



UFG

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
FACULDADE DE CIÊNCIAS SOCIAIS (FCS)
PROGRAMA DE PÓS-GRADUAÇÃO EM SOCIOLOGIA (PPGS)

MATHEUS GUIMARÃES MELLO

**Cultura do trabalho
de desenvolvedores de software livre**

GOIÂNIA
2024



UNIVERSIDADE FEDERAL DE GOIÁS
FACULDADE DE CIÊNCIAS SOCIAIS

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a [Lei 9.610/98](#), o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

Dissertação Tese Outro*: _____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

Exemplos: Estudo de caso ou Revisão sistemática ou outros formatos.

2. Nome completo do autor

Matheus Guimarães Mello

3. Título do trabalho

Cultura do trabalho de desenvolvedores de software livre

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento SIM NÃO¹

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:

a) consulta ao(a) autor(a) e ao(a) orientador(a);

b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.



Documento assinado eletronicamente por **Jordao Horta Nunes, Professor do Magistério Superior**, em 13/08/2024, às 11:02, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Matheus Guimarães Mello, Usuário Externo**, em 13/08/2024, às 13:53, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4740559** e o código CRC **E58C1D9E**.

Referência: Processo nº 23070.040553/2024-52

SEI nº 4740559

MATHEUS GUIMARÃES MELLO

**Cultura do trabalho
de desenvolvedores de software livre**

Dissertação apresentada ao Programa de Pós-Graduação em Sociologia, da Faculdade de Ciências Sociais, da Universidade Federal de Goiás (UFG) como requisito para obtenção do título de Mestre em Sociologia.

Área de Concentração: Sociedade, Política e Cultura

Linha de Pesquisa: Trabalho, Formação e Representações Culturais

Orientador: Prof. Dr. Jordão Horta Nunes

GOIÂNIA
2024

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Mello, Matheus Guimarães
Cultura do trabalho de desenvolvedores de software livre
[manuscrito] / Matheus Guimarães Mello. - 2024.
149 f.: il.

Orientador: Prof. Dr. Jordão Horta Nunes.
Dissertação (Mestrado) - Universidade Federal de Goiás,
Faculdade de Ciências Sociais (FCS), Programa de Pós-Graduação em
Sociologia, Goiânia, 2024.

Bibliografia. Apêndice.

Inclui siglas, abreviaturas, gráfico, tabelas, lista de figuras, lista de tabelas.

1. Cultura do trabalho. 2. Desenvolvedores de Software. 3. Software Livre. 4. Identidade Ocupacional. 5. Trabalho em Tecnologia da Informação. I. Nunes, Jordão Horta, orient. II. Título.

CDU 316



UNIVERSIDADE FEDERAL DE GOIÁS
FACULDADE DE CIÊNCIAS SOCIAIS
ATA DE DEFESA DE DISSERTAÇÃO

Ata nº 27 da sessão de Defesa de Dissertação do **MATHEUS GUIMARÃES MELLO**, que confere o título de Mestre em Sociologia, na área de concentração em **Sociedade, Política e Cultura**.

Aos sete de julho de dois mil e dezesseis, a partir das 14h, na Sala de Defesas da Faculdade de Ciências Sociais realizou-se a sessão pública de Defesa de Dissertação intitulada **CULTURA DO TRABALHO DE DESENVOLVEDORES DE SOFTWARE LIVRE**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Jordão Horta Nunes (UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Cleito Pereira dos Santos (UFG) membro titular interno e Professora Doutora Maria Aparecida da Cruz Bridi (UFPR), membra titular externa. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão reservada a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Jordão Horta Nunes, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora.

TÍTULO SUGERIDO PELA BANCA



Documento assinado eletronicamente por **Jordao Horta Nunes, Professor do Magistério Superior**, em 23/09/2024, às 11:21, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Cleito Pereira Dos Santos, Professor do Magistério Superior**, em 24/09/2024, às 21:14, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Flavio Munhoz Sofiati, Coordenador de Pós-Graduação**, em 26/09/2024, às 08:35, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4845880** e o código CRC **91DBF9F4**.

Referência: Processo nº 23070.040553/2024-52

SEI nº 4845880

Agradecimentos

Agradeço a todas as pessoas que cederam tempo para participar da pesquisa com o questionário e com as entrevistas;

A Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior pela concessão da Bolsa CAPES Demanda Social;

A minha família, Mirene, José Divino, Pedro Henrique pelo apoio incondicional;

A Sandra Regina, pela companhia e carinho;

Aos amigos e amigas: Ariel, Elis, Gêssica, Lúbia, Narrira, Mario, Michelle, Tatiele, Wildes e quantas mais;

Ao meu orientador Jordão pela grande paciência e amizade;

A Jociane Lessa pelas palavras de sabedoria;

Aos membros da banca de qualificação e da defesa pelas valiosas indicações;

Também todos os professores e professoras que me fizeram chegar até aqui;

A Richard Stallman, por criar e levar adiante o *software* livre;

Ao grupo Libre Planet São Paulo, pela paciência e receptividade de acolher um cientista social entre *hackers*;

A todos e todas desenvolvedoras de *software* livre que me permitiram utilizar um sistema operacional e ferramentas de pesquisa inteiramente livres;

E no mais, ainda outras incontáveis pessoas, por isso, acrescento que se estiver lendo isso, agradeço a você.

Resumo

Esta pesquisa busca compreender as possíveis relações entre uma identidade (construída a partir de uma ideologia de compartilhamento) e uma sociabilidade (um contínuo produto de socialização e trajetórias de vida) consideradas *hacker* e a cultura do trabalho (ou arranjos organizacionais) de desenvolvedores de *software* livre. O pressuposto é que a expansão das Tecnologias de Informação (TI) nas últimas duas décadas não deve ser descontextualizada da recente reestruturação do capitalismo e suas consequências. Um processo contínuo importante é a precarização das relações de trabalho, através do qual empresas em todo o mundo valorizam contratos de trabalho flexíveis, projetos de curto prazo e terceirizações. Neste contexto, as organizações de TI desempenham um papel fundamental, já que elas lidam com as informações para as novas formas de gestão do trabalho. Ao mesmo tempo, o trabalho em TI é, desde o seu surgimento, adaptado para o regime de flexibilidade. No entanto, deve-se considerar também que o desenvolvimento de *Software Livre* é inseparável da sua ética *hacker* original. A hipótese é que os valores de compartilhamento de conhecimentos e de trabalho em colaboração anônima da cultura *hacker* são significativos para compreender a relação entre os processos contínuos de socialização dos desenvolvedores e a cultura organizacional da empresa/fundação. O estudo se concentra sobre a existência de conflitos, negociações e adaptações entre os valores anteriores dos desenvolvedores e a maneira em que seu trabalho coletivo é organizado. Assim, o objetivo é entender a tecnologia, a cultura e o trabalho como um todo, levando em conta tanto a influência das características técnicas sobre a sociabilidade quanto o enfoque sobre como os processos de socialização das pessoas influencia sobre a maneira como o trabalho de desenvolver tecnologia é organizado.

Palavras-chave: Cultura do trabalho. Desenvolvedores de Software. Software Livre. Identidade Ocupacional. Trabalho em Tecnologia da Informação.

Abstract

This research aims to understand the possible relationship between hacker ethics and its sociability (a continuous product of socialization and life trajectories) with organizational culture (or *culture du travail* in French sociology literature) of Free and Open Source Software (FOSS) developers. It is based on the assumption that the expansion of the Information Technologies (IT) in the last two decades should not be de-contextualized from recent capitalism's restructuring of and its consequences. The increase of precarity in labour relations is a major ongoing process, by which companies throughout the world emphasizes flexible work contracts, short-term projects and outsourcing. In this context, IT organizations play labour relations a pivotal role, as they handle the information for the new forms of work managing. At the same time, working in IT is, since its emergence, adapted to the flexibility scheme. Nevertheless, the development of FOSS is inseparable from its original hacker ethics. The hypothesis is that the values of sharing knowledge and peer working of the hacker culture is significant to understand the relationship between developers' continuous processes of socialisation (identities formation) and the organizational culture of the company/foundation. The study will focus whether there are conflicts, negotiations and adaptations between the developers' previous values and the manner they arrange their collective work. Thereby, the goal is to understand technology, culture and work as a whole. To this end, semi-structured interviews (during an exploratory and a complementary stages) and a virtual survey among Brazilian FOSS developers will be conducted. These will consider the life course (and socialisation) as the main analytical unit, since it highlights the people who embody a culture of sharing and also stands behind the technique.

Key-words: Free Software. Organizational culture. Software development. Occupational identity. Information technology work.

English (en)

Synopsis

Iste recerca visa comprehendere le relationes possibile inter un identitate hacker (formate ex processos de socialisation e trajetorias de vita) e le cultura del travalio de developpatores de software libere e codice aperte. Le presupposition es que le expansion del Technologies de Information (TI) in le ultime duo decennios non debe esser decontextualisate ab le restructuration recente del capitalismo e su consequentias. Un processo importante es le precarisation del relationes de labor, per le quales companias in tote le mundo favori contractos de travalio flexibile, projectos de curte termino e externalisationes (outsourcing). In iste contexto, le organisationes de TI representa un rolo fundamental, post que illos tracta le informationes pro le nove formas de gestion de travalio. Al mesme tempore, le travalio in TI es, desde su principio, adaptate al regimento de flexibilitate. Non obstante, on debe anque considerar que le developpamento de software libere es inseparabile de su ethica hacker original. Le hypothese es que le valores del compartimento de cognoscimentos e de travalio in collaboration del cultura hacker son significative pro comprehendere le relation inter le processos continue de socialisation del developpatores e le cultura organisational del compania. Iste studio se concentra super le existentia de conflictos, negotiationes e adaptationes inter le valores anterior del developpatores e le maniera in le qual su travalio collective se organisa. Pro iste fin, le recerca include interviews semistructurate e un survey virtual inter developpatores brasilian de software libere. Assi, le objectivo es comprender le tecnologia, le cultura e le travalio como un toto, considerante tanto le influencia del characteristics technic super le sociabilitate quanto le foco in le maniera como le travalio de developpar tecnologia es organisate.

Vocabulos-clave: Software libere. Developpatores de software. Cultura del travalio. Identitate occupational. Travalio in Tecnologia de Information.

interlingua (ia)

Lista de ilustrações

Figura 1 – Estrutura organizacional de uma empresa de TI.....	43
Figura 2 – Tirinha humorística sobre o papel das ocupações envolvidas na engenharia de software.....	44

Lista de gráficos

Gráfico 1 – Variação de rendimentos por sexo e grupos ocupacionais do setor de TI: 2003 a 2014 – Brasil.....	52
Gráfico 2 – Distribuição de faixas etárias entre desenvolvedores(as) de <i>software</i> (exceto analistas de sistemas).....	63

Lista de quadros

Quadro 1 – Síntese dos rótulos tipificados para a análise de componentes de motivações entre desenvolvedores de <i>software</i>	100
Quadro 2 – Síntese dos rótulos tipificados para a análise de componentes de cultura do trabalho entre desenvolvedores de <i>software</i>	112

Lista de tabelas

Tabela 1 – Ocupações na Família “Desenvolvedores e Analistas de Programas e Aplicativos (Software) e Multimídia”.....	41
Tabela 2 – Número de Trabalhos por Ocupação.....	42
Tabela 3 – Ocupações e Escolaridade.....	45
Tabela 4 – Escolaridade de Desenvolvedores de <i>Software</i>	47
Tabela 5 – Ocupações e Posição na Ocupação.....	48
Tabela 6 – Rendimento de desenvolvedores de <i>software</i> (exceto analistas de sistemas) em Reais e Salários Mínimos de Julho de 2010.....	49
Tabela 7 – Rendimento de desenvolvedores de <i>software</i> (exceto analistas de sistemas) em reais e salários-mínimos de julho de 2010 segundo gênero.....	52
Tabela 8 – Desenvolvedores de <i>software</i> (exceto analistas de sistemas) segundo grandes regiões.....	53
Tabela 9 – Desenvolvedores de <i>software</i> (exceto analistas) por cor ou raça agrupada....	54
Tabela 10 – Desenvolvedores de <i>software</i> (exceto analistas de sistemas) por cor ou raça agrupada.....	54
Tabela 11 – Cor ou Raça da População Brasileira por Grande Região Geográfica.....	55
Tabela 12 – Rendimento de desenvolvedores de <i>software</i> (exceto analistas) em Reais e Salários Mínimos de Julho de 2010 segundo cor/raça agrupada.....	56
Tabela 13 – Rendimento de desenvolvedores de <i>software</i> (exceto analistas) por Grande Região Geográfica.....	56
Tabela 14 – Rendimento de desenvolvedores de <i>software</i> (exceto analistas) do Sul e Sudeste em Reais e Salários Mínimos de Julho de 2010 segundo cor/raça agrupada.....	57
Tabela 15 – Escolaridade de desenvolvedores de <i>software</i> (exceto analistas) do Sul e Sudeste segundo cor/raça agrupada.....	58
Tabela 16 – Frequência de desenvolvedores de <i>software</i> (exceto analistas) do Sul e Sudeste em cada ocupação segundo cor ou raça agrupada.....	58
Tabela 17 – Rendimento Médio em Reais de 2010 de desenvolvedores de <i>software</i> (exceto analistas) do Sul e Sudeste em cada ocupação segundo cor ou raça agrupada.....	59

