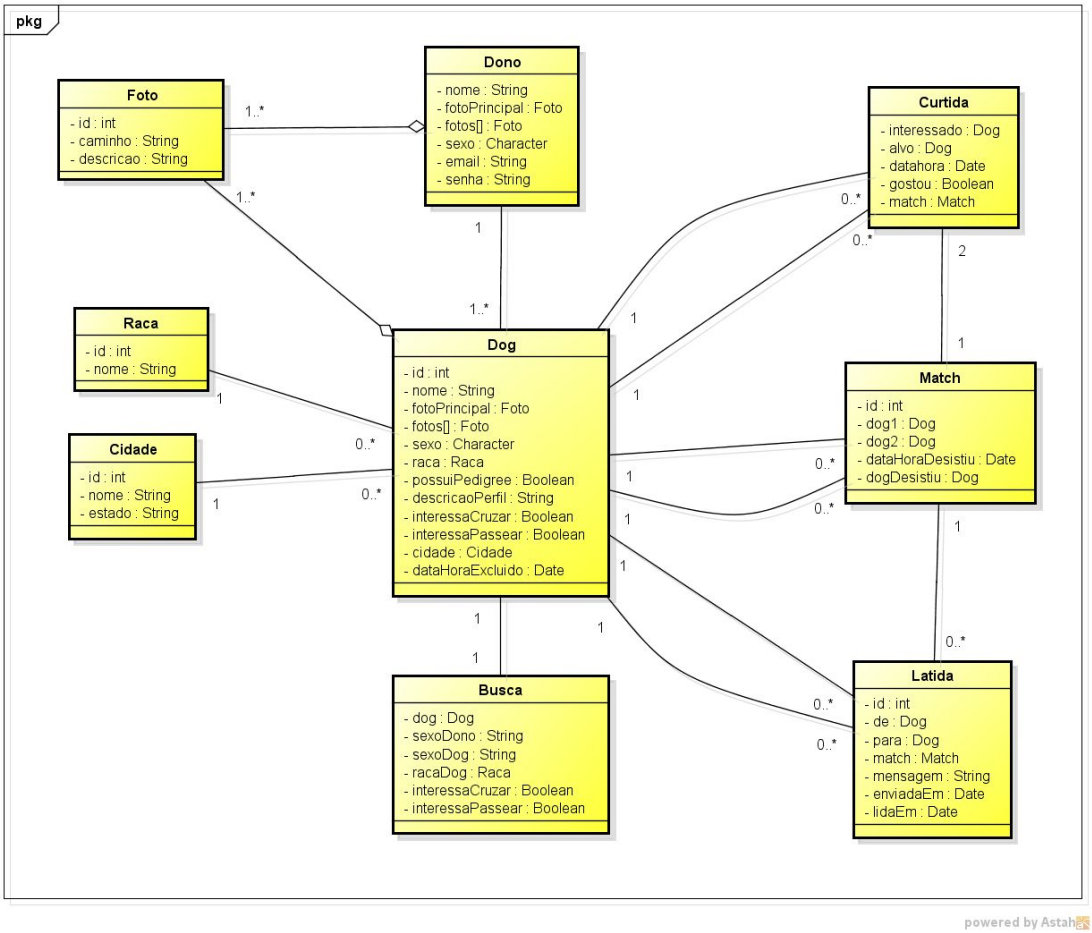

Figura 2- Diagrama de classes

A figura 2 apresenta o diagrama de classes definindo as entidades envolvidas no projeto, bem como as restrições e considerações sobre seus relacionamentos.

Para o diagrama de atividade foram considerados os dois fluxos mais complexos, sendo o diagrama de atividade de Login, representado na Figura 3, onde podemos verificar as opções dadas ao usuário, que são autenticação pelo sistema MatchDog ou pelo Facebook, incluindo as capacidades que o sistema deve ter para permitir completar o cadastro.

