

**COPPE - UFRJ**  
**Programa de Engenharia de Sistemas e Computação**  
**Universidade Federal do Rio de Janeiro**

**Computadores como Construções Sociotécnicas - 2o Período de 2014**

Professor: Henrique Cukierman

**Um estudo sobre processo de  
desenvolvimento de software livre: o caso do  
WordPress**

**Rodrigo Primo**

RIO DE JANEIRO – DEZEMBRO – 2014

# SUMÁRIO

1. PREFÁCIO.....	3
2. INTRODUÇÃO.....	4
2. METODOLOGIA.....	7
2.1. Questões para orientar a pesquisa.....	7
2.2. Documentos da comunidade.....	8
2.3. Mineração de dados.....	9
3. PROCESSO DE DESENVOLVIMENTO DO WORDPRESS.....	11
3.1. Histórico.....	11
3.2. Organização do projeto.....	13
3.3. Comunicação.....	14
3.4. Ciclo de desenvolvimento de uma nova versão.....	16
3.5. Dados quantitativos.....	17
3.5.1 Análise do repositório de código.....	17
3.5.2 Localização dos desenvolvedores.....	21
4. CONCLUSÕES.....	22
5. REFERÊNCIAS.....	25

# 1. PREFÁCIO

Estou pelo segundo ano seguido no Mission Bay Conference Center em San Francisco para o principal encontro anual da comunidade do WordPress (WP). O site do projeto define a comunidade como:

“Everything you see here, from the documentation to the code itself, was created by and for the community. WordPress is an Open Source project, which means there are hundreds of people all over the world working on it.”<sup>1</sup>.

Mais de mil pessoas participam do evento de um dos maiores softwares livres da atualidade. No palco um desenvolvedores apresenta uma palestra intitulada “*The Future of WordPress is Global*”<sup>2</sup>. Antes mesmo da palestra começar, me pergunto, já que o futuro é global, por que não estamos em uma cidade da África ou do leste europeu? A palestra começa afirmando que enquanto apenas cinco por cento da população mundial fala inglês, dois terços de todas as instalações do WordPress são neste idioma.

Resolvo abrir o mapa de empregados da Automattic<sup>3</sup>, a empresa do fundador do software e uma das que mais contribui com a comunidade. Esta empresa possui como uma de suas principais características o trabalho remoto. São cerca de duzentos e cinquenta funcionários, espalhados em mais de cem localidades distintas. Ao olhar o mapa vejo que seis funcionários estão na América do Sul (sendo um no Brasil), alguns na Oceania e Ásia. A grande maioria está nos Estados Unidos e um número significativo está na Europa. Não existe nenhum funcionário da Automattic na América Central, México, África e em boa parte da Ásia.

A palestra segue falando sobre como melhorar o suporte à diferentes idiomas no WordPress e como facilitar o processo de tradução. Minha atenção volta novamente ao computador onde busco identificar a nacionalidade dos trinta e dois desenvolvedores com acesso ao repositório de código. Destes vinte e sete são de países onde o inglês é o

---

1 Disponível em: <https://wordpress.org/about/>. Acesso em: 28 dez. 2014.

2 Disponível em: <https://wordpress.tv/2014/11/03/andrew-nacin-the-future-of-wordpress-is-global/>. Acesso em: 28 dez. 2014.

3 Disponível em: <http://automattic.com/map/>. Acesso em: 28 dez. 2014.

principal idioma, sendo dezoito dos Estados Unidos.

## 2. INTRODUÇÃO

ROSZAK (1988) em seu trabalho sobre o computador e a contracultura narra a história dos primeiros grupos nos Estados Unidos a usarem o computador como uma ferramenta de participação política:

“Aproximadamente no fim da década de 60, havia uma outra espécie de hackers no horizonte, vindos principalmente da Costa Oeste, das fileiras do movimento antibélico. Estes eram os hackers radicais ou guerrilheiros. (...) Nas mãos deles, a tecnologia da informação se tornaria mais próxima de um instrumento da política democrática.” (p. 209)

Para esses grupos o que estava em disputa era o controle da informação. O texto acaba em tom pessimista na década de 80 com a transformação da Apple, empresa que nasce de dentro da cultura hacker, numa nova IBM. Este trabalho pretende dar continuidade à história iniciada por ROSZAK e trazê-la para os dias de hoje. Um dos pressupostos é que o movimento do software livre atual é um desdobramento da ação dos hackers radicais na luta pelo controle da informação e pelo uso da tecnologia como um instrumento de participação política.

Será dado enfoque na maneira como se organiza o desenvolvimento do WordPress, software livre cujo um dos encontros foi narrado no prefácio. Em particular pretende-se problematizar o discurso presente na comunidade e descrito anteriormente de uma composição global da comunidade.

Como referência usa-se a reflexão feita por alguns autores dos Estudos de Ciência, Tecnologia e Sociedade (LATOURE, 1997; MARQUES; POLANCO, 1985) que desconstruem a ideia de uma ciência universal e falam do papel importante do contexto local onde as várias ciências são produzidas. Para eles, o que é defendido como ciência universal é uma ciência particular no poder, no caso a ciência produzida em alguns países da Europa e Estados Unidos. Alguns, como MARQUES e POLANCO, vão além e defendem a importância de se pensar uma ciência latino-americana produzida na região e preocupada com os problemas locais. POLANCO cria o termo “fuga interior de cérebros” justamente para criticar a ciência produzida nos países latino-americanos

voltada para as questões e problemas de outras realidades, mais especificamente as questões e problemas dos poucos países porta-vozes da ciência dita universal.

É possível extrapolar esta discussão e questionar até que ponto a universalidade do movimento do software livre não representa, assim como na ciência, um particular no poder?

Para buscar respostas para esta questão a ideia é investigar a organização do desenvolvimento do WordPress. Qual é a estrutura e o processo de desenvolvimento do software? Quem são os principais atores? Onde estão baseados os desenvolvedores do projeto? O levantamento desses dados contribui para a impressão apresentada na cena de abertura de que existe uma concentração grande de atores no eixo Estados Unidos e Europa, em especial nos países anglófonos?

Com esse trabalho pretende-se também contribuir para a aproximação das comunidades de software livre da comunidade de Engenharia de Software. Já que apesar do crescimento dos softwares livres nos últimos anos, ainda são poucas as pesquisas que procuram entender as peculiaridades desta maneira de desenvolver e publicar softwares. Como identifica SCACCHI (2010):

“Even though Free/Open Source Software (FOSS) is widely used, much of the Computer Science research community has yet to fully recognize its potential to change the world of research and development of software-intensive systems across disciplines. As little as four years ago, FOSS was still somewhat marginal to SE, appearing in one case as simply another concurrent development methodology, rather than an a new approach to software .” (p. 315)

O mesmo autor argumenta que a Engenharia de Software poderia se beneficiar ao se voltar para o mundo do software de código aberto (*idem*, p. 316). Segundo ele, ao estudar softwares proprietários, os pesquisadores muitas vezes se deparam com inúmeras dificuldades para obter os dados necessários para seus estudos. Ou então, quando conseguem estes dados, não são autorizados à publicá-los para que outros pesquisadores possam reproduzir seus trabalhos. Já no caso dos softwares livres, por sua natureza descentralizada e aberta, grande parte do material de interesse dos pesquisadores já se encontra disponível. O que tende a facilitar o trabalho de pesquisa.

“FOSSD project source code, artifacts, and online repositories represent and offer new publicly available data sources of a size,

diversity, and complexity not previously available for SE research, on a global basis.” (p. 318, SCACCHI, 2010).

Para possibilitar a comparação com outros estudos de caso de softwares livres optou-se por utilizar a mesma metodologia utilizada por MOCKUS *et al.* (2002) em estudo semelhante sobre o processo de desenvolvimento dos softwares Apache e Firefox. Analisa-se o WordPress sob aspectos qualitativos e aspectos quantitativos. Em relação ao primeiro aspecto, é apresentado um breve histórico do software, bem como a maneira como o seu desenvolvimento é organizado, com base no estudo de documentos produzidos pela própria comunidade. Já para o levantamento de aspectos quantitativos, foi utilizada a técnica de mineração de repositório de código para extrair métricas sobre a participação dos desenvolvedores no projeto.

Este trabalho está dividido em três partes. Primeiro é apresentada a metodologia utilizada para o levantamento de dados. Na sequência, vem a parte principal do trabalho, onde são descritas as características do processo de desenvolvimento do WordPress levantadas qualitativamente e quantitativamente. E por fim, a última parte traz algumas considerações finais.