Tabela 18 – Religião agrupada de desenvolvedores(as) de <i>software</i> em comparação com proporção das religiões na população brasileira.....	62
Tabela 19 – Estado Conjugal e Ocupações.....	65
Tabela 20 – Atribuição do primeiro contato com computação.....	66
Tabela 21 – Atribuição da importância para formação profissional.....	67
Tabela 22 – Posição na Ocupação em percentuais comparativos entre Censo 2010 (exceto analistas de sistemas) e Survey 2015.....	71
Tabela 23 – Número de funcionários da empresa em que trabalha.....	72
Tabela 24 – Local de Trabalho.....	72
Tabela 25 – Atividade predominante de desenvolvimento.....	73
Tabela 26 – Atribuição da pressão por resultados no ambiente de trabalho.....	75
Tabela 27 – Envolvimento profissional de desenvolvedores(as) com <i>software</i> livre, <i>software</i> não-livre ou ambos.....	78
Tabela 28 – Categorização de desenvolvedores(as) de <i>software</i> quanto ao envolvimento com <i>software</i> livre ou não-livre.....	78
Tabela 29 – Peso do trabalho com <i>software</i> livre na composição da renda pessoal entre desenvolvedores que trabalham simultaneamente com SLCA e SNL.....	81
Tabela 30 – Tempo relativo de trabalho dedicado ao <i>software</i> livre entre desenvolvedores que trabalham simultaneamente com SLCA e SNL.....	81
Tabela 31 – Posição na ocupação segundo tipo de licença do <i>software</i> desenvolvido.....	82
Tabela 32 – Tipo de relação com ou ausência de chefe ou gerente.....	84
Tabela 33 – Horas por semana de trabalho de desenvolvedores segundo tipo de licença do <i>software</i> desenvolvido.....	85
Tabela 34 – Horas por semana de atividades voluntárias de desenvolvedores segundo tipo de licença do <i>software</i> desenvolvido.....	85
Tabela 35 – Identificação com conceito de <i>software</i> livre, código aberto ou não-identificação entre desenvolvedores de <i>software</i> livre e código aberto.....	86
Tabela 36 – Autoatribuição como <i>hacker</i> ou parte da cultura <i>hacker</i> entre desenvolvedores segundo tipo de licença do <i>software</i> desenvolvido.....	89
Tabela 37 – Definições de <i>hacker</i> entre desenvolvedores de <i>software</i>	89

Tabela 38 – Definições de <i>hacker</i> segundo autoatribuição como <i>hacker</i>	92
Tabela 39 – Interesse pelo ativismo de <i>software</i> livre segundo autoatribuição como <i>hacker</i> ou parte da cultura <i>hacker</i>	94
Tabela 40 – Nível de satisfação com o trabalho atual.....	95
Tabela 41 – Motivações para o trabalho de desenvolvedores de <i>software</i>	96
Tabela 42 – Variância total explicada da análise das variáveis de motivações.....	99
Tabela 43 – Matriz padrão dos escores de componentes da análise de motivações segundo tipo de desenvolvimento de <i>software</i>	99
Tabela 44 – Concordância com o motivo “contribuir com o desenvolvimento da sociedade” segundo tipo de desenvolvimento de <i>software</i>	103
Tabela 45 – Concordância com o motivo “compartilhar meus conhecimentos e habilidades” segundo tipo de desenvolvimento de <i>software</i>	103
Tabela 46 – Concordância com o motivo “Diminuir o poder de grandes companhias de <i>software</i> ” segundo tipo de desenvolvimento de <i>software</i>	105
Tabela 47 – Percepção de elementos da cultura e do ambiente de trabalho entre desenvolvedores de <i>software</i>	108
Tabela 48 – Concordância com o elemento “meu salário varia de acordo com minha produtividade” segundo posição na ocupação.....	110
Tabela 49 – Variância total explicada da análise de variáveis de cultura do trabalho.....	110
Tabela 50 – Matriz padrão dos escores de componentes da análise de cultura do trabalho segundo tipo de desenvolvimento de <i>software</i>	111
Tabela 51 – Concordância com o elemento “predomina um espírito de trabalho em equipe” segundo tipo de desenvolvimento de <i>software</i>	114
Tabela 52 – Concordância com o elemento “ambiente agradável de trabalho” segundo tipo de desenvolvimento de <i>software</i>	115
Tabela 53 – Concordância com o elemento “desenvolvimento direcionado para uso específico de um cliente” segundo tipo de desenvolvimento de <i>software</i>	116
Tabela 54 – Concordância com o elemento “trabalho frequentemente entediante” segundo tipo de desenvolvimento de <i>software</i>	116

Lista de abreviaturas e siglas

BSD – *Berkeley Software Distribution*

CA – Código Aberto*

CBO – Classificação Brasileira de Ocupações

CIUO-08 – *Clasificación Internacional Uniforme de Ocupaciones*, 2008

CLT – Consolidação das Leis do Trabalho

CNPJ – Cadastro Nacional da Pessoa Jurídica

COD – Classificação de Ocupações para Pesquisas Domiciliares

DIEESE – Departamento Intersindical de Estatística e Estudos Socioeconômicos

EUA – Estados Unidos da América

FSF – *Free Software Foundation*

GDB – *GNU Debugger*

GNU – *GNU's Not Unix!*

GPL – *GNU General Public License*

IBGE – Instituto Brasileiro de Geografia e Estatística

IBM – *International Business Machines Corporation*

MIT – *Massachusetts Institute of Technology*

PJ – Pessoa Jurídica

PNAD – Pesquisa Nacional por Amostra de Domicílios

RAIS – Relação Anual de Informações Sociais

SL – *Software Livre**

SLCA – *Software Livre e de Código Aberto**

SNL – *Software Não-Livre**

OSI – *Open Source Initiative*

TI – Tecnologia(s) da Informação

TIC – Tecnologia(s) de Informação e Comunicação

* Essas abreviaturas não possuem uso corrente entre as comunidades de *software* livre e código aberto em português, mas são utilizadas exclusivamente pelo autor com o propósito didático de distinguir arbitrariamente entre os significados atribuídos às mesmas palavras no campo. Em inglês, apesar de também haver uma necessidade controversa, é comum o uso das siglas FOSS ou FLOSS: *Free (Libre) Open Source Software*.

Sumário

1 Apresentação.....	15
1.1 Justificativa.....	16
1.2 Metodologia.....	16
1.3 Estrutura.....	20
2 Contrapontos Teóricos.....	21
2.1 Construções Históricas.....	21
2.1.1 <i>Tecnologia da Informação no Mundo do Trabalho</i>	21
2.1.2 <i>Cultura Hacker e Software Livre</i>	26
2.2 Conflitos Discursivos.....	31
2.3 Cultura do Trabalho.....	34
3 Características Gerais e Contextos.....	39
3.1 Ocupações Correlatas.....	39
3.1.1 <i>Características Ocupacionais</i>	40
3.1.2 <i>Escolaridade</i>	45
3.1.3 <i>Posição na Ocupação</i>	47
3.2 Variáveis Sociológicas Clássicas.....	49
3.2.1 <i>Gênero</i>	50
3.2.2 <i>Região Geográfica</i>	53
3.2.3 <i>Cor/Raça</i>	54
4 Perfis do Desenvolvimento de Software.....	61
4.1 Socialização e Preferências Pessoais.....	61
4.2 Condições de Trabalho.....	69
5 Cultura do Trabalho no Software Livre.....	77
5.1 Singularidades do <i>Software Livre</i>	79
5.2 Identidade <i>Hacker</i>	86
5.3 Motivações e Cultura do Trabalho.....	95
5.3.1 <i>Motivações para o Desenvolvimento de Software</i>	96
5.3.2 <i>Cultura do Trabalho no Desenvolvimento de Software</i>	107
6 Considerações Finais.....	119
Referências.....	122
APÊNDICE A – Roteiro de Entrevista.....	129
APÊNDICE B – Estrutura do Questionário.....	132

1 Apresentação

Esta pesquisa parte da hipótese de que a cultura do trabalho de desenvolvedores de *software* livre possui especificidades em relação aos contextos mais gerais do mundo do trabalho. O pressuposto é que os valores da cultura *hacker* que influenciam direta (através de uma identidade *hacker* construída no decorrer da trajetória de vida de desenvolvedores de *software*) ou indiretamente (por consequência da relação histórica entre a criação do *software* livre e a ética *hacker* de compartilhamento das informações) a maneira como o trabalho de desenvolvedores de *software* livre é organizado. Para isso, analiso os processos de socialização desses trabalhadores e identifico quais tensões ou adaptações há entre valores dos desenvolvedores de *software* livre/código aberto e os arranjos organizacionais da empresa/fundação.

Assim, assumo o objetivo principal de compreender e analisar a cultura do trabalho de desenvolvedores de *software* livre brasileiros. Os seguintes objetivos secundários se estruturam então como etapas lógicas para alcançar o objetivo principal: analisar sociologicamente as condições de trabalho de desenvolvedores do *software* livre; compreender as relações hipotéticas entre uma ética *hacker* e os processos de socialização e construção de identidades laborais de desenvolvedores de *software* livre; identificar e interpretar as possíveis relações entre a constituição dessas formas identitárias e as alternativas específicas de contratação e de organização do trabalho.

O foco da pesquisa é a construção social da cultura do trabalho de desenvolvedores de *software* livre e código aberto; e assim, a análise dos tipos de vínculos empregatícios, dos tipos de interação entre trabalhadores, usuários e empregadores e das condições de trabalho em relação aos contextos mais amplos. Mas para empreender essa análise é necessária a compreensão de alguns fenômenos, no caso, pressupor a relação entre a construção de identidades ocupacionais, as formas de consciência política ou ideológica, as práticas organizacionais e as formas de socialização anterior e/ou contínua dessas pessoas. Essa decisão de focar diversos fenômenos simultaneamente parte de uma

perspectiva de complexidade do mundo social, de onde procuro me inspirar para considerar os diversos elementos relevantes para uma explicação.

1.1 Justificativa

A pesquisa dialoga com discussões correntes da sociologia do trabalho ao problematizar tensões particularmente interessantes a essa área: as pretensões de padronização do processo de desenvolvimento de projetos por parte da gerência (que se vincula a um contexto mais amplo do mundo do trabalho) não é, frequentemente, compatível com as condições específicas do desenvolvimento do *software*, que exigem maior flexibilidade cognitiva e arranjos sociais complexos (TOLEDO et al., 2009). Isso potencialmente se acentua no caso do *software* livre, onde a influência de uma ética *hacker* que valoriza a colaboração de informações e as liberdades de modificação, além da maior influência da comunidade de usuários no processo de desenvolvimento, pode criar condições peculiares de organização laboral. Dessa forma, o interesse é investigar como os contextos de flexibilidade do trabalho se configuram no caso peculiar de desenvolvedores de *software* livre.

Além do mais, nos anos recentes, tem crescido o interesse pelo *software* livre, seja por vantagens técnicas e econômicas, seja por motivações políticas e culturais, levando em conta o contexto cada vez mais abrangente da ideia de “compartilhamento” no ciberespaço, que deve muito à “cultura *hacker*” em suas diversas manifestações. Por essa razão, houve uma expansão de pesquisas e publicações incentivando o uso do *software* livre, e algumas enfocando as condições técnicas de produção. No entanto, ainda existem poucos trabalhos, mesmo em línguas estrangeiras, baseados em uma perspectiva sociológica do desenvolvimento do *software* livre, ainda mais quando se trata especificamente da questão das condições de trabalho dos desenvolvedores. Assim, espera-se que a pesquisa seja de grande interesse dos profissionais envolvidos com o *software* livre, já que poderá auxiliar no mapeamento deste campo de trabalho e suas especificidades.

1.2 Metodologia

Considero que a constante reformulação das questões de pesquisa significam um esforço para tornar cada vez mais clara a conciliação entre os níveis

epistemológicos do desenho da pesquisa – por um lado a teoria geral e teoria substantiva e por outro a metodologia e os métodos (SAUTU et al., 2005). E o fundamento tanto para a questão teórica quanto para a empírica são comparações, que pretendo levar em conta regularmente. Ainda que a explicação das variáveis se apoie fundamentalmente no método estatístico, e compreendendo não ser sensato fazer um uso indiscriminado da especificidade do método comparativo, as análises só serão compreendidas a partir de uma série de comparações entre as variáveis e entre os contextos das variáveis. Isso porque pressuponho que a comparação, enquanto procedimento intelectual, é a base para qualquer pesquisa científica. Além disso, a comparação quando empreendida a partir de um planejamento, aguça a capacidade de descrição e é um precioso estímulo para a formação de conceitos, conforme cita Collier (1994): “[a comparação] nos proporciona os critérios para submeter as hipóteses a verificação, contribui ao descobrimento por via indutiva de novas hipóteses e à construção de teorias” (p. 51, tradução minha).