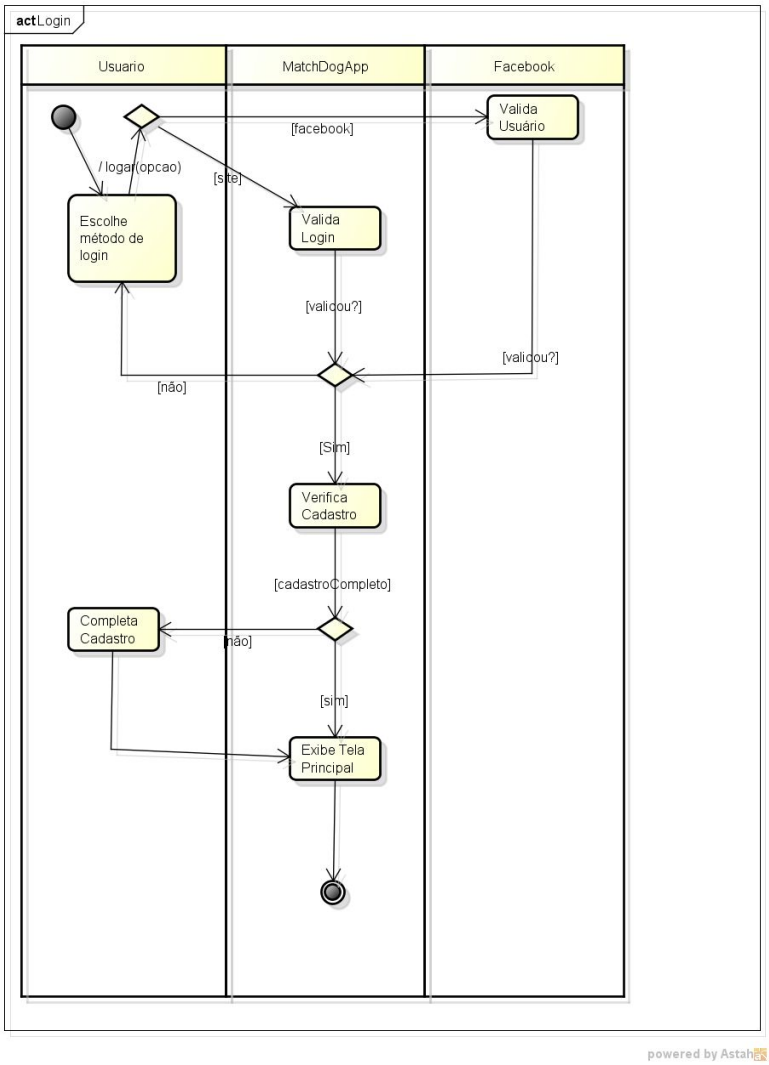

Figura 3 - Diagrama de atividade de Login

Outro fluxo importante, apresentado na Figura 4, é da tela inicial do usuário autenticado, que permite demonstrar seu interesse ou rejeição por um conjunto de dono/cachorro de acordo com sua busca e os comportamentos que o sistema deve exercer sobre cada ação.