## 2. METODOLOGIA

Com o intuito de permitir que este trabalho seja comparado com outros estudos de caso sobre processo de desenvolvimento de softwares livres ou proprietários, optou-se por seguir a mesma metodologia utilizada por MOCKUS *et al.* (2002) em um trabalho sobre o processo de desenvolvimento dos softwares livres Firefox (naquela época chamado de Mozilla) e Apache.

Para entender o processo de desenvolvimento do WordPress foram utilizadas as mesmas questões utilizadas no artigo citado acima e para respondê-las foram analisados dados qualitativos e dados quantitativos.

### 2.1. Questões para orientar a pesquisa

MOCKUS *et al.* (2002) utilizam seis questões para nortear a pesquisa sobre o processo de desenvolvimento dos softwares Firefox e Apache. Uma é respondida através de um levantamento qualitativo a cerca do software estudado, duas são respondidas através de mineração de dados do repositório de código e três são respondidas através de mineração do sistema utilizado para gestão das tarefas e relato de *bugs*.

Por uma limitação de tempo, não foi possível coletar e analisar os dados para responder as três últimas questões. Sendo assim, neste trabalho foram utilizadas apenas as três primeiras questões referentes ao levantamento qualitativo e a mineração de dados a partir do repositório de código. Também foi adicionada uma quarta questão, que não estava presente no trabalho usado como referência, para contribuir com a discussão sobre a composição global da comunidade do WP. São elas:

*Questão 1: Quais são os processos utilizados para o desenvolvimento do WordPress?*

Para responder esta pergunta será apresentado um breve histórico do software e os principais aspectos da organização da comunidade que participa de seu desenvolvimento. Mais informações sobre as fontes escolhidas para responder esta



pergunta podem ser vistas na seção “Documentos da comunidade”. As seções “Histórico”, “Organização do projeto”, “Comunicação” e “Ciclo de desenvolvimento de uma nova versão” discutem aspectos relacionados a esta pergunta.

*Questão 2: Quantas pessoas participam do desenvolvimento de novas funcionalidades? Quantas pessoas resolvem bugs?*

O objetivo desta questão é verificar o tamanho da comunidade de desenvolvimento do WordPress e levantar quantas pessoas participam de cada um destes tradicionais papéis do processo de desenvolvimento de um software (criação de novas funcionalidades e resolução de *bugs*). A seção “Dados quantitativos” discute aspectos relacionados a esta e as demais perguntas.

*Questão 3: Essas funções foram desenvolvidas por grupos distintos de pessoas?*

Como se dá a divisão do trabalho dentro de uma comunidade de software livre considerando que os participantes, via de regra, são livres para escolherem o que fazem? É possível identificar grupos distintos trabalhando na criação de novas funcionalidades e resolvendo *bugs*?

*Questão 4: Onde moram os desenvolvedores do WordPress?*

Para atender a especificidade deste trabalho de tentar identificar a onde estão baseados os programadores que participam do desenvolvimento do WP esta questão foi adicionada.

## **2.2. Documentos da comunidade**

Para responder a primeira questão, que diz respeito à aspectos qualitativos do processo de desenvolvimento do WordPress, foram utilizados materiais publicados pela própria comunidade, em especial o *WordPress Handbook*<sup>4</sup>. Este documento foi criado em 2013 durante um WordCamp<sup>5</sup> em San Francisco, justamente com o objetivo de centralizar em um único lugar as informações relativas à organização do desenvolvimento do software. Também foram utilizados *posts* em *blogs* de membros comunidade e outras publicações relevantes disponíveis na internet.

---

4 Disponível em: <http://make.wordpress.org/core/handbook/>. Acesso em: 28 dez. 2014.

5 Na seção “Histórico” explica-se o que é um WordCamp.

## 2.3. Mineração de dados

Para responder a segunda e a terceira questões foram minerados dados a partir do repositório Git do software<sup>6</sup>. Foi utilizada a ferramenta *gitinspector*<sup>7</sup>, um software para extração de dados estatísticos de repositórios criada por pesquisadores das universidades Chalmers University of Technology e Gothenburg University, ambas da Suécia.

Conforme será visto em maiores detalhes na seção “Organização do projeto”, apenas um pequeno grupo de desenvolvedores possui acesso de *commit* no repositório de código do WordPress. As contribuições de código da comunidade são realizadas enviando *patches* para um sistema de gestão de projetos chamado Trac. Os *patches* são revisados por um desenvolvedor com permissão e então enviados para o repositório de código em seu nome. O autor original da contribuição é identificado através de uma marcação com a *tag* “props” (do inglês *proposition*) seguido do seu nome de usuário na mensagem de *commit*. Também na mensagem de *commit* uma outra *tag*, “fixes”, é utilizada para identificar uma contribuição que corrige um *bug*.

Para dar conta destas peculiaridades do WordPress, foi necessário realizar algumas modificações no *gitinspector* para extrair o número de *commits* realizados por membros da comunidade que não tem acesso de escrita ao repositório de código e também para diferenciar *commits* que criam uma nova funcionalidade daqueles que corrigem um *bug*. Com o intuito de permitir que outros pesquisadores possam repetir este estudo, uma versão do *gitinspector* com as modificações utilizadas está disponível no GitHub do autor<sup>8</sup>.

Na mineração dos dados do repositório de código do WordPress foram analisados todos os *commits* realizados, desde o primeiro em abril de 2003 até maio de 2014. Apenas modificações em arquivos com código PHP, JS ou CSS foram considerados. Demais tipos de arquivos foram ignorados.

Para responder a quarta questão, utilizou-se a lista de desenvolvedores compilada para as duas questões anteriores e então foi criado um pequeno script<sup>9</sup> que consulta o

---

6 Disponível em: [git://develop.git.wordpress.org/](https://github.com/WordPress/WordPress). Acesso em: 28 dez. 2014.

7 Disponível em: <https://code.google.com/p/gitinspector/>. Acesso em: 28 dez. 2014.

8 Disponível em: [https://github.com/rodrigoprino/gitinspector\\_wp](https://github.com/rodrigoprino/gitinspector_wp). Acesso em: 28 dez. 2014.

9 Disponível em: [https://github.com/rodrigoprino/mestrado/tree/master/get\\_location](https://github.com/rodrigoprino/mestrado/tree/master/get_location). Acesso em: 28

perfil de cada um deles no site da comunidade do WordPress<sup>10</sup> em busca da informação sobre o seu local de residência. Esse é um campo aberto de preenchimento não obrigatório, por tanto foi necessário tratar dados. Isso foi feito em parte automaticamente pelo próprio script utilizando a API do Open Street Maps e em parte manualmente após a execução do script.

---

dez. 2014.

10 O formato da URL com o perfil de um usuário no site do WordPress é <https://profiles.wordpress.org/NOME-DO-USUARIO>

# 3. PROCESSO DE DESENVOLVIMENTO DO WORDPRESS

O WordPress é um sistema de gerenciamento de conteúdo (CMS) para a web escrito em PHP e JavaScript utilizando a base de dados MySQL. Sua primeira versão foi publicada em 2003 sob a licença de software livre GPLv2. O SVN é utilizado como sistema de controle de versão, sendo que desde 2013 foi disponibilizado um espelho oficial utilizando o Git.

## 3.1. Histórico

A história do WordPress começa no fim de 2002, quando o desenvolvedor original de um software para blogs pessoais chamado b2/cafelog, também escrito em PHP e publicado sob a GPLv2, abandonou o projeto por falta de tempo. Foi então que Matt Mullenweg e Mike Little, ambos usuários do b2/cafelog e preocupados com o destino de seus blogs pessoais, resolveram criar um *fork*<sup>11</sup>. Assim nasceu o WordPress. O primeiro *commit* foi feito em abril de 2003. Até maio de 2014 foram contabilizados no repositório de código 25.692 *commits*.

Até 2004 a base de usuários do software era bem modesta e não passava de algumas poucas dezenas. Neste ano porém, um outro software para blogs chamado Movable Type, com uma base de usuários muito maior, decidiu abandonar sua licença livre e adotar uma licença proprietária. Isso causou uma migração de boa parte da comunidade do Movable Type para o WordPress. Contribuindo significativamente para o crescimento do projeto. No mesmo ano, foi lançada a versão 1.2 do WordPress, com a introdução de um modelo de código estruturado para permitir o desenvolvimento de plugins, aspecto central da filosofia do projeto. A ideia é ter um núcleo enxuto com

---

11 O post escrito por Matt Mullenweg anunciando a intenção de criar o fork com um comentário de Mike Little demonstrando seu interesse em participar do novo projeto pode ser visto no link <http://ma.tt/2003/01/the-blogging-software-dilemma/>. Acesso em: 28 dez. 2014.

apenas alguns poucos recursos básicos e uma estrutura de *hooks* que permite a criação de novas funcionalidades através de plugins criados por membros da comunidade.