O percurso metodológico desta pesquisa se inspirou no modelo da teoria fundamentada (*grounded theory*) (FLICK, 2009a; STRAUSS; CORBIN, 2008), ainda que não necessariamente se feche em seus pressupostos teórico-epistemológicos, já que esta pesquisa não ambiciona redundar em uma nova teoria ao final, tampouco em abandonar de maneira radical os pressupostos teóricos *a priori*. O que proponho é partir de alguns modelos teóricos que suponho serem explicativos dessa realidade específica; e conforme a pesquisa empírica se desenvolve, os modelos são readaptados a fim de produzir novos construtos a partir do que antes seria um interstício ou ponto cego do modelo. A vantagem desse procedimento consiste em potencializar a capacidade dos construtos empíricos em produzir ou reinterpretar as teorias aceitas *a priori*, assumindo assim, uma função maior do que de simples “dados” a serem confirmados ou negados. Essa posição reconhece, portanto, que a pesquisa empírica se enriquece quando há constantes reinterpretações, e que os resultados empíricos são construídos pelo pesquisador por meio deste processo, e é particularmente frutífera quando se deseja abrir mais possibilidades teóricas a um objeto ainda não explorado o suficiente.

Assim, foi prevista a triangulação (FLICK, 2009b) entre métodos quantitativos e qualitativos, compreendendo que os resultados produzidos em cada método (um

survey e algumas entrevistas) complementar o outro durante a constante reinterpretação. A interoperabilidade entre técnicas quantitativas e qualitativas, que uma grande parte da literatura considera como essencialmente opostos ou inconciliáveis, ou ainda atribuindo maior valor para um procedimento em detrimento de outro, sem dúvida não é um procedimento qualquer. No entanto, alguns autores (KING; KEOHANE; VERBA, 1994) oferecem firmes subsídios epistemológicos para considerar que a pesquisa quantitativa e qualitativa valem-se igualmente de inferência empírica, e portanto plenamente conciliáveis, tomando as devidas cautelas. Isso foi exemplificado no trabalho de Bárbara Castro (2013) que, além de realizar essa conciliação metodológica ao pesquisar trabalhadores de TI, faz uso de alguns procedimentos que também julgo como valiosos: a amostragem teórica, a codificação aberta (por influência da teoria fundamentada) e o conceito metodológico de trajetória (que ajuda a explicar os processos de socialização).

Acrescento que é necessário cuidado para não confundir a proposta de deixar a pesquisa circular e receptiva à reformulação de hipóteses com a ideia de que a pesquisa começa aberta a toda e qualquer possibilidade teórica. O pesquisador(a) deve sim se abrir às enriquecedoras possibilidades que o campo insiste ou oferece, mas deve ter consciência de que ele/a não entra lá neutro(a): deve ter consciência de qual são seus pressupostos, seus interesses, suas hipóteses que buscar compreender¹.

A unidade de análise escolhida para selecionar e organizar as informações empíricas foi a trajetória de vida (como representação dos processos de socialização) e as características identitárias/laborais dos desenvolvedores individualmente, já que a partir disso será possível enfatizar as pessoas que incorporam a cultura de compartilhamento e são responsáveis por manter as técnicas como elas são. Por isso, a pesquisa se interessa primordialmente pelas trajetórias de vida e características dos trabalhadores em contextos generalizados de trabalho, e não em características de uma empresa particular, o que por sua vez espera evitar o risco de algum entrevistado se sentir desconfortável em participar em

1 É por isso que escrevo em primeira pessoa, por reconhecer dentre as escolhas metodológicas desta pesquisa meus interesses e minhas limitações agrupadas em função de meus próprios processos de socialização.

virtude de alguma posição em sua empresa. Foi ainda explicitado que os entrevistados não devem revelar informações sigilosas ou sensíveis da empresa em que trabalham, ainda que os nomes reais das empresas não serão publicados.

Com isso, um levantamento bibliográfico sobre o tema fundamentou as primeiras hipóteses e modelos teóricos, sendo apoiada neste período inicial já pelo mapeamento e análise de algumas informações relevantes em bases de dados oficiais e de domínio público, tendo destaque os microdados do Censo 2010. A partir desse material foram planejadas algumas entrevistas exploratórias, em número reduzido, apenas com o intuito de compreender melhor o campo de trabalho no desenvolvimento de *software* livre e suas especificidades, de modo que a etapa principal quantitativa esteja bem fundamentada. Deste modo, foi realizado um *survey* virtual, com uma amostragem estatística, divulgado em listas de e-mails, fóruns e *sites* brasileiros que discutem *software* livre e tecnologia da informação em geral, onde desenvolvedores já estão inscritos e trocam informações técnicas e políticas sobre sua área. Os resultados foram então produzidos inicialmente a partir de uma análise fatorial exploratória (FIELD; MILES; FIELD, 2012), que indicará as relações estruturais entre conjuntos de variáveis, a serem explicadas por uma combinação de estatísticas descritivas e análise de trechos das entrevistas exploratórias. A partir dos resultados não esperados da análise fatorial, que foram numa direção contrária à hipótese inicial, os materiais empíricos foram reinterpretados, para se compreender melhor como esses fenômenos se manifestam no discurso, e consequentemente nas identidades e na cultura do trabalho, e para permitir explicações ainda mais precisas entre as variáveis selecionadas pela pesquisa. Desse modo, a pesquisa é considerada como um dispositivo de aproximações sucessivas ao tema, partindo de considerações vagas até redundar em uma tradução da realidade social mais densa e sistemática.

Foram selecionados para a pesquisa desenvolvedores e desenvolvedoras brasileiros de *software* livre e de *software* proprietário (não-livre). Foram incluídos trabalhadores residindo em qualquer região do Brasil, incluindo os que trabalham em empresas estrangeiras virtualmente, desde que residam no país. Considerou-se “desenvolvedor” uma pessoa que trabalha ou que já trabalhou por mais de um ano com as etapas do processo de desenvolvimento direto do código de *software* livre:

pesquisa, desenho, implementação e teste. Foi considerado o “software livre” em sua acepção mais ampla, incluindo aqueles rotulados como “código aberto”, isto é, basta que o código seja disponibilizado publicamente, havendo ou não uma filosofia que justifica essas práticas. Também foram levados em conta os desenvolvedores de *software* proprietário para permitir a comparação entre os perfis identitários e os arranjos organizacionais entre as duas categorias. Os seguintes segmentos não foram considerados em todas as etapas da pesquisa: entusiastas e contribuidores de código para *software* livre, mas que não atuam profissionalmente com a área; trabalhadores de tecnologia da informação que lidam com *software* livre, mas não desenvolvem códigos.

1.3 Estrutura

A seguir apresento a sequência de desenvolvimento da dissertação: o segundo capítulo realiza uma reconstrução da literatura dos principais temas envolvidos; o terceiro pauta sobre uma comparação entre os resultados empíricos de meu próprio *survey* com o banco de dados² do Censo 2010; já o quarto capítulo realiza uma apuração sobre os perfis de socialização, preferências pessoais e condições de trabalho de desenvolvedores de *software* em geral; finalmente o quinto capítulo enfoca a discussão central da relação entre identidade *hacker* e cultura do trabalho dos desenvolvedores de *software* livre, partindo das características de singularidade do desenvolvimento de *software* livre até a comparação com o desenvolvimento de *software* não livre, tratando das motivações e dos elementos principais da cultura do trabalho envolvidos.

2 De um ponto de vista construcionista, o conceito de “dado” deve ser revisado, já que parte de um pressuposto problemático do positivismo de que a realidade já está “dada” para o(a) pesquisador(a), bastando apenas uma observação “imparcial”. Conforme foi postulado na *Metodologia*, o conhecimento envolvido pela pesquisa sempre envolve construções ativas do(a) pesquisador(a) e opções teóricas e metodológicas. Por isso, quando a palavra “banco de dados” for usada, favor interpretar como um simples vício de linguagem, pois este é o termo utilizado pelas próprias fontes de informação empírica.

2 Contrapontos Teóricos

Neste capítulo, procuro reconstruir algumas das discussões temáticas que fundamentam a problemática da pesquisa, com o intuito de construir um método adequado para os objetivos e técnicas expostos na *Apresentação*. Três tópicos então se complementam: no primeiro, busco contextualizar historicamente a relação entre a cultura *hacker* e *software* livre, que justifica a escolha da unidade de análise sobre a trajetória de vida; no segundo apresento os conflitos discursivos e políticos que envolvem o tema da liberdade como fundamento e dinâmica do *software* livre; e no terceiro argumento em favor da noção de cultura do trabalho como ferramenta de análise das consequências da cultura *hacker* sobre o contexto do trabalho.

2.1 Construções Históricas

Profundas transformações sociais têm ocorrido em todo o mundo nas últimas décadas a partir da reestruturação do capitalismo. Enfocarei rapidamente apenas duas manifestações desse macrofenômeno, as mudanças no “mundo do trabalho” e o surgimento da “sociedade informacional”, para contextualizar a problemática desta pesquisa.

2.1.1 Tecnologia da Informação no Mundo do Trabalho

A expansão das Tecnologias de Informação (TI) nas últimas duas décadas não deve ser descontextualizada da recente reestruturação do capitalismo e suas consequências. Isso significa questionar o pressuposto comum de que a tecnologia possui uma existência à parte da dinâmica social, e que portanto apresenta mais “impactos” sobre a sociedade do que o inverso. Muito se escreve sobre o que as novas tecnologias podem proporcionar de novo para as pessoas, instituições e suas relações; porém é urgente saber também de que forma essas tecnologias são construídas por pessoas, seus contextos e interesses, uma vez que isso aponta os limites e potenciais do que os objetos e processos técnicos alcançam. Portanto, a valorização econômica e simbólica das TI não surgiu apenas porque essa técnica se tornou possível, mas tornou-se possível porque em seu contexto histórico específico houve um interesse coletivo por seu desenvolvimento.

Enquanto muitos acadêmicos se pautaram sobre o conceito de globalização para explicar as grandes mudanças do final do século XX para o início do XXI, essa noção está se tornando insuficiente para compreender a fundo as transformações, já que ela pouco revela a respeito de quais são suas causas. Predomina no discurso comum a ideia de que a globalização é um efeito da disseminação recente das TI, que permitiram a comunicação e troca de informações ao redor do mundo com mais agilidade, e caracterizando então um processo de internacionalização econômica e cultural. Ora, com base nessa concepção genérica é possível precisar as mudanças contemporâneas com mais profundidade sociológica através da crítica de Wallerstein, o que por sua vez permitirá a articulação entre os macrofenômenos e a problemática deste estudo:

Globalização: Este termo foi inventado nos anos 1980. Geralmente é usado ao se referir a uma reconfiguração da economia-mundo que só recentemente veio à existência, na qual é forte a pressão sobre todos os governos para que abram suas fronteiras à livre circulação de mercadorias e capital. Argumenta-se que isso é o resultado de avanços tecnológicos, sobretudo no campo da informática. O termo é tanto prescrição quanto descrição. Já para analistas do sistema-mundo, aquilo que é descrito como algo novo (fronteiras relativamente abertas) tem na verdade sido uma ocorrência cíclica através da história do moderno sistema-mundo. (WALLERSTEIN, 2006, p. 93, tradução minha)

Partindo do extremo de uma perspectiva macrossocial, Wallerstein tem como conceito central a ideia de sistema-mundo, na qual não só as classes sociais em um país (ou em uma “sociedade”) apresentam uma divisão do trabalho geradora de desigualdade e exploração, mas também os próprios países produzem esses efeitos entre si. Com isso, a formação do sistema-mundo moderno, que integra os circuitos comerciais do globo, se deu gradualmente através dos séculos XVI ao XX, tendo início na colonização das Américas segundo a revisão de Mignolo (2005). Esses processos levaram a cabo a hegemonia europeia até a Segunda Guerra Mundial, momento em que os Estados Unidos consolidam o domínio do sistema-mundo (WALLERSTEIN, 2009), já que se erguem como a maior potência industrial, financiando a reconstrução da Europa Ocidental e Japão, e mantendo esses e demais aliados congruentes com sua política externa, com base na justificativa de um poderoso inimigo militar comum, a União Soviética.