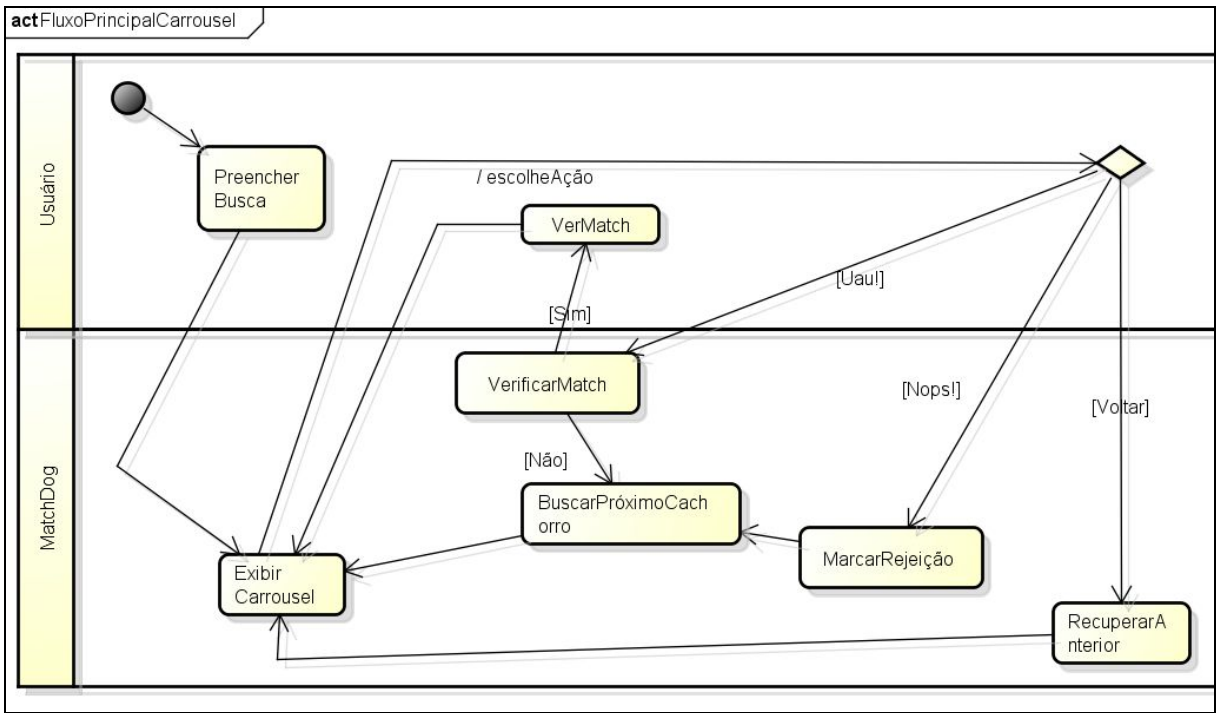

Figura 4 - Diagrama de atividade do Fluxo Principal do Sistema

8 Visão de Implantação

Para implantação por tratar-se de um projeto piloto, optou por utilizar uma máquina no Heroku, a equipe preferiu seguir este caminho pela facilidade de implantação e compatibilidade dos requisitos e baixo custo de implantação, podendo ser executado em http://matchdog.herokuapp.com/ .

Os guias de implantação estão descritos na *Wiki* do projeto no GitHub e acessíveis pelos links a seguir:

- Ubuntu Server 14.04: https://github.com/plinio/matchDog/wiki/Guia-de-Implantação

- Implantação no Heroku: https://github.com/plinio/matchDog/wiki/Guia-de-Implantação-no-Heroku

O processo gerou uma Release que foi nomeada de v1.0, que está disponível juntamento com sua Release Note em https://github.com/plinio/matchDog/releases.

9 Visão de Uso

O principal foco foi na preocupação de um ambiente intuitivo, ou seja, simples para o usuário navegar. Também houve um cuidado com a estética para que seja confortável aos olhos dos usuários.

O manual completo pode ser acessado em: https://github.com/plinio/matchDog/blob/master/docs/manualUsuarioMatchDog.pdf

10 Revisão do Projeto

A ideia inicial foi de fazer um processo totalmente experimental, com características de alguns processos bem definidos, mas também deixar livre para verificar como seria a execução do projeto considerando a sintonia dos integrantes, visto que é o quinto projeto realizado pelo mesmo time. A sinergia da equipe permitiu observar os seguintes fatores (se comparados com outros dois projetos que os mesmos participam com outros 2 integrantes dististos):

- **Liberdade de opinião:** Os integrantes da equipe criaram facilidade em falar o que pensam sem que o outro fique ofendido, comparado com outro projeto, a falta dessa liberdade atrapalhou o desenvolvimento.
- **Potencializar as habilidades:** Com tendências diferentes, aparentemente um modelo personalizado de processo permitiu cada um explorar suas habilidades no momento que ela era necessária e que cada um sentia-se à vontade.