No ano seguinte, duas novas versões foram publicadas introduzindo outros recursos importantes da ferramenta: um sistema de temas, a possibilidade de criação de páginas estáticas e um sistema de controle de permissões dos usuários. Também em 2005, foi criada a Automattic. Empresa de um dos fundadores, Matt Mullenweg, é a responsável pelo portal de hospedagem de blogs e sites [wordpress.com](http://wordpress.com) e até hoje uma das principais empresas a investir no WP.

Em agosto de 2006 foi organizado em San Francisco o primeiro WordCamp, um evento local com palestras voltadas para usuários, desenvolvedores e designers. Com o passar do tempo os WordCamps se tornaram o principal evento da comunidade. Desde 2006 até maio de 2014, foram organizados 364 WordCamps, em 164 cidades diferentes e em 50 países distintos<sup>12</sup>. Além destes, anualmente ocorre um encontro de uma semana com todos os desenvolvedores *core* do software. A partir de 2012, este encontro foi aberto também para membros da comunidade e passou a ser chamado de WordPress Community Summit.

Inicialmente focado exclusivamente na criação de blogs, o WordPress foi aos poucos ampliando seu escopo para se tornar hoje um dos principais softwares para gestão de conteúdo, ou *Content Management System* (CMS). A versão 3.0, lançada em 2010, foi um marco nesse sentido, introduzindo novos recursos que marcaram de vez a transição do WordPress de um sistema de blogs para um CMS completo. Pesquisa recente mostra que 18,9% dos sites da internet servem conteúdo a partir deste software<sup>13</sup>. Estima-se que existam mais de 60 milhões de sites nesta plataforma<sup>14</sup> e somente a versão 4.1, a última lançada no momento de escrita deste artigo, foi baixada 22 milhões de vezes<sup>15</sup>.

---

12 As estatísticas sobre os WordCamps foram extraídas da página “Past WordCamps” (<http://central.wordcamp.org/schedule/past-wordcamps/>) do site oficial que concentra informações sobre este evento.

13 Dado apresentado por Matt Mullenweg durante sua apresentação, State of the Word 2013, no WordCamp San Francisco 2013. Disponível em: <http://www.slideshare.net/photomatt/state-of-the-word-2013/39>. Acesso em: 28 dez. 2014.

14 COALLO (2012).

15 Dado obtivo no contador de downloads do WordPress. Disponível em: <https://wordpress.org/download/counter/>. Acesso em: 28 dez. 2014.

## 3.2. Organização do projeto

O WordPress é um projeto que se define como uma meritocracia, governada por um time de dez desenvolvedores, chamados *core developers*, tendo como líder o fundador do projeto Matt Mullenweg. Essa equipe é a responsável por todas as decisões relacionadas com o desenvolvimento do software. Destes dez desenvolvedores, cinco são da Automattic. Os demais são funcionários de outras empresas ou então são profissionais autônomos.

Além dos *core developers*, tem acesso de *commit* ao repositório de código os *contributing developers* (desenvolvedores que no passado foram *core developers*) e os *guest committers* (desenvolvedores com acesso temporário ao repositório geralmente para o desenvolvimento de algum recurso específico)<sup>16</sup>. Em maio de 2014, esse grupo era composto por trinta e dois desenvolvedores com acesso ao repositório de código oficial. Sendo que destes vinte e dois enviaram alguma contribuição no último ano.

Somente este grupo possui acesso de escrita ao repositório de código. Os demais membros da comunidade se quiserem fazer alguma alteração no código do WordPress devem enviar uma sugestão de modificação criando um novo *ticket* no Trac, um dos sistemas utilizados que será explicado mais adiante.

O desenvolvimento é orientado por um conjunto de oito filosofias criadas ao longo do tempo e descritas a seguir<sup>17</sup>. Essas filosofias expressam bem os princípios da comunidade e ajudam os desenvolvedores a tomarem decisões na hora de criar novas funcionalidades.

- *Out of the Box*: O software deve ser instalado e configurado em pouco tempo e perguntando ao usuário o menor número possível de perguntas.
- Design para a maioria: A maioria dos usuários do WordPress não são desenvolvedores e não tem conhecimento técnico. O software deve ser desenvolvido tendo eles como público-alvo.
- Decisões ao invés de opções: Considerando que a maioria dos usuários não possuem conhecimento técnico, ao invés de criar muitas opções para possibilitar

---

16 Para mais informações sobre a organização do projeto veja a página *Project Organization*. Disponível em: <http://make.wordpress.org/core/handbook/project-organization/>. Acesso em: 28 dez. 2014.

17 Uma descrição mais detalhada das filosofias pode ser encontrada no link <http://make.wordpress.org/core/handbook/our-philosophies/>. Acesso em: 28 dez. 2014.

que o software resolva diferentes cenários, os desenvolvedores devem tomar decisões e implementar o que será usado pela maioria. Opções devem ser delegadas à plugins.

- *Clean, Lean, and Mean*: O WordPress deve oferecer um sólido conjunto de funcionalidades básicas. Deve ser enxuto e rápido. A regra é oferecer funcionalidades que 80% ou mais dos usuários irão utilizar. O restante deve ser implementado na forma de plugins.
- Busca por simplicidade: O objetivo é tornar o software mais simples de usar a cada novo *release*.
- Prazos são flexíveis: Prazos são importantes mas devem ser tomados como absolutos. Uma nova versão do WordPress será publicada quando estiver pronta. Não é um problema estender um prazo definido se necessário.
- A minoria ativa: As pessoas que costumam se manifestar nos fóruns da comunidade representam apenas uma pequena fração dos usuários do WordPress. Não se deve tomar decisões ouvindo apenas esta minoria ativa. É necessário buscar ouvir a opinião dos demais usuários através do contato com eles em WordCamps e em outras instâncias.
- Nossa Declaração dos Direitos dos Cidadãos: As quatro liberdades da GPLv2 (direito de executar o programa para qualquer fim, estudar o código, redistribuí-lo e distribuir cópias modificadas) são consideradas como a Declaração dos Direitos dos Cidadãos do WordPress.

### 3.3. Comunicação

A comunidade do WordPress utiliza três principais canais de comunicação pela internet para além dos encontros presenciais que ocorrem nos WordCamps e uma vez por ano no WordPress Community Summit. Todos esses canais são abertos para qualquer pessoa que queira participar.

Para toda comunicação relacionada diretamente com código é utilizado o Trac (<https://core.trac.wordpress.org/>), um software livre escrito em Python para o controle de mudanças em um projeto de desenvolvimento. Ele é utilizado tanto para o relato e resolução de *bugs*, quanto para a discussão e desenvolvimento de novos recursos.

Através dele é possível navegar por todo histórico de *commits* do projeto. Cada commit feito é associado a um *ticket* criado dentro do Trac, onde é possível acompanhar a discussão realizada em torno da funcionalidade ou *bug* em questão.

Conforme comentado anteriormente, como somente um grupo fechado de desenvolvedores possui acesso de escrita ao repositório, é através do Trac que outros membros da comunidade podem contribuir com sugestões de código para o WP. Isso pode ser feito por qualquer pessoa que deve criar um novo *ticket* descrevendo o problema que ele pretende solucionar ou então a funcionalidade nova que gostaria de ver implementada. Se desejar, a pessoa pode enviar junto um *patch* com as alterações sugeridas. Criado o *ticket*, ele é então discutido e pode ser aprovado ou não por um dos *core developers*.

O segundo meio de comunicação utilizado pela comunidade é o canal #wordpress-dev no servidor de IRC [irc.freenode.net](http://irc.freenode.net). É aqui que ocorre a maior parte da comunicação em tempo real. Toda semana ocorre nele uma reunião de desenvolvimento, além de outras reuniões de times trabalhando em questões específicas. A maior parte dos *core developers* fica online neste canal enquanto está trabalhando e em muitos momentos ele é utilizado para a solução de questões pontuais. O histórico do canal é público e aberto para qualquer um que queira consultá-lo através do endereço <http://irclogs.wordpress.org/>.