Mas segundo o mesmo autor (2009) os anos 1970 marcam o início do declínio dessa hegemonia estadunidense por vários motivos: a derrota militar no Vietnã (que racha a opinião pública sobre o envio de soldados para guerras), o fortalecimento de contraculturas e da Nova Esquerda (os movimentos negro, feminista, ecológico, homossexual, dentre outros), o crescimento industrial da Europa Ocidental e Japão (que começam a competir economicamente com os EUA) e a crise do petróleo. Em resposta a essa crise estrutural, as políticas neoliberais começam a serem postas em prática na década de 1980 em diante, se apoiando em um fracasso do desenvolvimentismo³ como solução das desigualdades no chamado Terceiro Mundo, e argumentando a ineficiência do Estado de Bem-Estar Social (*Welfare State*). Tomando como base os governos de Thatcher e Reagan, com ajuda do Fundo Monetário Internacional e do Fórum Econômico Mundial de Davos, o plano neoliberal foi imposto às políticas econômicas do maior número possível de países integrantes do sistema-mundo:⁴

Uma nova definição do caminho para a terra prometida – o chamado Consenso de Washington – inverteu a maioria dos dogmas do desenvolvimentismo. A industrialização por substituição de importações era agora definida como um processo de favorecimentos corrupto; a construção estatal, como alimentando uma burocracia inchada; a ajuda financeira dos países ricos, como dinheiro derramado na sarjeta; e as estruturas paraestatais, como barreiras mortais para uma atividade empresarial lucrativa. Os Estados foram impelidos a adiar gastos com a educação e com a saúde. E foi realçado que as empresas públicas, consideradas, por definição, ineficientes, deveriam ser privatizadas o quanto antes. O “mercado”, mais do que o bem-estar da população, agora se tornava a medida de toda atividade adequada do Estado. O Fundo Monetário Internacional fortaleceu essa visão, tornando os seus empréstimos dependentes do “ajuste estrutural”, o que significa essencialmente seguir as prescrições do Consenso de Washington. (WALLERSTEIN, 2009, p. 62)

-
- 3 “A ideia de que, de algum modo, adotando uma política estatal correta, qualquer país poderia atingir o alto padrão de vida dos países mais ricos. [...] A proposta subjacente era de que a combinação de urbanização, agricultura mais eficiente, industrialização, educação e protecionismo de curto prazo (substituição de importações) constituíam o caminho para a terra prometida do desenvolvimento.” (WALLERSTEIN, 2009, p. 61)
- 4 Ainda que o poder de influência hegemônico dos Estados Unidos, segundo Wallerstein, tenha começado a declinar a partir de 1970 (e inclusive em um ritmo mais acelerado a partir de 2001), a proposta neoliberal conseguiu se efetivar e reestruturou o capitalismo e suas justificativas. Podemos considerar que, como toda etapa do capitalismo, eventualmente após um ápice surge uma crise, especificamente a partir de 2008, que trouxe consigo novas ondas de protestos, reviravoltas e desalinhamentos políticos, e críticas aos Estados Unidos e União Europeia.

Dessa forma, foi realizada uma transformação estrutural na economia do sistema-mundo nas últimas décadas, e a partir dela, uma mudança ideológica da hegemonia capitalista: a nova receita para que os países evitem e resolvam as crises passa a ser o conjunto das práticas propostas pelo neoliberalismo, conforme podemos observar segundo a síntese de David Harvey:

O neoliberalismo é em primeiro lugar uma teoria das práticas político-econômicas que propõe que o bem-estar humano pode ser melhor promovido liberando-se as liberdades e capacidades empreendedoras individuais no âmbito de uma estrutura institucional caracterizada por sólidos direitos a propriedade privada, livres mercados e livre comércio. [...] As intervenções do Estado nos mercados (uma vez criados) devem ser mantidas num nível mínimo, porque, de acordo com a teoria, o Estado possivelmente não possui informações suficientes para entender devidamente os sinais do mercado (preços) e porque poderosos grupos de interesse vão inevitavelmente distorcer e viciar as intervenções do Estado (particularmente nas democracias) em seu próprio benefício. [...] Quase todos os Estados, dos recém-criados após o colapso da União Soviética às social-democracias e Estados do bem-estar social ao velho estilo, como a Nova Zelândia e a Suécia, adotaram, às vezes voluntariamente e em outros casos em resposta a pressões coercivas, alguma versão da teoria neoliberal e ajustaram ao menos algumas políticas e práticas aos seus termos. [...] Além disso, os defensores da proposta neoliberal ocupam atualmente posições de considerável influência no campo da educação [...], nos meios de comunicação, em conselhos de administração de corporações e instituições financeiras, em instituições-chave do Estado [...], bem como em instituições internacionais como o Fundo Monetário Internacional (FMI), o Banco Mundial e a Organização Mundial do Comércio (OMC), que regulam as finanças e o comércio globais. Em suma, o neoliberalismo se tornou hegemônico como modalidade de discurso e passou a afetar tão amplamente os modos de pensamento que se incorporou às maneiras cotidianas de muitas pessoas interpretarem, viverem e compreenderem o mundo. (HARVEY, 2008, p. 12-13)

Em parte exatamente por causa dessas mudanças e em parte apenas simultaneamente a esse contexto, um processo continuamente construído é a precarização das relações de trabalho, através do qual empresas em todo o mundo valorizam contratos de trabalho flexíveis, projetos de curto prazo e terceirizações. Nesse sentido, as organizações de TI desempenham um papel fundamental, já que elas lidam com as informações para as novas formas de gestão do trabalho. Paralelo a isso, é preciso lembrar que o trabalho em TI é, desde o seu surgimento, adaptado para o regime de flexibilidade (CASTRO, 2013).

Enquanto que, entre diversos trabalhadores da área de TI, a flexibilização das formas de trabalho é entendida de forma positiva (tal como explicitado no próximo capítulo), o processo de flexibilização se constituiu historicamente apenas como uma

nova estratégia de controle do trabalho em direção à eficiência no contexto da crise estrutural dos modelos anteriores.

O novo modelo que viria a ser conhecido por toyotismo se baseou nas inovações da Toyota a partir dos anos 1960-1970, que consistiu em adaptar o modelo fordista às necessidades japonesas do pós-guerra: pouco espaço físico, demanda débil, e indústria diversificada (GOUNET, 2002). Uma parte importante na iniciativa vanguardista dos japoneses foi padronizar as autopeças, para não exigir que cada fábrica “reinvente a roda” em sua própria linha de produção. Isso permitiu que empresas se especializassem em peças que poderia ser usadas entre modelos de diferentes companhias automobilísticas. A flexibilização dos componentes permitiu assim que um mercado ainda em formação se estruturasse mais rápido e chegasse ao nível competitivo com menos custos. Vale lembrar que esse modelo se torna tão eficiente que se expande para outras indústrias, incluindo a de *software*, onde por exemplo, uma das chamadas “metodologias ágeis de programação” utiliza explicitamente conceitos do toyotismo: o *kanban* e o *just in time*.

Com a gradual sobreposição do toyotismo sobre o fordismo, não apenas o modo de produção industrial tem sido reestruturado, mas também as condições de trabalho se alteraram para uma conjuntura de flexibilidade e precarização, incluindo, ainda que com suas especificidades, o trabalho nos setores de serviços (NUNES, 2011). Assim, a precariedade pôs em curso uma situação generalizada e permanente de insegurança, buscando cada vez mais naturalizar os contextos de exploração, derivando mais de escolhas políticas – o hegemônico discurso neoliberal – do que de limitações estritamente econômicas (BOURDIEU, 1998).

Compreender as reestruturações do capitalismo, por meio de uma investigação das mudanças ideológicas subjacentes, foi o objetivo de Boltanski e Chiapello (2009). Tendo por método uma análise sistemática de literatura de gestão empresarial e também de textos críticos, o autor e a autora procuraram demonstrar como cada estado do capitalismo se constrói sobre uma justificativa discursiva. Essas justificativas se estruturam com tal força social a fim de perpetuar a acumulação ilimitada de capital, que até mesmo os discursos críticos acabam por sustentar também esse vocabulário de justificativas. Eles identificam então, modelos de “cidades”, uma metatipologia das justificativas do capitalismo: a cidade inspirada,

a doméstica, a da fama, a cívica, a mercantil, a industrial e a mais recente, a cidade por projetos. E é assim que a partir dos anos 90, toma força um novo ideal de trabalhador, marcado pela capacidade de adaptação, flexibilidade e gestão de projetos transitórios.

Nesse sentido, vale lembrar que as empresas de TI (Tecnologia da Informação), devido ao seu recente desenvolvimento organizacional, já surgem flexíveis, descentralizadas e estruturadas em projetos e equipes, como notam Rosenfield e Mossi (2011). Além disso, as autoras problematizam que, entre os quadros superiores de TI, tanto a flexibilidade negativa (a precariedade) quanto a positiva (humanizada) se combinam, formando uma peculiar e complexa configuração de trabalho.

2.1.2 *Cultura Hacker e Software Livre*

Tendo essa contextualização histórica em conta, cabe retomar outra feição da reestruturação do capitalismo, que Castells (2005) denomina como “sociedade informacional”. Com a crescente valorização das tecnologias da informação, a indústria de *software* tem se destacado como vital para a organização e controle das informações nos mais diversos contextos. Por isso, é desde os primórdios da história da informática que se toma consciência política desse campo estratégico. Já é reconhecida a relação histórica entre a cultura *hacker* e o desenvolvimento do *software* livre (SILVEIRA, 2010) e da própria Internet (CASTELLS, 2003).

Para compreender isso, devemos notar que a definição êmica⁵ de *hacker* não diz respeito necessariamente à questão de invasão ilícita de dados informáticos, conforme frequentemente exposto pelas grandes mídias, que agrupam as noções *hacker*, *crackers* e *ciberpiratas* sob o mesmo rótulo. As definições de *hackers* e *ciberpiratas* que circulam hegemonicamente, ainda que não sejam adotadas juridicamente, constituem representações sociais que justificam e/ou condenam práticas específicas. Para melhor interpretar isso, levo em conta a definição de que “as representações sociais são uma forma de conhecimento socialmente elaborado e compartilhado, com um objetivo prático, e que contribui para a construção de uma

⁵ A definição êmica diz respeito às categorias e valores a serem interpretados a partir do próprio grupo social e cultural na qual é utilizado.

realidade comum a um conjunto social” (JODELET, 2002, p. 22). Ainda que esta definição não inclua mais precisamente as disputas políticas que as formas de conhecimento põem em jogo, ela enfatiza o caráter construtivista da relação entre os discursos e as práticas. Por exemplo, no Dicionário Online de Português, o termo hacker é definido da seguinte forma: “aquele que possui um vasto conhecimento na área informática, geralmente invadindo sistemas computacionais ou computadores com propósitos ilegais” (HACKER, 2016).

No que diz respeito às práticas de troca e reprodução de arquivos digitais, também é comum a confusão, por vezes proposital, entre categorias diferentes, agrupando-as sob a ideia de contravenção. Por exemplo, o dicionário Houaiss define hacker apenas como um sinônimo de “ciberpirata”, e este verbete por sua vez como “pessoa com profundos conhecimentos de informática que eventualmente os utiliza para violar sistemas ou exercer outras atividades ilegais; pirata eletrônico” (HOUAISS; VILLAR, 2009, p. 461).

O debate sociológico sobre as práticas de reprodução e compartilhamento não-autorizado de arquivos digitais apresenta diferentes perspectivas. Por um lado vários ativistas começam a utilizar os termos a que são criticados em seu favor, como por exemplo os Partidos Piratas⁶. Contudo, o termo “pirataria” ou a alcunha de “pirata”, por um lado, é utilizado, principalmente pelas organizações que representam os detentores dos direitos autorais, como forma de criticar moralmente os usuários que praticam a troca não autorizada de arquivos digitais. Por outro, a utilização desses termos acaba por instalar uma confusão sobre a real contravenção que está sendo supostamente realizada, pois o mesmo termo é utilizado para a produção não autorizada de cópias físicas e digitais (com ou sem fins lucrativos) de produtos culturais e para outros tipos de crime, como a falsificação de produtos, de marcas ou a contrafação de propriedade industrial.

Não parece haver um consenso a respeito da reapropriação desse termo pelos ativistas, pois existem muitas posições políticas possíveis, incluindo também a

⁶ Existem vários partidos políticos, principalmente na Europa, que adotam o rótulo de “Partido Pirata”, enquanto que em outros países, como o Brasil, esse nome está associado a movimentos que ainda não conseguiram o registro válido para eleições. Pautas comuns, tanto entre os diferentes partidos como movimentos, são: reforma das leis de direitos autorais e patentes; valorização do livre compartilhamento de conteúdos livres; privacidade da informação; transparência estatal; liberdade de informação; neutralidade da rede. (PPI, 2015)

recusa com a associação entre modificação de conteúdo informático/cultural e o ato criminoso de roubo. Ainda assim, é possível sintetizar a posição da maioria dos movimentos relacionados aos rótulos de “*hacker*”, “piratas” e “*software* livre” a partir da valorização da liberdade dos usuários “comuns”, tal como na proposição de outro movimento correlato, o denominado *cypherpunk*: “privacidade para os fracos e transparência para os poderosos” (ASSANGE, 2013). Todavia, daqui em diante a discussão não levará em conta a questão da marginalidade legal ou associação da ideia de *hacker* com representações negativas, já que o foco será tratar o que diz respeito ao conceito de *hacker* em seu contexto histórico original e o surgimento do *software* livre.

Quanto à noção original de *hacker*, segundo Silveira (2010, p. 32), trata-se de “um programador de computador talentoso que poderia resolver qualquer problema muito rapidamente, de modo inovador e utilizando meios não convencionais”. Vale lembrar como essa característica *hacker* se assemelha com a ideia de “flexibilidade cognitiva” que Toledo et al. (2009) atribuem aos programadores de *software* em geral: a “disposição do indivíduo para assumir uma atitude relativamente aberta na resolução de problemas, que implica aprendizagem pessoal e ampliação de habilidades cognitivas” (p. 138).