Talvez a percepção de que estes fatores contribuíram podem ser relacionados ao entendimento deste grupo e da mesma forma podem ter funcionado devido às características dos integrantes quanto ao entendimento de trabalho de equipe.

10.1 Processo de Desenvolvimento

No início do projeto definiu-se um desafio, que seria de fazer experimentação até mesmo na definicação das etapas, tirando os integrantes da equipe das suas zonas de conforto, mas também utilizando alguns elementos conhecidos por pelo menos um dos membros para transferir conhecimento aos demais.

Para melhor entendimento do que o sistema deveria executar, optou-se por todos os membros da equipe participarem de cada etapa, sendo assim o processo de desenvolvimento foi dividido em quatro atividades principais, como já detalhadas na seção 4 deste documento: Elicitação de Requisitos, Análise de Requisitos, Implementação e Testes, e Implantação. Sendo que o avanço para uma etapa não impediu o retorno a uma das etapas anteriores ou o avanço para uma próxima etapa, assim, fica clara nossa intenção de fazer um processo iterativo e incremental.

10.2 Principais problemas e tomada de decisão

Muitos dos problemas encontrados surgiram na utilização da linguagem e do framework, pois na questão das regras do sistema consideramos que ficaram suficientemente claras. Mesmo assim, algumas decisões de funcionalidades descritas nos requisitos foram redefinidas durante a atividade de implementação, algumas questões promovidas pelo avanço do conhecimento, como por exemplo, a utilização da biblioteca Omniauth para integrar com o Facebook ao invés de desenvolver algo próprio; e outras questões promovidas pela verificação de que a funcionalidade não faria tanto sentido da maneira que foi planejada, como é o caso da opção de voltar em caso de arrependimento de não ter sinalizado interesse no cachorro anterior.

Alguns problemas surgiram com a questão de *layout*, posicionamento de elementos que foram sendo adequados para otimizar a usabilidade.

Todos os pontos problemáticos foram sempre colocados em votação ou por conversa no Whatsapp ou por Hangout antes de serem efetivamente alterados.

10.3 Atribuição de atividades

Os integrantes, por livre adesão em reunião, identificaram pontos nos quais gostariam de trabalhar, ficando responsável pela etapa.

Inicialmente todos participaram da questão de levantamento de ideias e preenchimento dos cartões de história de usuários. Depois do levantamento de requisitos, um membro ficou responsável de criar as *Issues* no GitHub e cada um pegou para si a tarefa que tinha mais afinidade. A lista de *Issues* e de quem a executou está disponível em https://github.com/plinio/matchDog/issues?q=is%3Aissue+is%3Aclosed.

10.4 Técnicas de gerenciamento, monitoramento e controle

A cada reunião definida era marcada uma próxima reunião com os pontos que cada um deveria cumprir, a definição de prazos auxiliou a equipe em cumprir o prazo de maneira satisfatória, visto que todos os pontos vistos como prioritários foram atingidos.

A equipe optou por não elaborar um cronograma fixo por entender que seria melhor focar nos objetivos a serem atingidos invés do prazo, colocando apenas como data limite para fim da codificação o prazo de 15/07/2015. Este prazo foi atingido com antecedência e permitiu realizar alguns ajustes finais.

Foi verificada a disponibilidade de tempo de cada integrante para equilibrar a sobrecarga de tarefas com outras atividades.

11 Revisão Individual

Os membros do projeto tiveram experiências pessoais importantes, que são relatadas nesta seção do documento.

11.1 Artur Carvalho Alves (aca5)

A jornada do desenvolvimento desse trabalho começou com a definição do tema. Inicialmente nenhum dos membros da equipe, que já estava formada desde antes da proposta do trabalho, se mostrou interessado nos temas pré-estabelecidos. Iniciou-se então uma discussão entre os três sobre qual tema propor, daí diversas ideias empreendedoras surgiram, parecia mais que estávamos montando uma startup.