Para comunicação assíncrona, e que não é diretamente relacionada com código, é utilizado uma instalação do próprio WordPress com um tema chamado p2 que foi criado justamente para este fim. Seu funcionamento é semelhante ao de um fórum de discussão online. Este site, disponível no endereço <http://make.wordpress.org/>, é dividido em onze grandes grupos que cuidam de partes distintas do ecossistema do projeto. São eles:

- *Core*: Dedicado ao desenvolvimento do WordPress. É aqui que são publicados as pautas das reuniões semanais de desenvolvimento e as metas de um novo *release*. Também é comum discussões maiores sobre o futuro de algum recurso ou a reorganização de uma parte do código começarem neste canal ao invés do Trac.
- *UI*: Discussão sobre design e usabilidade do projeto.



- **Acessibilidade:** Discussões sobre temas ligados à acessibilidade do software.
- **Dispositivos móveis:** Cuida de um aplicativo do WordPress para celulares Android, iOS, Windows e Blackberry.
- **Políglotas:** Coordenação dos trabalhos de tradução do software.
- **Suporte:** Discussão sobre as ferramentas de suporte e atendimento à comunidade.
- **Documentação:** Discussão sobre a documentação do WordPress.
- **Temas:** Todos os temas que são criados pela comunidade e distribuídos no repositório oficial de temas são aprovados por um grupo de revisores. Este grupo é utilizado para discussões sobre o processo de revisão de temas.
- **Plugins:** O mesmo que o grupo de temas mas dedicado aos plugins.
- **Comunidade:** Discussão sobre a organização de WordCamps e outros eventos da comunidade.
- **Meta:** Discussão voltada para o site wordpress.org.

Além dos três canais descritos acima, existe também uma lista de discussão por e-mail. Esta lista foi o principal canal de comunicação nos primeiros anos do projeto. Porém, com o passar do tempo, ela perdeu importância e hoje é um canal secundário.

### **3.4. Ciclo de desenvolvimento de uma nova versão**

Existem dois tipos de versões do WordPress: *minor version* e *major version*. Uma *minor version* é uma versão com correções de problemas ou falhas de segurança em uma *major version* publicada anteriormente. Este tipo de versão não tem uma periodicidade definida e fica a cargo do desenvolvedor responsável pela última *major version* publicada.

Uma *major version* do WordPress é publicada mais ou menos a cada quatro meses. Seu ciclo de desenvolvimento começa com a escolha do coordenador da versão dentro do grupo de *core developers*. Escolhido o coordenador da versão, é marcada uma reunião no canal no IRC para discutir quais os novos recursos serão desenvolvidos e por

quem. São criados times e definido um líder para cada time de cada novo recurso que entrará na próxima versão. Nesta reunião também é definido um calendário para o desenvolvimento.

Após essa reunião inicial, o desenvolvimento começa. Cada time trabalha separadamente e reuniões semanais ocorrem, também no canal no IRC, para que todos, e especialmente o coordenador da versão, possam ter uma visão global do progresso.

Ao fim do ciclo de desenvolvimento, é lançado um primeiro beta da futura versão e pede-se para que a comunidade teste e relate possíveis problemas encontrados. A partir deste momento não é mais possível adicionar novos recursos à versão em desenvolvimento e somente são aceitas correções de problemas. Se necessário, novos betas são lançados até que exista um consenso de que a versão é estável. Neste momento é lançado o primeiro *release candidate* para que seja feita uma última verificação. É nesse ponto que os times de tradução começam a trabalhar para traduzir o WordPress para seus idiomas. Por fim é publicada a nova versão.

### **3.5. Dados quantitativos**

Nesta seção serão apresentados os dados obtidos a partir da análise do repositório de código do WordPress. Bem como a análise posterior feita a partir da lista de desenvolvedores extraída do repositório para determinar sua localização.

#### **3.5.1 Análise do repositório de código**

Para simplificar, serão chamados de *core developers* o grupo de desenvolvedores que possui acesso de escrita ao repositório de código. Ao todo foram analisados 25.692 *commits* realizados por 32 *core developers* distintos com acesso direto ao repositório. Sendo que 22 destes desenvolvedores realizaram pelo menos uma contribuição no último ano. O desenvolvedor mais ativo realizou 7813 modificações no repositório, enquanto o menos ativo realizou 2 modificações.

A figura 1 mostra a distribuição cumulativa de contribuições por desenvolvedor com acesso de escrita ao repositório ao longo de toda a história do software. Foi utilizada a escala logarítmica para o eixo X de todos os gráficos apresentados neste trabalho para seguir o padrão dos gráficos apresentados por MOCKUS *et al.* (2002). As

diferentes linhas indicam a fração acumulada de *commits* realizados, de linhas adicionadas e de linhas removidas. Como pode ser visto, 9 desenvolvedores são responsáveis por mais de 80% das contribuições tanto em número de *commits*, quanto linhas adicionadas e linhas removidas.

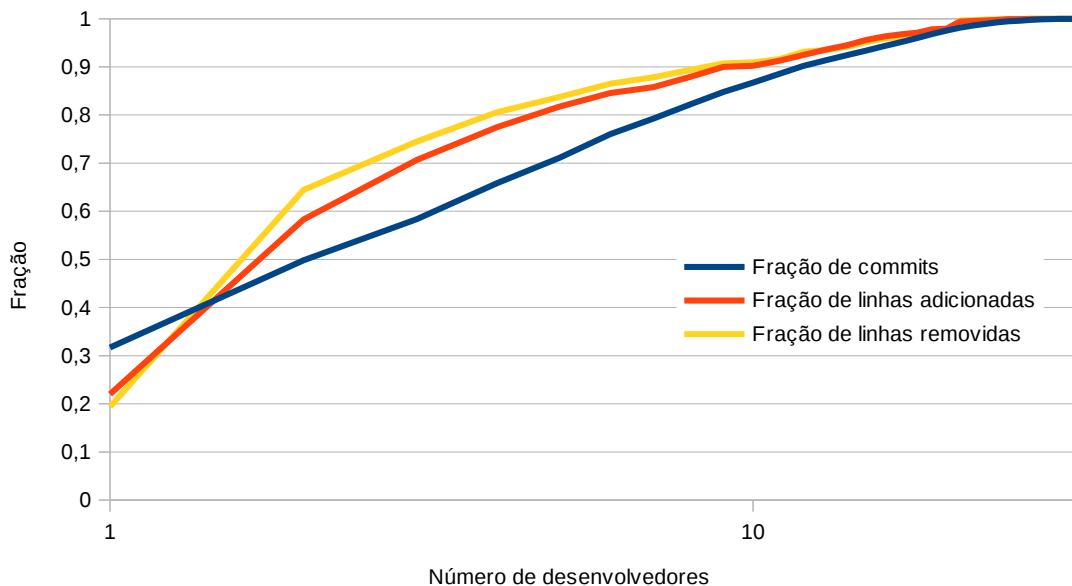


Figura 1: Distribuição cumulativa de contribuições ao repositório de código realizadas pelos desenvolvedores com acesso de escrita

Já a figura 2 apresenta a distribuição cumulativa das frações dos *commits*, linhas adicionadas e linhas removidas porém considerando um universo maior de desenvolvedores. Nela são considerados todos os desenvolvedores da comunidade, inclusive aqueles que não tem acesso de escrita ao repositório de código e, por isso, realizaram suas contribuições através de um *patch* que foi enviado ao Trac e aprovado por um *core developer*. Como foi discutido na metodologia, para conseguir esse dado foi necessário analisar as mensagens de cada um dos *commits*.

Neste segundo cenário, o número total de desenvolvedores é de 1378, sendo 32 *core developers* e 1346 desenvolvedores que fazem parte da comunidade. Deste último grupo, 767 realizaram uma contribuição e 368 realizaram duas contribuições. Considerando apenas as contribuições próprias, o *core developer* mais ativo realizou 4508 *commits*. Os 3305 *commits* restantes foram de contribuições realizadas por outros desenvolvedores que ele revisou e aprovou. Neste caso há uma distribuição maior das

contribuições, 50 desenvolvedores são responsáveis por 80% delas. Se considerarmos todo o conjunto de 25.692 contribuições analisadas, temos que os *core developers* realizaram 14.882 *commits* (ou 58% do total), enquanto os desenvolvedores da comunidade realizaram 10.810 *commits* (ou 42% do total). Somente a versão 4,1, última lançada, contou com a contribuição de 283 desenvolvedores da comunidade.