Eric Raymond, por ter feito parte dos primeiros grupos *hacker* nos anos 70, remete a uma legitimação da comunidade essa atribuição: “os *hackers* são aqueles que a cultura *hacker* reconhece como tais” (*apud* CASTELLS, 2003, p. 38). Apesar de parecer tautológica, essa definição remonta à ideia de *outsider* exposta por Becker (2008), na qual é o rótulo da diferença que agrupa as pessoas em determinadas situações sociais, o que por sua vez, permite o desenvolvimento de subculturas. Por fim, Himanen (2001) considera como *hackers* as pessoas com as seguintes características: conhecimentos avançados de programação; motivação pela computação mais por paixão do que por interesses instrumentais; empenho em compartilhar o que produziu e aperfeiçoar o que foi produzido por outros. E é precisamente a partir dessa noção de motivação passional pelo compartilhamento, que configura o que Himanen considera como “ética *hacker*”, que se constituiu a ideologia de desenvolvimento de *software* livre.

Em seu clássico ensaio *A catedral e o bazar*, Eric Raymond (1998) buscou, já na década de 90, compreender o surgimento e rápida expansão do modelo do desenvolvimento do *kernel* Linux de código aberto. Para isso, ele chegou a conceituar uma diferença entre dois modelos de desenvolvimento em código aberto: a metáfora da catedral, “habilmente criad[a] com cuidado por mágicos ou pequenos grupos de magos trabalhando em esplêndido isolamento, com nenhum beta para ser liberado antes de seu tempo” (p. 5); e o bazar, onde o código é aberto publicamente desde as etapas iniciais, com uma construção colaborativa da comunidade, que revisa incessantemente as novas versões lançadas com frequência. O ensaio, argumentando sobre uma maior eficiência técnica do modelo bazar, com a máxima de que “havendo olhos suficientes todos os erros são óbvios” (p. 16), chegou a ser uma das influências para a Netscape a liberar seus códigos, iniciando o projeto Mozilla (CASTELLS, 2003).

Em resposta ao crescente uso do termo “código aberto” (*open source*), outro ícone histórico da cultura *hacker*, Richard Stallman (2010) manteve a defesa do termo original “*software* livre” por enfatizar menos as consequências técnicas do código aberto, e mais o que ele considera as quatro liberdades fundamentais para um *software* ser considerado livre: “a liberdade para executar o programa, para qualquer propósito; a liberdade de estudar como o programa funciona, e adaptá-lo às suas necessidades; a liberdade de redistribuir cópias, de maneira que outras pessoas possam se beneficiar; a liberdade de distribuir cópias de suas versões modificadas a outros, de maneira que toda a comunidade possa se beneficiar de suas mudanças” (p. 3, tradução minha).

Em meio a essa disputa semântica e política pelo conceito mais adequado, é preciso enfatizar que de um ponto de vista rigorosamente técnico é inviável, senão impossível, diferenciar um programa ou um código como “livre” ou “aberto”. Por mais que as duas nomenclaturas sejam mais do que apenas simbólicas, a diferença só se evidencia no âmbito das motivações das duas organizações promotoras de SLCA: a *Open Source Initiative* (OSI) e a *Free Software Foundation* (FSF), conforme especificarei a seguir. Nem mesmo as diferenças entre as licenças de *software* conduzem a um contraste sem ambiguidades entre os dois termos, já que a maioria das licenças para esse tipo de *software* é reconhecida (ou recomendada) por ambas

organizações. E é a partir das diferenças entre as motivações da FSF e OSI que se constroem não só discursos, mas perspectivas divergentes sobre qual a finalidade e quais as consequências de se desenvolver *software* sem restrições ao código-fonte.

Para complicar ainda mais a situação terminológica, nem sempre o conceito de *software* livre (SL) é usado com os requisitos que a FSF atribuiu originalmente, ou seja, para muitos usuários e desenvolvedores que não conhecem ou não se importam com essa disputa semântica, o termo *software* livre é empregado como se fosse o mesmo que *software* de código aberto (CA), transformando-o em um hiperônimo, que inclui ambos significados.

Não cabe aqui se aprofundar na história do movimento do *software* livre, mas o propósito dessa contextualização foi demonstrar que existe uma relação teórica e histórica entre uma ideologia *hacker* e o desenvolvimento do *software* livre. Falta então, entender as possibilidades contemporâneas, no caso brasileiro, de uma relação sociológica nos contextos de trabalho envolvidos, de forma que pudessem redundar em culturas do trabalho alternativas. Apesar de ser uma área ainda em mapeamento pela sociologia do trabalho, é sabido que o Brasil possui um número relativamente expressivo de desenvolvedores de *software* livre, especialmente devido às políticas governamentais da última década, que têm incentivado o uso e desenvolvimento dessa proposta. Isso tem levado alguns desenvolvedores de *software* a se especializarem profissionalmente nessa área, devido à demanda de algumas empresas pela utilização de *software* em código aberto, mas também por interesse ideológico desses profissionais. Isso faz com que uma parte desses trabalhadores se dedique ao *software* livre de maneira diretamente remunerada, seja por interesse instrumental ou passional, seja indiretamente, por meio de outros projetos que desenvolvem simultaneamente como *freelancer*, por exemplo, ou ainda, por um trabalho completamente voluntário, realizado no tempo livre – mesmo que esse tempo esteja fracionado nos intervalos do tempo de trabalho remunerado.

Mas ainda restam várias questões a serem exploradas, como quais outros projetos ou empregos se ocupam aqueles desenvolvedores que se flexibilizam ao máximo, lidando tanto com *software* livre quanto com proprietário? Será que mesmo nessas condições se poderia continuar pensando numa influência da ética *hacker* sobre os contextos de colaboração e inovação no trabalho? Também será

necessário ao longo do trabalho pensar se mesmo nas condições mais favoráveis para o trabalho em *software* livre, estes trabalhadores interpretam suas atividades como uma alternativa à dinâmica capitalista tradicional de trabalho ou não.

Ademais, Enrique de la Garza Toledo e outros (2009) apresentaram algumas características gerais do trabalho de desenvolvedores de *software* que convém notar: programadores de *software* costumam se identificar mais com a profissão de programador do que às empresas em que trabalham; por isso, é igualmente frequente que programadores considerem seus empregos apenas como uma forma de aprendizado para um trabalho melhor no futuro, seja em outras empresas, seja por conta própria, o que permite que se submetam temporariamente a algumas condições de flexibilidade precarizada. Por fim, o desenvolvimento de *software* é um trabalho onde, durante todo o processo, há uma intensa inter-relação entre o grupo de programadores, os clientes/usuários e os responsáveis pelo projeto informático.

E mais um ponto a se acrescentar é a respeito da maneira pela qual o *software* livre é comercializado. Nesse modelo, a liberdade de distribuição deve ser assegurada, e portanto, o produto final pode ser vendido, mas não se pode restringir sua aquisição apenas à venda, já que no mínimo é possível compilar o código-fonte para criar e, eventualmente, redistribuir os executáveis. Assim, a principal maneira pela qual as empresas de *software* livre e código aberto produzem seu lucro é oferecendo serviços de assistência técnica para as empresas que utilizam o *software*. Essa configuração se baseia portanto nas considerações pós-fordistas de se remeter o valor do produto para um serviço.

2.2 Conflitos Discursivos

Nesta seção busco explicitar como a noção de liberdade é valorizada em um primeiro momento pelo movimento do *software* livre e como ela influencia outros setores relacionados ao desenvolvimento de *software*, ou melhor, como essa noção é ressignificada pela iniciativa *open source*. Além disso, será necessário ressaltar uma conexão prevista pela literatura entre os ideais do SLCA e alguns fundamentos do liberalismo. Mais do que simplesmente uma relação direta, que identificaria o *software* livre com essa ideologia política, também essa conexão se constrói de forma confusa e complexa, já que para além das motivações originais dos

personagens-chave do *software* livre e do *software* de código aberto surgem reinterpretações de novos movimentos que veem um maior leque de possibilidades com essas ferramentas e formas de colaboração. Dito de forma mais direta, é comum ou ao menos implícito que pesquisas ou pesquisadores das ciências humanas tenham a tendência em associar o *software* livre com uma iniciativa anticapitalista, o que não é preciso na maioria dos casos, já que a motivação de liberdade é compatível com as demais liberdades valorizadas pelo liberalismo clássico. No entanto, também não devemos chegar ao cúmulo de negar essa relação possível: quando me referi a “novos movimentos” quero dizer que existem também propostas políticas radicais pautadas ou na filosofia do SL ou na quase dependência das ferramentas livres, ou seja, disponíveis para qualquer tipo de uso.

Assim, cabe fazer algumas considerações acerca de um contraponto entre o discurso ideológico desses trabalhadores – no discurso de alguns ativistas de *software* livre fica implícita uma organização do trabalho não-capitalista – e uma contextualização mais geral do mundo do trabalho e das formas organizacionais – que apresentam compatibilidade entre os valores do código aberto e os interesses do mercado em flexibilizar a produção e o modo de agir e pensar de seus trabalhadores.

Por um lado, podemos considerar que os participantes das comunidades de SLCA (incluindo então além dos desenvolvedores, os usuários e entusiastas) apresentam em seus discursos a importância de motivações consideradas como altruístas (BAYTIYEH; PFAFFMAN, 2010) para a dedicação de várias horas de trabalho a projetos que frequentemente não se relacionam diretamente com seu próprio trabalho remunerado. E é desse sentido de valorização semântica das motivações dos desenvolvedores e entusiastas de *software* livre que Mazières (2009) extrai uma noção de agnosticismo político.

Por outro lado, é possível identificar dentre os fundamentos filosóficos do *software* livre e do código aberto as justificativas do liberalismo clássico, conforme demonstra Gabriella Coleman (2013): os *hackers* preocupados em distribuir e readaptar informações sem barreiras, a fim de construir um conhecimento livre, produzem com isso a crítica ao neoliberalismo (a transformação da informação e do conhecimento em propriedade privada) a partir de fundamentos do liberalismo

clássico (a liberdade de um termina onde começa a liberdade de outro). Essa fundamentação política para além dos discursos demonstra então uma possível articulação com modelos mais tradicionais das teorias das organizações.

A respeito da dicotomia entre a escolha de um uso mais “livre” e um uso mais “seguro” do ciberespaço, tal como problematizado por Bauman (1998) a partir da discussão clássica na política, podemos considerar que de um ponto de vista teórico essa polarização segue plausível, mas ainda não parece ser suficiente para centralizar as tensões do campo. Em *O Mal-Estar da Pós-Modernidade*, Bauman estabelece que o ponto referencial para se compreender a pós-modernidade⁷, ou seja, seu valor supremo, é a “vontade de liberdade”; isto é, na trilha das mudanças (líquidas) sociais, econômicas, culturais, tecnológicas, cotidianas, em que se convencionou definir por pós-modernidade, Bauman entende que há agora uma nova experiência social, na qual “o mundo é vivido como incerto, incontrolável e assustador”, uma consequência da opção pela “liberdade” em detrimento da segurança e da “ordem social”. Isso por sua vez se oporia à escolha feita durante a modernidade (sólida) da “segurança projetada em torno de uma vida social estável, ou da ordem”, que subjugava os indivíduos limitando seus desejos e possibilidades, conforme pensou Freud em *o Mal-Estar da Civilização*.

Do ponto de vista das discussões e dinâmicas próprias do *software* livre, essa dicotomia tem um potencial explicativo reduzido, já que o *software* livre busca ao mesmo tempo tanto a liberdade (através das quatro “liberdades” específicas) e a segurança (atingida tecnicamente através da ideia de *accountability* coletiva ou escrutínio público). Uma tensão mais comumente citada no meio é a que acontece entre a segurança e a praticidade: dito de forma genérica quanto mais prática for uma tarefa para o usuário, mais prática também será para um atacante.

Um outro tema comum, a valorização vigente da “inovação” se manifesta principalmente no interesse pela propriedade intelectual, na tentativa de proteger esse tipo de bem cada vez mais valorizado economicamente, o conhecimento. As

7 O conceito de pós-modernidade continua, mesmo após décadas, extremamente polêmico, tanto na variação de rótulos (modernidade líquida, modernidade tardia, neomodernismo...) quanto na aceitação de suas características e mesmo sua existência. Nesta dissertação me contento em afirmar que o mundo passou por “grandes transformações nas últimas décadas”, identificando na seção *Construções Históricas* como algumas dessas mudanças se correlacionam com a temática central e assim ignorando outras.

licenças abertas, por sua vez, apesar de serem uma das armas fortes dos movimentos anticapitalistas, mantêm sua sustentabilidade logística sobre o interesse de novos tipos de empresa (aquelas com mentalidade “*open source*”), que compreendem que o modelo vigente de propriedade intelectual a longo prazo poderá significar não só a ineficiência como também a imobilização da indústria de *software*.