E, nesse clima de empolgação, conseguimos propor um tema motivador e logo começamos as atividades. A proposta de definir um processo de desenvolvimento surgiu, na verdade, logo depois de já termos sujado um pouco das mãos no código-fonte. A intenção de definir o processo foi, de fato, colocar algum controle naquela empolgação toda. E no meu entendimento foi uma decisão bastante acertada, caso contrário o código estaria pronto antes dos requisitos.

Logo no início do projeto sentimos um pouco a dificuldade relacionada ao fato de formarmos uma equipe de desenvolvimento totalmente remota, com cada um dos integrantes em uma cidade distinta, cada um com diversas outras atividades e projetos, cada um com horários distintos para dedicação ao projeto, e com poucos horários em comum para realizar conversas em conjunto. Nesse sentido, um fator que muito colaborou foi o de conseguirmos definir e detalhar o máximo de requisitos logo no início do projeto. Além disso, o fato dos membros já terem trabalhado juntos e já conhecerem a opinião crítica uns dos outros nos dava a liberdade de tomar decisões por conta própria muitas vezes.

No entanto, muito se engana quem acredita que é possível pegar uma tarefa - principalmente se estiver especificada sobre requisitos documentados num processo de desenvolvimento ágil - sentar isoladamente com sua xicára de café e desenvolver algo válido. Portanto, para o andamento desse projeto, percebi o quanto foi importante definir os meios de

comunicação, o quanto as reuniões pré-agendadas do Hangout e as conversas via grupo de WhatsApp foram importantes para sanar dúvidas e até mesmo informar bugs encontrados.

Do ponto de vista de implementação, o uso da linguagem Ruby com certeza foi um dificutador inicial para todos da equipe. Para mim, o que colaborou para isso foi o fato de todos já terem trabalhado com ferramentas ágeis, mas nenhum com RoR. Então sempre ficava aquele sentimento: poxa vida, naquele outro framework isso sairia tão mais rápido. Por 'sorte', a curva de aprendizado realmente é baixa, e no meio do projeto todos já tinham vencido essa fase. E no meu caso, posso dizer que com certeza foi um ótimo aprendizado, pois apesar de já ser um entusiasta de tecnologias ágeis, ainda não tinha me aprofundado nessa linguagem.

Além da linguagem, outra ferramenta que era nova para todos da equipe foi a IDE do Cloud9. E me causou estranheza a velocidade com todos da equipe nos acostumamos a utlizá-la. Lembro que, em uma das reuniões, comentei em brincadeira que estava desenvolvendo 'na minha máquina' com uma outra IDE, e imediatamente ouvi os comentários; 'Por que? Você tá louco?', 'Dúvido...'. Percebi que o uso da IDE, que permite o desenvolvimento colaborativo, somado a ferramenta de comunicação Hangout ajudou bastante a resolver o problema da distancia entre os membros da equipe. Não me vejo fazendo programação em pares com nenhum dos analistas na instituição onde trabalho, talvez apenas por não gostar de ter alguém na minha estação; mas nesse projeto fizemos programação em pares algumas vezes para integrar algumas funcionalidades e funcionou muito bem.

Outra ferramenta com a qual gostei bastante de trabalhar foi o GitHub. No início do trabalho coincidiu de estar migrando do SVN para o Git na instituição onde trabalho, portanto a adoção do Git também nesse projeto me fez não querer nem pensar mais em SVN. Além disso, fizemos questão de usar ao máximo os recursos do GitHub, como as Issues e o Wiki. Essa escolha foi feita pois todos os membros da equipe concordaram com a ideia de que quanto mais estático for o formato da documentação de um projeto, mais 'preguiça' teremos de mantê-lo.