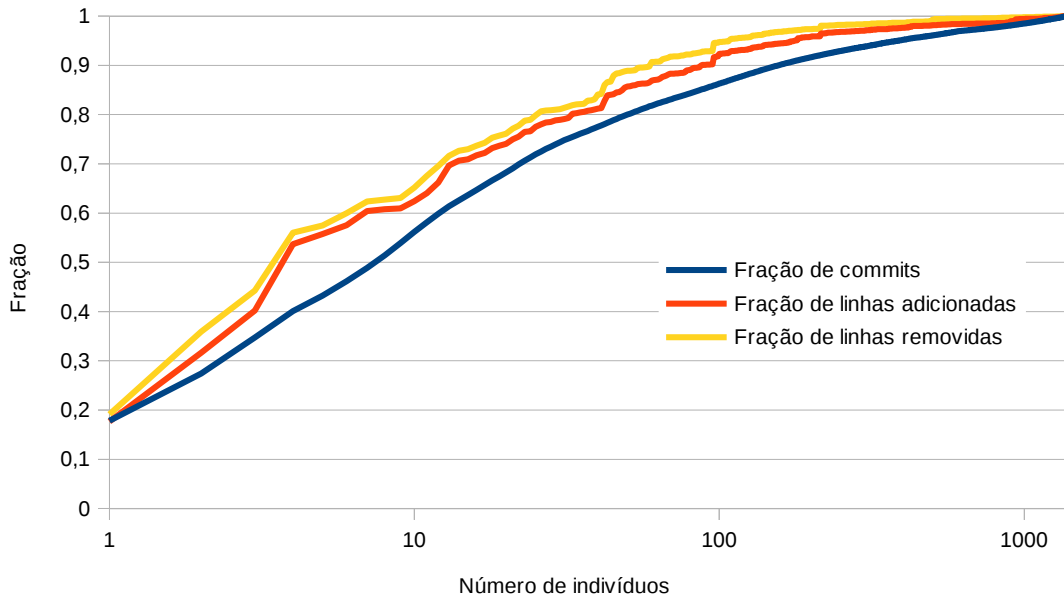


Figura 2: Distribuição cumulativa de contribuições ao repositório de código realizadas por todos os desenvolvedores que fazem parte da comunidade

As próximas duas figuras mostram separadamente as contribuições de todo o universo de desenvolvedores para a criação de novos recursos e para a correção de *bugs*. Essa separação foi criada com o intuito de verificar se há uma participação distinta dos dois grupos (*core developers* e desenvolvedores da comunidade) nestas duas atividades.

A figura 3 mostra a distribuição cumulativa de contribuições de novos recursos. Como pode ser visto, não são todos os desenvolvedores que contribuem com o WordPress que participam criando novos recursos para o software. Do universo de 1378 desenvolvedores, 451 contribuíram com novos recursos. Sendo que destes, 236 indivíduos, ou pouco mais da metade, realizou um *commit*. É importante notar que essa atividade possui uma concentração muito maior de contribuições em um grupo pequeno de desenvolvedores se comparado com a atividade de resolução de problemas. Neste cenário, 16 desenvolvedores foram responsáveis por 80% dos *commits* (enquanto que no

cenário da figura 2, 50 desenvolvedores foram responsáveis por 80% dos *commits*).

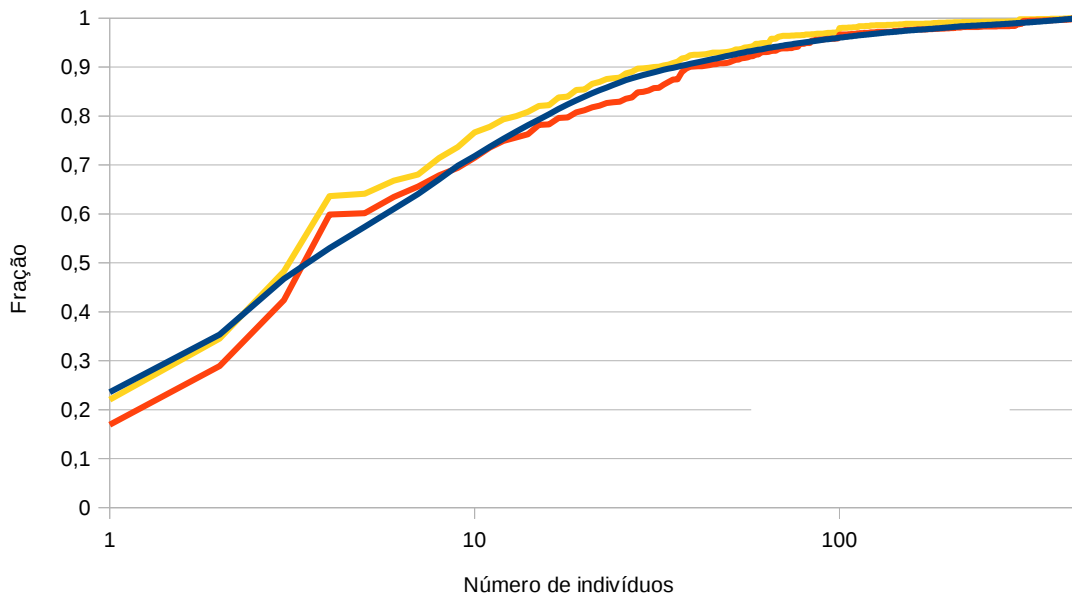


Figura 3: Distribuição cumulativa de contribuições de novos recursos

A figura 4 mostra uma participação muito maior dos desenvolvedores que não possuem acesso de escrita ao repositório de código na correção de bugs. Do universo de 1378 desenvolvedores, 1249 (ou 90%) contribuíram com esta tarefa. Sendo que destes, 719 indivíduos, ou pouco mais da metade, realizaram apenas um *commit*. Comparado com a atividade de criação de novos recursos, a correção de *bugs* é realizada por um número muito maior de desenvolvedores e é também uma atividade com uma concentração menor. Neste caso, 131 desenvolvedores foram responsáveis por 80% dos *commits*.

Como será visto na conclusão, essa diferença de participação entre os *core developers* e os desenvolvedores da comunidade na criação de novos recursos e na correção de *bugs* é semelhante ao que foi verificado no estudo do Apache e Firefox. Parece natural que um grupo menor de desenvolvedores esteja envolvido com a atividade de criação de novos recursos. Já que esta tarefa tipicamente requer um investimento maior de tempo e um conhecimento maior do software como um todo e de sua cultura de desenvolvimento. Enquanto que a atividade de correção de *bugs* é mais pontual e costuma requerer um esforço menor.

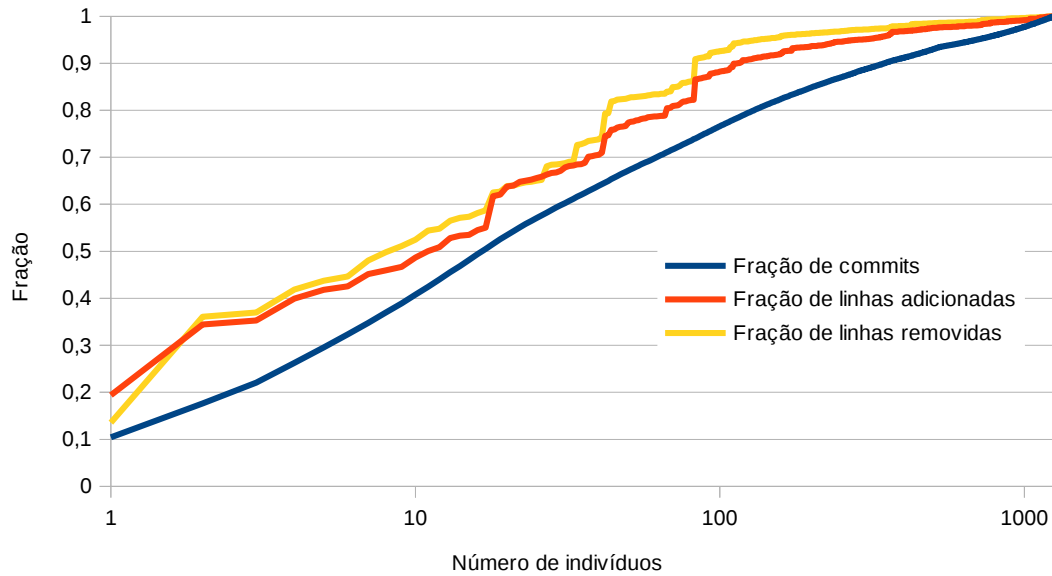


Figura 4: Distribuição cumulativa de correções de bugs

### 3.5.2 Localização dos desenvolvedores

Foi possível encontrar o país onde vivem todos os 32 *core developers* do WordPress. Em relação aos demais desenvolvedores, do universo total de 1378 foi possível determinar o país de residência de 603, ou 43% da amostra.