Além de tudo, a indústria de *software* se consolidou como uma ferramenta vital para o armazenamento e circulação do capital, através dos setores de TI em empresas, da disseminação das TIC, e ainda por meio da comoditização de *big data*⁸. Ademais, como pontuado na seção anterior, ela se torna através do *software* livre um novo modelo para a flexibilização do capital, na medida em que empresas que “competem” entre si também colaboram na produção do código: ao terem a liberdade de adaptar uma mesma fonte a várias necessidades específicas, todas as empresas se beneficiam na eficiência de não precisarem partir do zero na programação, e também na segurança de possuir o código escrutinado e corrigido publicamente.

2.3 Cultura do Trabalho

Apesar da polissemia, e conseqüente dificuldade metodológica, do conceito de cultura, é viável e profícuo aplicar o conceito de cultura do trabalho (FONTAINE, 2011) para compreender como as diferentes identidades ocupacionais se constroem por meio de processos de socialização mais amplos (DUBAR, 2005) e das disputas em torno dos significados atribuídos ao trabalho.

A noção de cultura do trabalho é mais predominante na sociologia francesa do trabalho, enquanto que um conceito próximo (e por isso também levado em conta neste estudo) é o de arranjos organizacionais, proveniente da literatura da administração. Ainda, existem as variantes “cultura organizacional” e “cultura de empresa” utilizadas em um sentido que considera as discussões da teoria tradicional das organizações sob influência da noção de cultura.

⁸ Conjunto de informações em uma escala muito grande, de modo que só possam ser coletados ou processados com computadores, especialmente com referência ao seu potencial para permitir novos entendimentos em um campo particular de estudo (BIG DATA, 2019).

Inicialmente a noção de “cultura da empresa” surgiu nos anos 70 nos Estados Unidos e nos 80 na França, através do discurso de empresas que ansiavam por justificar as mudanças na sua organização do trabalho, atribuindo as mudanças a uma “cultura” exterior, estática e acima das decisões individuais que, por conseguinte, deve ser incorporada ao empregado contratado (CUCHE, 2002). No entanto, o conceito foi reapropriado pela sociologia do trabalho, através de considerações mais críticas do conceito de cultura, sem vê-la como um fenômeno supraindividual, estático e uniforme, mas sim um processo de disputas simbólicas.

Annie Fontaine (2011) utiliza três possibilidades teóricas de dimensionamento da noção de “cultura do trabalho”, a construtivista, a interacionista e a etnometodológica. Trata-se de compreender respectivamente: a cultura como um universo compartilhado e negociado de elementos simbólicos; a cultura como processo e produto de interações; e as conversas e rotinas como construção de um mundo comum.

Posto isso, busquei compreender a cultura do trabalho sob inspiração da tradição francesa da sociologia do trabalho, que enfatiza a importância dos processos de socialização na construção desses campos simbólicos. Para isso, o eixo teórico fundamental será Claude Dubar (2005), que propõe uma teoria sociológica operacional da construção das identidades.

O autor centra seu argumento na assunção de que as identidades são construídas a partir de transações objetivas e subjetivas, respondendo a dois conflitos possíveis: entre a identidade herdada e a identidade visada, estando em jogo os “compromissos internos” do *self*; mas também entre a identidade atribuída pelos outros e a identidade incorporada por si, estando em jogo as “negociações externas”. Ademais, essa dinâmica apresenta dois eixos em que se desenrola a construção de identidades: “Um eixo ‘sincrônico’, ligado a um contexto de ação e a uma definição de situação, um espaço dado, culturalmente marcado; e um eixo ‘diacrônico’, ligado a uma trajetória subjetiva e a uma interpretação da história pessoal, socialmente construída” (DUBAR, 2005). Através da articulação desses dois eixos se constroem as definições de si, ao mesmo tempo enquanto um ator de um sistema determinado e enquanto um produto de uma trajetória específica. Essa dualidade torna problemáticas as identificações: entre as definições “oficiais”, atribuídas por outrem, e as identificações “subjetivas”, reivindicadas por si e

submetidas ao reconhecimento de outrem, todas as combinações são possíveis, em um contexto dado. Assim, as identidades são pensadas como em movimento, e numa perspectiva relacional. Além disso, cabe ressaltar as negociações identitárias, por constituírem um processo comunicativo complexo, irreduzível a uma “rotulagem” autoritária de identidades predefinidas com base nas trajetórias individuais.

Reporto agora aos trabalhos que, ao tipificar estratégias, formas identitárias e motivações, ajudam a reconstruir teoricamente culturas do trabalho específicas: as estratégias que desenvolvedores empreendem para conciliar a programação em *software* de código aberto e de código fechado, ao trabalharem em empresas (ROLANDSSON, BERGQUIST, LJUNGBERG, 2011); os tipos de inovação organizacional em empresas desenvolvedoras de *software* livre (SÁNCHEZ, 2010); as motivações profissionais dos programadores em *software* livre (AUGUSTO, 2003), a saber, o hobby (ou ética *hacker*), a aprendizagem instrumental, e a obrigação profissional; e as próprias formas identitárias de Dubar (2005), que mapeiam as relações entre os processos de socialização e as transações subjetivas e objetivas da identidade profissionais e ocupacional.

Compreendendo que a construção de tipologias não pode ser um fim em si mesmo para uma pesquisa empírica na sociologia (ORTIZ, 2008), intentei instrumentalizar as tipologias – tanto as já construídas na teoria e as que construirei a partir do campo – para construir pontes entre o geral e o específico, usando os conceitos teóricos para trazer precisão analítica durante as etapas da pesquisa, e usando os resultados empíricos assim construídos para reinterpretar os modelos teóricos assumidos até então.

Cabe ainda fazer um contraponto, para apresentar outra interpretação possível: que a valorização da flexibilidade por esses profissionais, essa cultura organizacional mais livre e adaptada às necessidades dos desenvolvedores seja também uma estratégia de controle de qualidade por parte das empresas, que buscam assim, apenas mudar seus critérios por outros. E para isso, desde já é preciso lembrar que muitas das teorias da organização que buscam se sobrepor ao fordismo, apresentando novos elementos a se levar em conta, como a cultura ou a psicologia, na verdade se compõem a partir de pressupostos de controle do trabalho

presentes desde Taylor, havendo não uma negação simples, mas uma constante readaptação (HELOANI, 2003).

Posto isso, cabe compreender até que ponto o desenvolvimento de *software* livre consegue realmente apresentar uma alternativa ao modelo hegemônico de cultura organizacional, em que os trabalhadores produzem em um contexto de flexibilidade favorável a eles, que apenas por conseguinte, produziria um bom retorno financeiro. Ou concluir se de fato, a tese da precarização geral do trabalho é implacável, e o que acontece é que esses trabalhadores estão na verdade em tão grande simbiose com o meio em que trabalham, que não percebem que ela seria também uma forma sutil de manipulação, justamente por incutir (ou se aproveitar) das socializações anteriores desses indivíduos, em vez de treiná-los a uma nova forma de trabalhar.

Dentre alguns dos conceitos a serem revisados, cabe lembrar que a dinâmica de desenvolvimento do *software* livre – o modelo bazar em que não há um pré-planejamento a ser seguido por uma diretoria, mas uma constante reformulação pela comunidade de desenvolvedores – tem muito a ser explicada pelas considerações sobre a cooperação feitas por Marx (2002), que valoriza desde já a força coletiva como qualitativamente (ou seja, não apenas quantitativamente) diferente da individual. Explicada a partir dos primórdios do capitalismo, os princípios de aprimoramento da cooperação no trabalho – não apenas dividir as diferentes tarefas de um mesmo produto, mas também o simples fato de várias pessoas estarem juntas em um mesmo lugar para cumprir um número de horas delimitada – permitiram uma maior extração de mais-valia, por aumentarem consideravelmente a produtividade. Tendo isso em mente, não vamos longe se considerarmos que a cultura de compartilhamento valorizada no âmbito do *software* livre – mesmo que ideologicamente se oponha ao modelo de patentes fechadas do *software* proprietário – tem também um viés capitalista no sentido de que o resultado da produção de um *software* em código aberto é mais produtivo: desenvolvido mais rapidamente, com menos custos, mais seguro (por conter consideravelmente menos erros). A ideia de “abertura” (do código) se torna assim facilmente uma flexibilidade para o capital.

Quanto ao conceito de “flexibilidade cognitiva”, que Garza et al. (2009) propõem como uma habilidade particularmente necessária para os desenvolvedores de *software*, cabe ressaltar que essas mesmas competências são cada vez mais valorizadas pelos administradores de grandes empresas, em consonância com as contemporâneas reestruturações do trabalho. Trata-se então não apenas de produzir trabalhadores com uma mentalidade específica, que esteja de acordo com o ideal organizacional da empresa, através de várias técnicas de controle da subjetividade, tal como descritas por Max Pagès et al. (1987). Mas sim, de radicalizar o controle de eficiência que Taylor valorizou, e assumir uma relação entre organização e subjetividade mais poderosa, justamente porque não é uma simples imposição, mas é previamente valorizada pelos indivíduos. Tendo isso em mente, o foco continua a ser compreender as possíveis tensões deste contexto específico com as condições históricas mais amplas, que por sua vez configuram uma continuidade complexa com as técnicas de controle dos trabalhadores detalhadas pelas teorias clássicas das organizações.

3 Características Gerais e Contextos

Neste capítulo, investigo a relação entre os conflitos, negociações e adaptações entre os valores anteriores dos desenvolvedores e a maneira em que seu trabalho coletivo é organizado. São apresentados alguns resultados parciais, provenientes de um nível exploratório da pesquisa, que por sua vez objetiva o embasamento para a etapa do *survey* virtual. Realizei um levantamento das características gerais da categoria “desenvolvedores de programas e aplicativos (software)” nos microdados da amostra do Censo 2010 do IBGE; e por fim, trechos de entrevistas para ajudar na compreensão de variáveis importantes. Dessa forma, estão expostas a seguir as interpretações dos principais temas levantados pela combinação dessas fontes de informação empírica.

3.1 Ocupações Correlatas

Para as seguintes análises, a ênfase está nas diferenças entre algumas das ocupações próximas à de desenvolvimento de *software*, a fim de compreender de que maneira o desenvolvimento de *software* livre pode ser comparado. Em outras palavras, cabe saber quais características são similares, quais não são, e o que influencia as diferenças.

Com esse intuito, elaborei um banco de dados a partir dos microdados do Censo 2010, filtrando-o para levar em conta apenas as ocupações da família de ocupações “desenvolvedores e analistas de programas e aplicativos (software) e multimídia”, que possui o código da variável COD 251. A escolha pelos microdados do Censo traz como consequência a discussão sobre ocupações que compõem a Classificação de Ocupações para Pesquisas Domiciliares (COD), utilizada pelo IBGE em suas pesquisas, tais como o Censo e a PNAD. Por isso, existem algumas diferenças entre essa classificação e a tradicional Classificação Brasileira de Ocupações (CBO), usada pelo então Ministério do Trabalho e Emprego (MTE). Uma dessas diferenças reside em uma das propostas orientadoras da COD, que é a compatibilidade internacional com a CIUO-08, elaborada pela Organização Internacional do Trabalho, o que por sua vez agrupa as ocupações relacionadas às TIC numa família de ocupações muito similar àquela decorrida da COD na

Apresentação, distinguindo-a das ocupações do nível de diretores/gerentes de TI (COD 133) e de ocupações no nível técnico (COD 351), como assistência ao usuário, suporte à rede, entre outras. Já no caso da CBO, a distinção entre essas ocupações toma outra forma, já que a ocupação “programador de sistemas de informação”, que inclui atividades diversas de desenvolvimento computacional está localizada na família ocupacional “técnicos de desenvolvimento de sistemas e aplicações” (CBO 3171), é considerada de nível técnico; apenas a denominação de “engenheiros em computação” (CBO 2122) é considerada de nível superior, o que separou dois grupos próximos a partir da relação com o diploma exigido para a função e não pelas características da atividade⁹.

Retornando à questão da composição do banco de dados, considerei pertinentes alguns ajustes: eliminei os casos (36 no total) que responderam ter rendimento zero no trabalho principal (no caso o desenvolvimento de software) para que a média não fosse afetada por respondentes que não estão trabalhando de forma remunerada (já que a mesma pré-seleção foi aplicada em meu *survey*). Ademais, todos os valores de frequência já são os valores ponderados a partir dos pesos calculados pela metodologia de amostragem complexa do IBGE (CENSO, 2011), de forma a ser esperado que os valores totais representem o conjunto da população brasileira naquele ano¹⁰.

3.1.1 Características Ocupacionais

Posto isso, podemos enfim observar a proporção do número de trabalhadores(as) entre as ocupações relacionadas com o desenvolvimento de *software* na *Tabela 1*. Logo à primeira vista a ocupação nomeada “analistas de sistemas” é a que possui maior frequência, com 74% do total de trabalhadores(as) da área. No entanto, esta é a ocupação que possui características mais distintas das demais nesta família de ocupações. De acordo com um site especializado em guias

⁹ O critério do diploma de nível superior como exigência para a inserção profissional de desenvolvedores não é uma questão simples e direta, pois nem sempre é uma exigência formal, apesar de suas reconhecidas vantagens. Assim, essa se torna uma divisão extrínseca do campo, dificultando a seleção da ocupação de “desenvolvedores” seja com ou sem diploma, e que ao mesmo tempo não incluía ocupações demasiadamente distintas.