Com tudo isso, considero que conquistei um bom punhado de conhecimento durante o desenvolvimento deste projeto. Foram diversas ferramentas e tecnologias que ainda não conhecia, e que agora considero que possuo um bom entendimento, e melhor que isso, considero a possibilidade de utilizar qualquer uma delas em projetos futuros.

Além do conhecimento adquirido, também foi bom perceber que o uso das ferramentas de comunicação ajudou, e muito, no andamento do projeto. E que o clima que descontração, sempre cultivado na equipe, pôde ser considerado um dos fatores que levaram ao sucesso do trabalho.

11.2 Plinio Antunes Garcia (pag2)

Sob meu ponto de vista, este projeto foi um verdadeiro desafio. O cenário desde desafio incluía três desenvolvedores (e owners) que nunca programaram em Ruby e que estavam geograficamente distribuídos.

A definição do tema do projeto foi um presságio do que seria o nível de interação entre os colegas de equipe. Decidimos não optar por um tema fornecido em sala de aula e aproveitar a oportunidade de criarmos um tema. Após algumas ideias e discussões para amadurecê-las decidimos pelo MatchDog em comum acordo e todos ficamos motivados a desenvolvê-lo.

A reunião de *brainstorming* serviu para alinharmos as ideias e excluir aquelas que poderiam tirar o foco da aplicação. Após isto, outro ponto que me chamou a atenção foi a definição do processo de desenvolvimento. O cenário era favorável ao eXtreme Go Horse, contudo decidimos definir um processo, ainda que personalizado, para que possamos traçar o caminho do projeto até o objetivo final. Particularmente, eu era preso à metodologias e à segui-las o mais próximo possível do que seus autores propunham, porém, a experiência de colaborar em uma metodologia "híbrida" que melhor se adequasse ao nosso cenário me pareceu muito positiva e pretendo aplicar este aprendizado em outros projetos particulares.

Outro ponto de aprendizado a se destacar foi à linguagem e ao framework utilizados. Nunca havia programado em Ruby, o que inicialmente foi um obstáculo na fase de implementação, mas que foi superado devido à curva de aprendizado da linguagem e ao fácil acesso aos materiais na internet. Com relação ao Rails, meu aprendizado foi duplo, pois minha experiência com frameworks MVC era pouca e, portanto, me permitiu conhecer não somente Rails mas também um pouco mais sobre MVC, com a ajuda dos meus colegas.

Sobre a IDE escolhida, à princípio me causou estranheza a ideia de “programar na nuvem”, porém esta percepção foi mudando ao longo do projeto e as funcionalidades colaborativas do ambiente Cloud9 foram sobrepondo a repúdia inicial. Além do Cloud9, pude observar que as ferramentas de comunicação foram de suma importância para o bom andamento do projeto neste cenário de integrantes distantes geograficamente. Utilizamos muitas reuniões via videoconferência (Hangout) e o app Whatsapp serviu de apoio na troca de informações curtas.

Eu estava há muitos anos trabalhando mais com Subversion, e este projeto me ajudou a renovar minha experiência com o Git. Destaco o compromisso que a equipe teve em utilizar os recursos do GitHub, atuando não somente como um repositório de código fonte mas também como nosso centralizador de artefatos e controle de tarefas.

Acredito que o projeto cumpriu sua missão de nos ensinar as etapas da engenharia de software, apesar da implementação ter ocupado a maior parte do tempo, as fases de projeto e análise foram exercitadas com seriedade e considero que se estas não tivessem sido bem executadas o sucesso do projeto estaria comprometido.

O aprendizado técnico e científico foi grande mas o aprendizado humano também merece destaque. O comprometimento da equipe e o alto grau de alinhamento entre os membros foram determinantes para que este desafio fosse cumprido.

11.3 Tiago Heineck (th)

O projeto surgiu de forma espontânea em conversa sobre possíveis ideias que poderiam ser desenvolvidas. Logo depois foram trocadas ideias de quais funcionalidades poderia ter para atender de maneira simples o que seria pretendido. Foi quando paramos para definir quais etapas teria o nosso processo de desenvolvimento, este ponto foi importante por guiar toda a execução até o seu final.