Em relação aos *core developers*, a figura 5 mostra seu país de residência. Mais da metade deles, 18 desenvolvedores, residem nos Estados Unidos, 5 estão no Reino Unido, 3 no Canadá e os demais 6 desenvolvedores estão distribuídos em 6 países distintos.

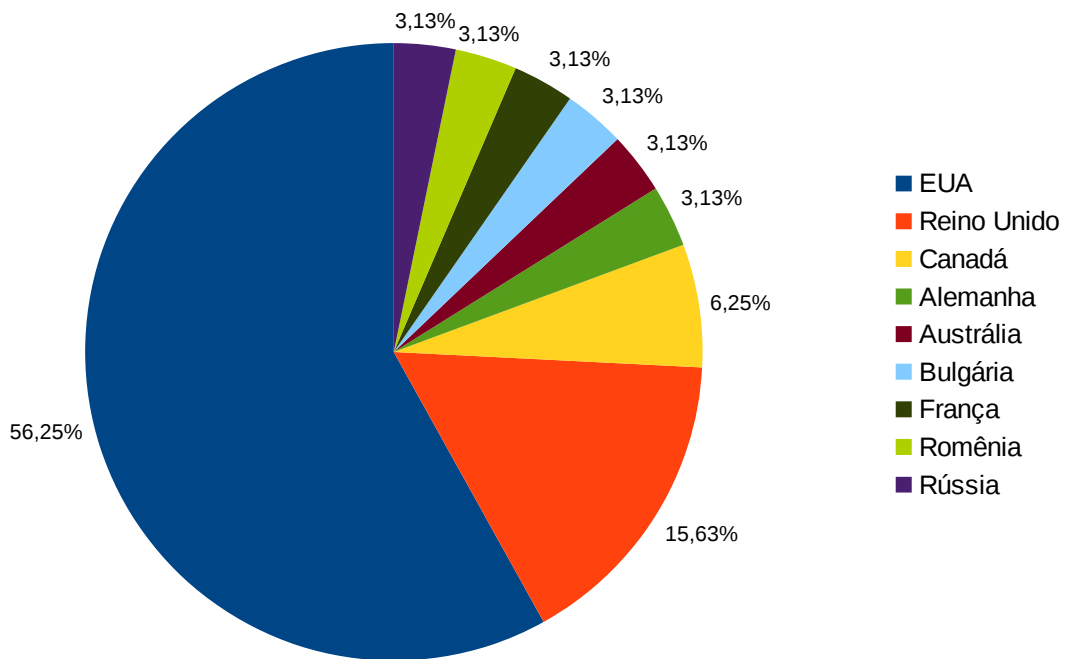


Figura 5: Distribuição dos core developers por país

Já a figura 6 apresenta a mesma informação porém agrupada por continente.

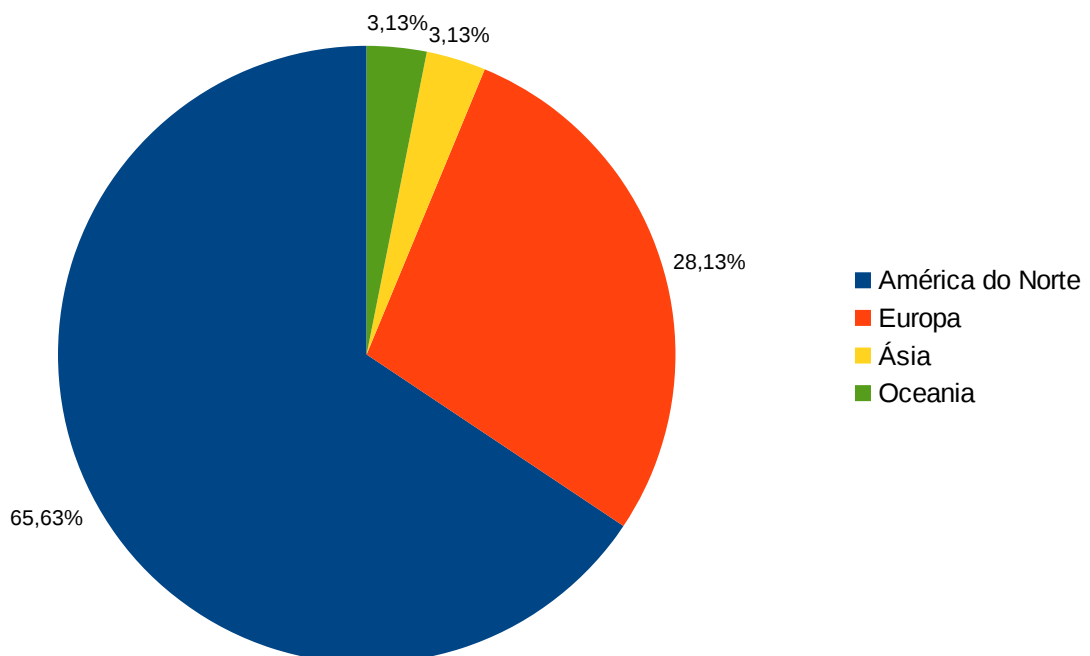


Figura 6: Distribuição dos core developers por continente

Os dois gráficos deixam claro que o grupo que tem acesso de escrita ao código do WP está concentrado apenas na América do Norte e na Europa. Com a exceção de um desenvolvedor na Ásia e outro na Oceania. De maneira semelhante ao que acontece com a distribuição dos funcionários da Automattic citada no prefácio, não existem *core developers* na América Latina ou na África.

A figura 7 mostra a distribuição dos *core developers* considerando o principal idioma falado em seu país de residência. Vale destacar que foi utilizado o idioma mais falado nos casos de países com mais de um idioma. Aqui a concentração é ainda maior, das 32 pessoas, 27 moram em países onde o principal idioma é o inglês.

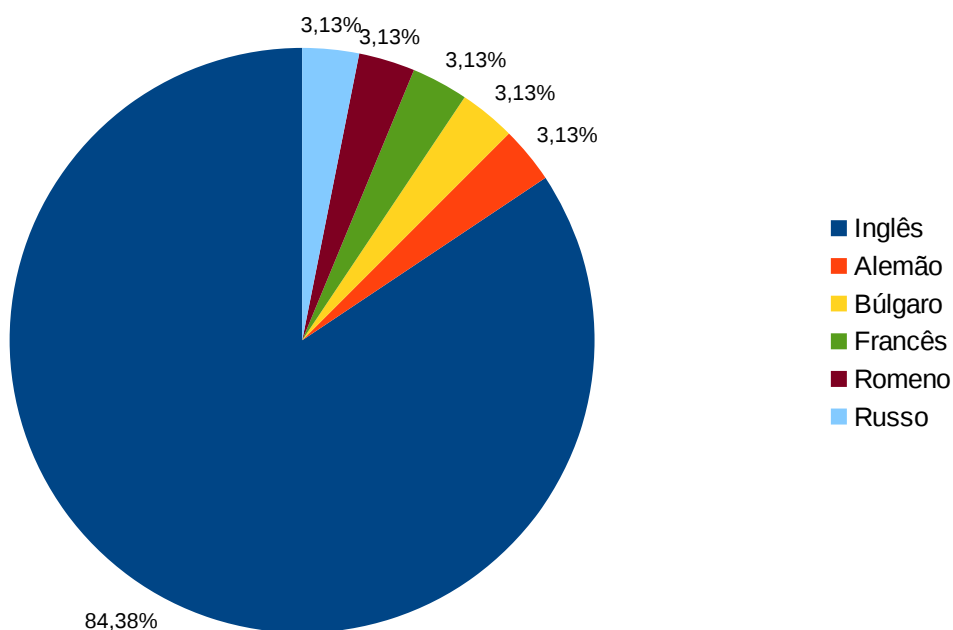


Figura 7: Principal idioma falado no país de residência dos *core developers*

Ao considerarmos o universo de 603 desenvolvedores que já contribuíram com código para o WordPress, a distribuição geográfica por países é semelhante à distribuição dos *core developers*. Como pode ser visto na figura 8, praticamente metade (298 pessoas) está nos Estados Unidos, aproximadamente 10% (55 pessoas) no Reino Unido e 5% (32 pessoas) no Canadá. Os 35% restantes estão distribuídos em outros 54 países. Sendo que em 19 países reside apenas 1 desenvolvedor e em 11 apenas 2 desenvolvedores. Neste universo foram encontrados 5 desenvolvedores que moram no Brasil.