¹⁰ O universo real de respondentes do questionário da amostra na família ocupacional de desenvolvedores de *software* foi 21.371, representando uma população ponderada em 321.239.

ocupacionais, um analista de sistemas “analisa e desenvolve projetos de sistemas, levanta requisitos, mapeia processos e realiza modelagem de dados, com objetivo de estudar e implementar sistemas de acordo com as regras de negócio” (CATHO, 2015). Esse tipo de cargo envolve conhecimentos de programação de código, mas não se especializa nessa atividade pois sua função é planejar uma mediação entre o trabalho mais técnico da programação e as necessidades empresariais da companhia ou do(a) cliente, o que provavelmente justifica sua predominância no mercado de trabalho.

Tabela 1 – Ocupações na Família “Desenvolvedores e Analistas de Programas e Aplicativos (Software) e Multimídia”

	COD	Frequência	Percentual
Analistas de Sistemas	2511	239.160	74,45%
Desenvolvedores de Programas e Aplicativos (Software)	2512	21.117	6,57%
Desenvolvedores de Páginas de Internet (Web) e Multimídia	2513	4.755	1,48%
Programadores de Aplicações	2514	53.985	16,81%
Desenvolvedores e Analistas de Programas e Aplicativos (Software) e Multimídia Não Classificados Anteriormente	2519	2.221	0,69%
Total		321.239	100,00%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Quanto às outras ocupações, lembro que a ênfase desta dissertação estará na especialização acerca da ocupação chamada aqui como “desenvolvedores de programas e aplicativos (software)”, que está mais associada à atividade de programação em geral por excelência, e portanto mais próxima à reconstrução histórica realizada no capítulo anterior. Contudo, tenho alguma incerteza sobre quais as distinções entre essa ocupação e aquela rotulada como “programadores de aplicações”, já que nem a metodologia do Censo nem as especificações da CIUO-08 descrevem as ocupações tal como a CBO, mas apenas nomeiam-nas. E baseando-se apenas pelo nome as duas ocupações poderiam sobrepor características, já que ocasionalmente a diferença entre desenvolvedor e programador é sutil, com os termos se intercambiando. A interpretação mais bem sustentada é que os(as) programadores(as) estão mais concentrados na escrita do código em um ambiente em que a divisão de tarefas é bem alta e sofisticada, enquanto os(as) desenvolvedores(as) são responsáveis pelo código simultaneamente a outras tarefas menos

técnicas do projeto (INFORMANT, 2014), sem, no entanto, chegar no nível de versatilidade dos analistas de sistemas, que não estão concentrados na escrita do código. Quanto ao conceito de “aplicação”, este é geralmente entendido como um sinônimo de “aplicativo”, e formalmente seria possível enquadrar respondentes da variável “Programadores de Aplicações” na categoria “Desenvolvedores de Programas e Aplicativos (Software)”, já que os aplicativos são uma categoria de *software*. Em seu significado mais amplo, a distinção se resume ao *software* de sistema (que provê a infraestrutura tal como o sistema operacional, *drivers* de dispositivo e outros componentes funcionais) e o *software* aplicativo (que oferece a maior parte das tarefas específicas ao usuário, como processadores de texto, navegadores *web*, *games*, etc) (APPLICATION PROGRAM, 2015).

É preciso notar também que essas ocupações se referem ao trabalho considerado “principal” no caso de pessoas que possuem mais de um trabalho. Para considerar um dos trabalhos como “principal” o IBGE definiu os seguintes critérios: 1) Maior número de horas normalmente trabalhadas por semana; 2) Maior rendimento mensal habitual; 3) Trabalho que possui há mais tempo. Contudo, também revelou que dentre o conjunto desses(as) trabalhadores(as) a maioria possui apenas um trabalho, e com pouca variação entre as ocupações da família, tal como exposto na *Tabela 2*.

Tabela 2 – Número de Trabalhos por Ocupação

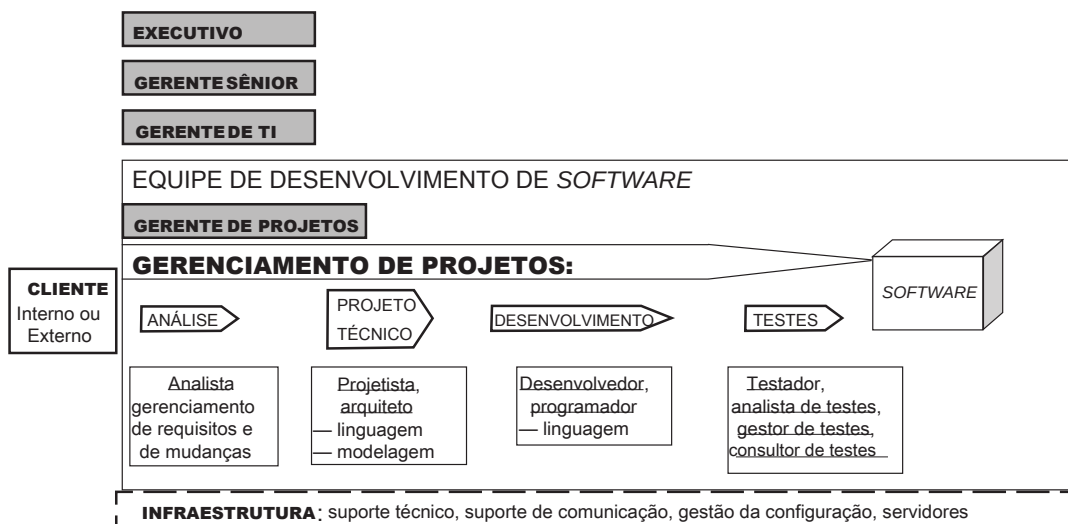
Ocupação	Quantos trabalhos tinha		Total
	Um	Dois ou mais	
Analistas de Sistemas	228.272	10.888	239.160
Percentual (%) da linha	95,45%	4,55%	100%
Desenvolvedores de Programas e Aplicativos (Software)	19.508	1.609	21.117
% da linha	92,38%	7,62%	100%
Desenvolvedores de Páginas de Internet [...]	4.330	426	4.756
% da linha	91,04%	8,96%	100%
Programadores de Aplicações	51.315	2.670	53.985
% da linha	95,05%	4,95%	100%
Desenvolvedores [...] Não Classificados Anteriormente	2.188	33	2.221
% da linha	98,51%	1,49%	100%
Total	305.613	15.626	321.239
% da linha	95,14%	4,89%	100%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Podemos inferir a partir da *Tabela 2* que algumas dessas ocupações possibilitam um pouco melhor uma noção centrada de identidade ocupacional já que um número maior de respondentes se concentra em apenas um trabalho, seja outro vínculo no mesmo tipo de ocupação ou em uma ocupação diferente. No caso, os “Desenvolvedores de Programas e Aplicativos (Software)” e os “Desenvolvedores de Páginas de Internet (Web) e Multimídia” possuem maior chance de buscarem outro trabalho.

E para compreender melhor as relações entre essas ocupações, Rosenfield e Mossi (2011) apresentaram um esquema gráfico da estrutura organizacional das posições ocupacionais no processo de desenvolvimento do *software* como um produto, tal como exposto na *Figura 1*. Este esquema se refere aos casos em que há uma empresa de tamanho razoável, que permite ou valoriza a divisão de tarefas com maior especialidades; enquanto isso, é comum que microempresas ou *freelancers* exerçam seu trabalho levando em conta muitas ou até todas essas atividades. As autoras completam: “Subjetivamente, a organização horizontal e em rede demanda uma postura de iniciativa, colaboração e empenhamento do trabalhador; ao mesmo tempo em que a flexibilidade estratégica, que visa à adaptação da empresa à variação do número de clientes e da carga de trabalho, torna-o mais vulnerável a tais mudanças.” (p. 86).

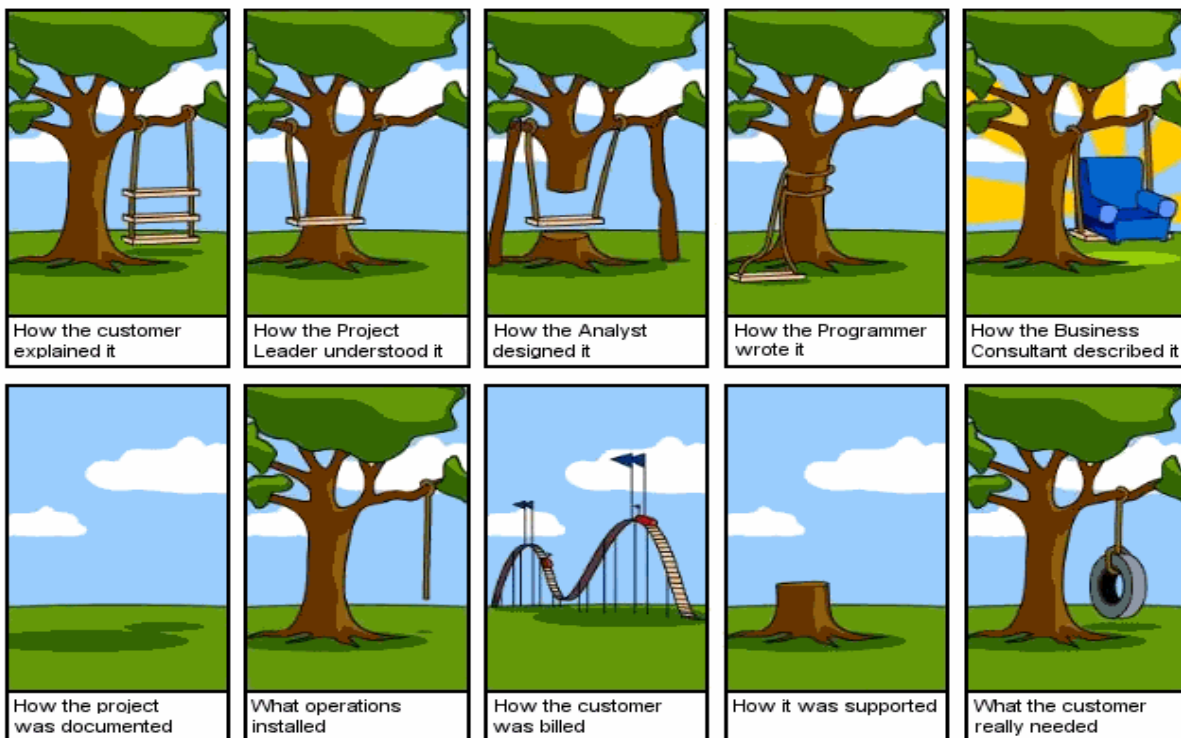
Figura 1 – Estrutura organizacional de uma empresa de TI



Fonte: ROSENFIED; MOSSI, 2011, p. 85.

Mas além de pontuar como essas ocupações se complementam para apresentar o *software* como produto final, vale observar um pouco sobre as tensões internas ao desenvolvimento de um projeto, conforme ilustra satiricamente a *Figura 2*. Minha ênfase é apontar para possíveis pontos de conflito entre a cultura organizacional imposta pela direção e os problemas de implementação dos(as) trabalhadores(as).

Figura 2 – Tirinha humorística sobre o papel das ocupações envolvidas na engenharia de software



Fonte: <<http://blog.codinghorror.com/on-software-engineering/>>. Acesso em: 20 out. 2015.

Notas: Tradução dos quadros: Como o cliente explicou; Como o líder do projeto entendeu; Como o analista de sistemas projetou; Como o programador escreveu; Como o consultor de negócios descreveu; Como o projeto foi documentado; Quais operações foram instaladas; Como o cliente foi cobrado; Como foi o suporte técnico; O que o cliente realmente precisava.

A ação de incluir uma tirinha em tom humorístico em um texto com propósito acadêmico exige um cuidado extra na interpretação: evidentemente esse tipo de narrativa não deve ser entendida de forma literal, pois um dos elementos-chave do humor é justamente a aposta no exagero para fazer rir. Entretanto nem tudo o que o humor traz é necessariamente fantasia, pois a “graça” também se baseia na referência a algum elemento social previamente conhecido, sendo muitas vezes uma

forma socialmente justificada de exceção aos temas que não se pode falar¹¹. Além disso, o humor é um elemento central para a sociabilidade *hacker* já há várias décadas, onde frequentemente se chega ao cúmulo de inserir piadas (algumas vezes compreendidas apenas por um grupo muito seletivo) no código-fonte, na documentação do programa, e até no nome dos projetos¹².

3.1.2 Escolaridade

Em acréscimo a essas observações, vale salientar também algumas características gerais de desenvolvedores/as de *software* que assinalam para a estabelecimento de diferentes perfis socioeconômicos entre as ocupações correlatas. Tomemos por início as proporções de escolaridade na *Tabela 3*.