Uma primeira impressão de dificuldade foi na questão da equipe estar distribuída, porém logo foi suprimida pelas ferramentas utilizadas, que ofereceram recursos suficientes para boas reuniões e seções de trabalho em conjunto para sanar problemas.

No início do desenvolvimento surgiu uma dificuldade real na utilização do Ruby e Ruby On Rails, na questão de conhecer os recursos e a organização dos mesmos, então

analisando a estrutura foi identificado que era muito similar a frameworks que utilizo em outras linguagens. Depois dessa breve análise ficou mais tranquilo de “inverter a chave”, pois os recursos eram claros, como helpers, padrão MVC, convenção na disposição de pastas de views e o padrão *Active Record*.

O ponto mais positivo, vejo que foi a sinergia entre os membros da equipe, dividindo tarefas e responsabilidades e organizando os trabalhos para não deixar a parte do outro atrasar, prova disso foram algumas seções de *Hangout* onde ficávamos conectados para trocar ajuda mútua e resolver os pontos mais rapidamente.

A participação deste projeto com meus colegas reforçou algo que já vinha pensando para os projetos que faço parte em minha instituição, que é reavaliar a maneira como nos comunicamos, por ter uma equipe distribuída em 16 unidades, esse de fato é o ponto que falta para aprimorar o nosso processo.

As principais lições foram: é possível trabalhar com software com equipe distribuída desde que haja comunicação eficiente; a pró-atividade dos membros ajuda no andamento; conhecer bem as ferramentas utilizadas antes de começar facilita o processo; e preciso aprimorar mais nos estudos de Engenharia de Software e Requisitos para definir melhor os projetos que participo.

12 Entrega

Todos os itens relatados neste projeto estão disponíveis nos links identificados a seguir.

- Wiki do projeto contendo todas as seções que serão citadas a seguir disponível em https://github.com/plinio/matchDog/wiki
- Atas de reuniões em nossa wiki no link https://github.com/plinio/matchDog/wiki/Reuniões
- User Story Cards disponíveis em https://github.com/plinio/matchDog/wiki/User-Story-Cards

- Diagramas, logotipos e protótipos disponíveis na pasta de documentação juntamente com os códigos fontes, acessível em https://github.com/plinio/matchDog/tree/master/docs
- Processo de desenvolvimento e todos os artefatos produzidos disponíveis no link https://github.com/plinio/matchDog/wiki/Processo-de-Desenvolvimento
- Configuração do Ambiente de Desenvolvimento disponível em https://github.com/plinio/matchDog/wiki/Ambiente-de-Desenvolvimento
- As tarefas na fase de desenvolvimento estão em https://github.com/plinio/matchDog/issues
- Código fonte disponível no GitHub pelo Link https://github.com/plinio/matchDog
- Guia de Implantação no Heroku disponível em https://github.com/plinio/matchDog/wiki/Guia-de-Implantação-no-Heroku
- Guia de Implantação em Ubuntu Server 14.04 disponível em https://github.com/plinio/matchDog/wiki/Guia-de-Implantação
- Projeto Final rodando em http://matchdog.herokuapp.com
- Vídeo demonstrativo das principais funcionalidades: https://youtu.be/1eYSvEpVx-w

13 Referências

IBGE, 2013 - | sala de imprensa | notícias | PNS 2013: três em cada quatro brasileiros costumam buscar atendimento médico na rede pública de saúde. Disponível em: <http://saladeimprensa.ibge.gov.br/noticias?view=noticia&id=1&busca=1&idnoticia=2902>. Acesso em: 17 jul. 2015.

IBM, 2003 - Rational Unified Process.

AGILE MODELING, 2015 - User Stories: An Agile Introduction. Disponível em: <http://www.agilemodeling.com/artifacts/userStory.htm>. Acesso em: 18 jul. 2015.