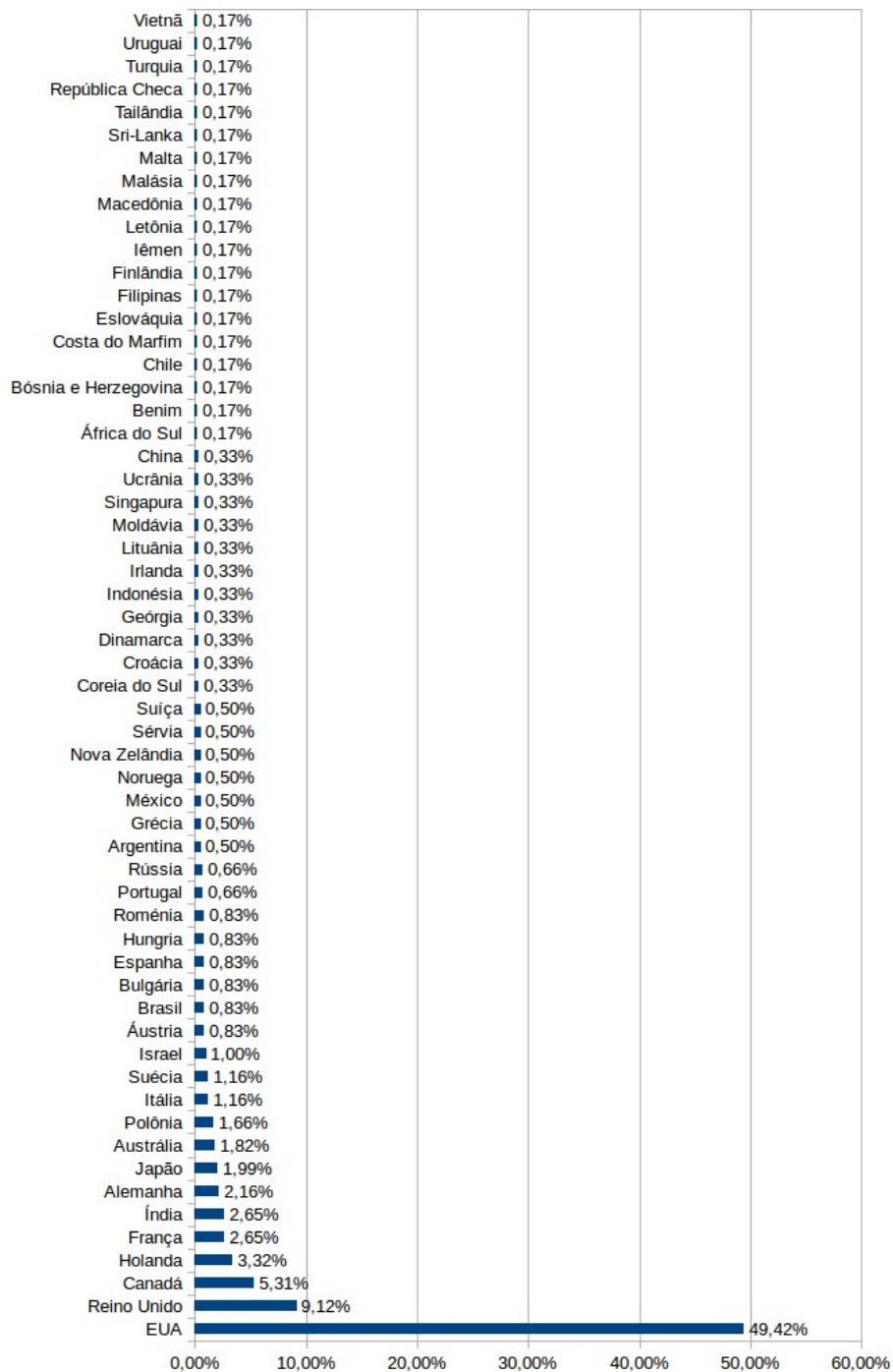
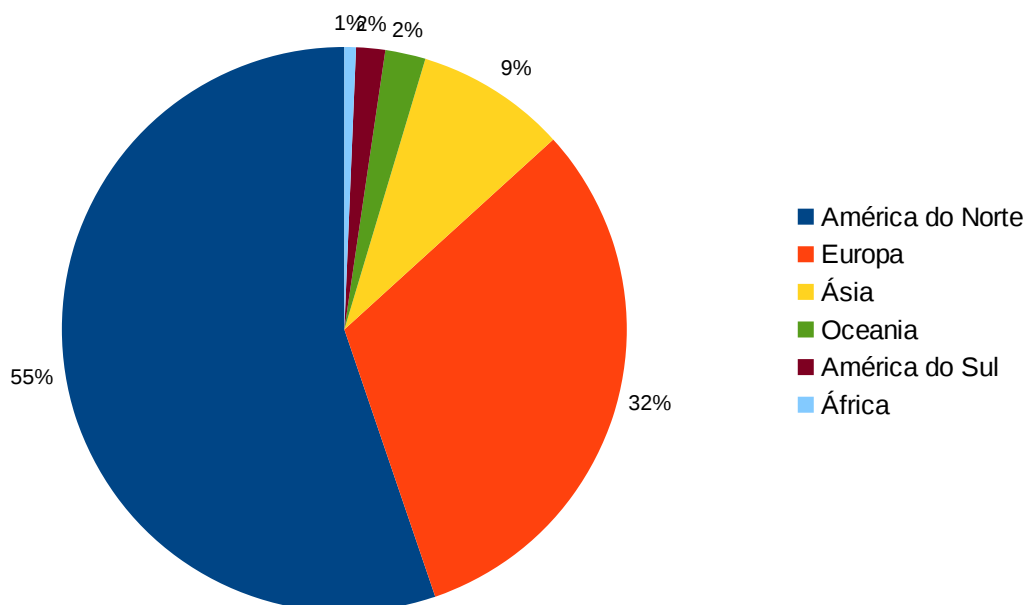


Figura 8: Distribuição dos desenvolvedores por país

A figura 9 mostra o mesmo conjunto de desenvolvedores agrupados por continente. Mais da metade está na América do Norte e um terço na Europa. África e América do Sul que não aparecem no mesmo gráfico para os *core developers* aparecem mas com um percentual bem pequeno.



*Figura 9: Distribuição dos desenvolvedores por continente*

Por fim a figura 10 mostra o idioma falado no país de residência dos desenvolvedores do WP. Novamente pode ser visto um domínio do inglês com 67%, sendo que o segundo idioma, o alemão, possui apenas 3,48%.

Durante a coleta de dados uma curiosidade reforça os indícios de que existe uma concentração da comunidade em torno dos países que falam inglês, em especial os Estados Unidos. Como dito anteriormente o campo com o local de residência dos desenvolvedores é um campo aberto, onde é possível preencher qualquer informação. Dos 298 desenvolvedores dos EUA apenas 43 indicaram seu país de origem. A grande maioria colocou a cidade e o estado e um número menor apenas a cidade ou então o código postal. Enquanto que os desenvolvedores dos demais países, de um universo de 305 pessoas, apenas 47 não indicaram qual o seu país de residência.

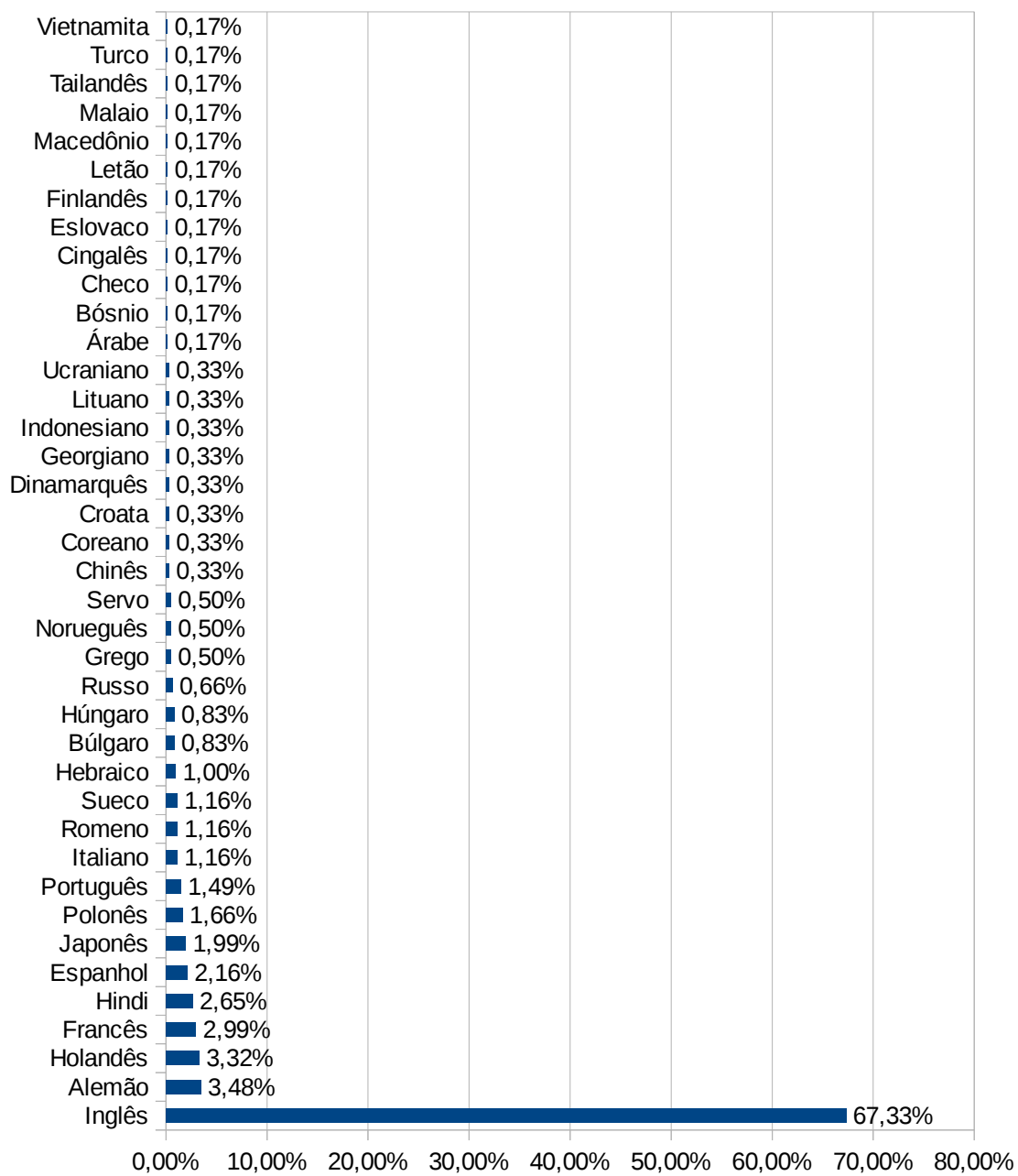


Figura 10: Principal idioma falado no país de residência dos desenvolvedores

## 4. CONCLUSÕES

O processo de criação deste trabalho parece confirmar a hipótese de que, por sua natureza aberta e descentralizada, projetos de software livre tendem a ser um bom terreno para serem explorados por pesquisadores da área de Engenharia de Software. Não houve nenhuma grande dificuldade em conseguir as informações necessárias para atingir os objetivos propostos. É claro que em muitos casos apenas o dado bruto estava disponível. Mas isso não chegou a ser um obstáculo significativo, como no caso do repositório de código, já que graças a outro projeto de software livre, o *gitinspector*, foi possível extrair as informações necessárias.

Já em relação as características do processo de desenvolvimento do WordPress, MOCKUS *et al.* (2002) ao final de seu trabalho sobre o Apache e o Firefox, levantam um conjunto de sete hipóteses com base nas características observadas nestes dois softwares. Destas sete hipóteses, cinco não podem ser analisadas por este estudo, pois os dados relacionados à elas não foram levantados. As duas restantes serão discutidas a seguir.

*Hipótese 1: Um software livre sempre terá um grupo de no máximo dez a quinze desenvolvedores responsáveis por pelo menos 80% das contribuições.*

Aqui fica aberta a possibilidade para trabalhos futuros investigarem o porque dessa diferença grande entre o que foi encontrado no caso do Firefox e do Apache e o que foi encontrado neste estudo em relação ao WordPress. Já que para este último software um grupo de cinquenta desenvolvedores são responsáveis por 80% das contribuições. Um número significativamente maior do que os quinze sugeridos pelo trabalho anterior. Os autores ao elaborarem essa hipótese partiram do princípio de que não poderiam haver mais de dez a quinze pessoas envolvidas com a maior parte do código de um único projeto pois isso dificultaria a gestão do desenvolvimento. Provavelmente influencia nessa diferença o fato de o WordPress ter um número muito maior de desenvolvedores participando do software. Enquanto para o Firefox foram identificadas cerca de 200 pessoas que enviaram contribuições e para o Apache cerca de 400, no caso do WordPress foram identificadas cerca de 1400 pessoas.

*Hipótese 2: Em um projeto de software livre um grupo maior por uma ordem de*

*grandeza que os desenvolvedores principais será responsável por corrigir bugs e outro grupo outra ordem de grandeza maior será responsável por relatar problemas.*

Parte dessa hipótese é confirmada pelo presente estudo. No WordPress um grupo de 16 desenvolvedores foi responsável por 80% dos *commits* de novos recursos, enquanto um grupo de 130 desenvolvedores foi responsável por 80% das correções de problemas. Como não foram levantados dados acerca do número de pessoas que reportam problemas, esta parte da hipótese não pôde ser verificada.

Em relação a quarta pergunta, específica deste trabalho, a coleta de dados indica que, pelo menos no caso do WordPress, é possível extrapolar a discussão dos Estudos CTS sobre a universalidade da ciência como um particular no poder para o software livre. No discurso o WP é apresentado como construído por uma comunidade global que está aberta a participação de todos independente de onde estejam. Porém, no que diz respeito aos desenvolvedores, viu-se que eles estão concentrados em apenas uns poucos países e em sua grande maioria tem o inglês como seu primeiro idioma.

Diante dessa diferença entre o discurso e o que se verifica na prática, surge a questão de quais seriam as barreiras que impedem a participação de desenvolvedores de um número maior de países, em especial do Brasil, no processo de construção do WordPress?

Também é necessária uma reflexão sobre as implicações desta concentração de desenvolvedores em uns poucos países para o movimento do software livre pensado na perspectiva de um movimento que dá continuidade a luta dos hackers radicais de ROSZAK pelo controle da informação e pelo uso da tecnologia como um instrumento de participação política.

Espera-se que trabalhos futuros possam utilizar os dados levantados por este trabalho para investigar as perguntas elaboradas acima.

## 5. REFERÊNCIAS

- COALO, J.J. “With 60 Million Websites, WordPress Rules The Web. So Where's The Money?”. In: **Forbes**, 5 setembro 2012. Disponível em:  
<http://www.forbes.com/sites/jjcolao/2012/09/05/the-internets-mother-tongue/>.  
Acesso em: 28 dez. 2014.
- MULLENWEG, Matt. “The Blogging Software Dilemma” In: Matt Mulleneg, 2003.  
Disponível em: <http://ma.tt/2003/01/the-blogging-software-dilemma/>. Acesso em:  
28 dez. 2014.
- MULLENWEG, Matt. **State of the Word 2013**, 2013. Disponível em:  
<http://www.slideshare.net/photomatt/state-of-the-word-2013>. Acesso em: 28 dez.  
2014.
- MOCKUS, Audris; FIELDING, Roy T.; HERBSLEB, James D. “Two case studies of open source software development: Apache and Mozilla”. In: **ACM Transactions on Software Engineering and Methodology (TOSEM)**, v. 11, n. 3, pp. 309-346, 2002.
- ROSZAK, Theodore, "O Computador e a Contracultura". In: **O Culto da Informação**, capítulo 7, Editora Brasiliense, 1988.
- SCACCHI, Walt. “The future of research in free/open source software development”. In: **Proceedings of the FSE/SDP workshop on Future of software engineering research**. ACM, 2010, pp. 315-320.
- WORDPRESS FOUNDATION. **Past WordCamps**. Disponível em:  
<http://central.wordcamp.org/schedule/past-wordcamps/>. Acesso em: 28 dez. 2014.
- WORDPRESS FOUNDATION. **WordPress History**. Disponível em:  
<https://codex.wordpress.org/History>. Acesso em: 28 dez. 2014.
- WORDPRESS FOUNDATION. **WordPress Handbook**. Disponível em:  
<http://make.wordpress.org/core/handbook/>. Acesso em: 28 dez. 2014.

WORDPRESS FOUNDATION. **WordPress Requirements**. Disponível em:  
<http://wordpress.org/about/requirements/>. Acesso em: 28 dez. 2014.