Tabela 3 – Ocupações e Escolaridade

Ocupação	Nível de Instrução				Total
	Sem instrução e fundamental incompleto	Fundamental completo e médio incompleto	Médio completo e superior incompleto	Superior completo	
Analistas de Sistemas	6.313	7.658	62.992	161.703	238.667
% da linha	2,65%	3,21%	26,39%	67,75%	100%
Desenvolvedores de Programas e Aplicativos [...]	555	944	6.641	12.879	21.019
% da linha	2,64%	4,49%	31,59%	61,27%	100%
Desenvolvedores de Páginas de Internet [...]	223	495	2.779	1.259	4.756
% da linha	4,69%	10,41%	58,43%	26,47%	100%
Programadores de Aplicações	2.028	3.144	25.814	22.764	53.750
% da linha	3,77%	5,85%	48%	42,35%	100%
Desenvolvedores [...] Não Classificados Anteriormente	28	192	914	1.065	2.199
% da linha	1,29%	8,71%	41,56%	48,43%	100%
Total	9.149	12.433	99.139	199.670	320.391
% da linha	2,86%	3,88%	30,94%	62,32%	100%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

¹¹ Seja para aliviar algum fardo (como o humor sobre seus próprios problemas pessoais ou de trabalho), seja para disfarçar processos de exclusão (por exemplo com o sexismo, homofobia, racismo, etc), ou ainda para satirizar grupos e pessoas politicamente privilegiadas.

¹² Entre os entusiastas do software livre, é clássica a referência ao nome do projeto fundado por Richard Stallman: o projeto GNU, que é uma sigla autorrecursiva, pois significa "GNU is NOT UNIX", ou seja o significado da primeira letra remete ao próprio nome completo.

Notas: Para esta tabela foram excluídos os casos em que o nível de instrução correspondia a “não determinado”, o que equivale a 0,26% dos casos totais. As diferenças entre soma de parcelas e respectivos totais são provenientes do critério de arredondamento.

A informação principal evidenciada nessa tabela é, na verdade, uma ênfase das diferenças principais entre as ocupações discutidas anteriormente: conforme uma ocupação dispõe de maior estabilidade no mercado de trabalho ela pode exigir um nível maior de escolaridade. Isso se manifesta com a ocupação “Analistas de Sistemas” apresentando uma percentagem maior de seus(suas) trabalhadores(as) com ensino superior em um grande contraste com a ocupação “Desenvolvedores de Páginas de Internet (Web) e Multimídia” com uma taxa baixa de graduados.

No que diz respeito à exigência de diploma de curso superior, tanto em trechos de entrevistas realizadas por mim quanto em casos da literatura (MACEDO, 2011, p. 70-74) o diploma aparece constantemente nas falas de desenvolvedores como um fator valorizado e desejável, porém não obrigatório:

— *Você disse que é formado em outra área, Engenharia Elétrica, você acha que em geral, seja em computação ou software livre, diplomas de ensino superior ou diplomas de cursos específicos não são requisitos? Ou outras competências são mais importantes para conseguir um trabalho na área?*

— *Eu acho que ter a formação é importante, mas não é uma exigência. Eu acredito que as pessoas conseguem trabalhar... Eu tenho um pouquinho de formação na área de computação, mas acredito que as pessoas conseguem trabalhar sem precisar ter uma formação superior mesmo. Eu acho que principalmente na parte *software* livre a maioria das pessoas é autodidata e isso normalmente é mais do que suficiente. A gente tem alguns casos que eu considero até atípicos, mas casos famosos na comunidade de advogado, médico, que fazem contribuições acima de outros desenvolvedores tanto em quantidade quanto em qualidade. Mas, são casos atípicos. Depende muito da pessoa ter a perseverança de ir lá e resolver o problema. (Geeko¹³)*

Ainda assim, o que parece ser a questão para esses(as) trabalhadores(as) é a disputa entre qual o diploma pode ser aceito além de cursos relacionados diretamente com a computação, e não tanto sobre não possuir diploma algum. Digo disputa por remeter à discussão sobre regulamentação das profissões de TI e as diversas posições sobre o assunto, onde alguns argumentam a favor da proteção do campo para esses profissionais graduados e especializados, e outros identifiquem

¹³ Os codinomes das pessoas entrevistadas foram escolhidos pelo pesquisador, buscando apenas o anonimato, a partir de nomes de mascotes conhecidos no *software* livre, sem refletir relações necessárias com esses programas ou com opiniões específicas.

como um ponto positivo da área ser capaz de captar pessoas com formações mais diversas (NUNES; MELLO; SOUZA, 2015).

É interessante notar também a participação (mesmo que proporcionalmente bem pequena) de respondentes que afirmaram possuir nenhuma instrução formal ou escolaridade até o fundamental incompleto nesta família de ocupações que exige um grau bastante elevado de conhecimentos técnicos e especializados. Porém, é difícil interpretar se isso se deve a algum erro metodológico da coleta do Censo ou se esses respondentes na verdade possuem um perfil distinto daquele que busco denominar aqui como “desenvolvedores de *software*”, pertencendo a um lado mais “técnico” das ocupações de TI (levando em conta o problema das definições das variáveis ocupacionais COD e CBO citada anteriormente). De qualquer modo, no *survey* produzido por mim não aparecem casos atípicos chamando a atenção, como visto na *Tabela 4*.

Tabela 4 – Escolaridade de Desenvolvedores de *Software*

Escolaridade	Frequência	Percentual
Sem Instrução Formal ou Fundamental Incompleto	0	0,00%
Fundamental Completo	1	0,38%
Médio Completo	54	20,53%
Superior Completo	123	46,77%
Pós-Graduação	85	32,32%
Total	263	100,00%

Fonte: Questionário aplicado pelo pesquisador no período de 8 de Fevereiro a 17 de Março de 2015.

3.1.3 Posição na Ocupação

Outra diferença entre as ocupações foi notada entre a variável “posição na ocupação”, definida pelo IBGE como “a relação de trabalho existente entre a pessoa e o empreendimento em que trabalhava” (CENSO, 2011, p. 49), apresentada na *Tabela 5*. Podemos observar que entre os/as analistas de sistema mais uma vez está a indicação de maior estabilidade no trabalho, já que possuem o maior índice de trabalhadores(as) com vínculo de carteira de trabalho assinada, comparando-se com as demais ocupações da família, além de possuírem proporcionalmente menos pessoas empregadas sem carteira de trabalho assinada. Enquanto isso, a categoria “Desenvolvedores de Páginas de Internet (Web) e Multimídia” tem menos da metade

de seus integrantes com carteira de trabalho assinada, e uma grande cota de pessoas trabalhando por “conta própria”. Com isso, os(as) desenvolvedores(as) de *software* em geral estão relativamente equilibrados entre as duas tendências, com uma possibilidade razoável de trabalho no mercado formal, mas também em um número expressivo de formas flexibilizadas, cada vez mais valorizado pelo capitalismo contemporâneo. Essas formas variam entre o trabalho como *freelancer* ou como pequenos empreendedores com CNPJ (CASTRO, 2013), algumas vezes prestando serviços específicos para empresas por sua vez terceirizadas em relação à empresa principal. Outra possibilidade flexibilizada é a categoria de autoempreendedores, que transitam em uma “zona cinzenta” entre formal e informal, estável e instável, e por entre essas categorias tradicionais (ROSENFELD, 2015).

Tabela 5 – Ocupações e Posição na Ocupação

Ocupação	Nesse Trabalho Era ¹⁴						Total
	Empregado com carteira de trabalho assinada	Empregado pelo regime jurídico dos funcionários públicos	Empregado sem carteira de trabalho assinada	Conta própria	Empregador	Não remunerado	
Analistas de Sistemas	179.764	10.131	12.383	33.031	3.614	238	239.160
% da linha	75,16%	4,24%	5,18%	13,81%	1,51%	0,1%	100%
Desenvolvedores de Programas e Aplicativos [...]	13.711	491	1.964	4.427	481	44	21.117
% da linha	64,93%	2,33%	9,3%	20,96%	2,28%	0,21%	100%
Desenvolvedores de Páginas de Internet [...]	2.071	75	884	1.644	68	15	4.756
% da linha	43,54%	1,57%	18,59%	34,56%	1,42%	0,32%	100%
Programadores de Aplicações	36.932	1.309	4.965	9.888	854	38	53.986
% da linha	68,41%	2,42%	9,20%	18,32%	1,58%	0,07%	100%
[...] Não Classificados [...]	1.634	231	168	138	40	10	2.221
% da linha	73,58%	10,42%	7,56%	6,21%	1,78%	0,46%	100%
Total	234.112	12.238	20.363	49.127	5.056	344	321.238
% da linha	72,88%	3,81%	6,34%	15,29%	1,57%	0,11%	100%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Nota: A diferença entre soma de parcelas e seus totais ocorre devido ao arredondamento.

¹⁴ Durante a semana de referência de 25 a 31 de julho de 2010.

3.2 Variáveis Sociológicas Clássicas

Para esta seção compus outro banco de dados derivado daquele baseado no Censo 2010 herdando os mesmos ajustes metodológicos; porém, agora desconsiderando a ocupação “analista de sistemas”, já que possuem uma frequência muito maior do que as outras, além de apresentar características levemente distintas do foco nos programadores e desenvolvedores, conforme a seção anterior mostrou. Aliás, as seguintes análises ignoram a comparação entre as ocupações porque após uma avaliação preliminar, não considerei a diferença entre as ocupações relevantes para explicar as seguintes questões, focando agora em causas mais diversas.

Vamos então começar apresentando a questão relativa à renda do trabalho, uma variável importante para várias interpretações sociológicas, já que permite não apenas detectar o poder de compra imediato de uma pessoa (que por sua vez seria melhor mensurado através de outras variáveis complementares, como o rendimento domiciliar *per capita*), mas principalmente reflete a valorização social de uma ocupação, também em uma possível comparação com outras ocupações em relação ao mercado de trabalho. Além disso, segundo a teoria de Bourdieu (2008), o rendimento (ou neste caso, o “capital econômico”) pode se converter em várias formas de “capital simbólico”, e também inversamente, os “capitais simbólicos” frequentemente constituem privilégios sociais para se adquirir posições de *status* e ocupações com maior rendimento. Dessa forma, a *Tabela 6* expõe um panorama generalizante do rendimento de desenvolvedores de *software* no Brasil em 2010.

Tabela 6 – Rendimento de desenvolvedores de *software* (exceto analistas de sistemas) em Reais e Salários Mínimos de Julho de 2010

	Medidas de Distribuição Central					
	Média	Moda	Quartil Inferior	Mediana	Quartil Superior	Desvio Padrão
Rendimento no Trabalho Principal em Reais de Julho de 2010	2.410,85	2.000	1.000	1.800	3.000	2.247,686
Rendimento no Trabalho Principal em Nº de Salários Mínimos de Julho de 2010	4,73	4	1,96	3,53	5,88	4,407

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

A fim de aprofundarmos neste assunto, será necessário se remeter à comparação dos rendimentos entre dois estratos clássicos de distinção social: o de sexo/gênero e o de cor/raça. Infelizmente, as clivagens étnicas e de gênero na sociedade brasileira ainda persistem em diferentes setores socioeconômicos, refletindo em diferenças de rendimento, tendo por causa preconceitos abertos ou velados, dificuldades específicas de capacitação para competição justa, especialização em campos mais ou menos valorizados, e ainda uma complexa combinação desses e outros fatores. Tendo isso em mente, e antes de tratar em particular dessas desigualdades, é oportuno mostrar a proporção de participação dessas categorias nas ocupações de desenvolvimento de *software*.

3.2.1 Gênero

Ao fazer uma rápida observação sobre os percentuais de participação masculina e feminina nas ocupações pesquisadas, fica comprovada a predominância masculina dentre o número total de trabalhadores(as) do setor: segundo o Censo 2010, 84,8% (69.629) são homens, enquanto 15,2% (12.450) são mulheres, o que não deve ser entendido como um mero significado numérico, já que possui um importante peso simbólico. Essa disparidade não chega a ser uma particularidade brasileira, pois proporções similares são encontradas em estudos internacionais: entre uma amostra de desenvolvedores de *software* livre provenientes de diversos países (LIBRESOFT, 2014)¹⁵ e dentre os casos “válidos”, 91.71% dos respondentes foram homens e apenas 8.29% de mulheres.

Devemos levar em conta que, mesmo que eu não explique aqui as causas diretas da baixa participação feminina, vários possíveis fatores são conhecidos (cf. NUNES, 2016), e plausivelmente coexistem: ao focar nos processos de socialização, poderemos aludir à valorização social que incentiva os homens muito mais frequentemente do que as mulheres a se encontrarem em espaços considerados técnicos, onde há uma relação com a indústria e com máquinas, enquanto a presença das mulheres é largamente concentrada no setor de serviços, onde são

¹⁵ Tendo como base os microdados do “FLOSS Survey 2013” – aplicado internacionalmente em ambiente virtual e direcionado a pessoas envolvidas com *software* livre e de código aberto – elaborei um banco de dados incluindo apenas pessoas cuja profissão foi considerada “engenheiro(a) de *software*” ou “programador(a)”.

mais valorizadas as relações interpessoais (DAUNE-RICHARD, 2003), o que leva homens e mulheres a escolherem carreiras e especializações distintas desde a infância.

Além disso, conforme argumentei em uma pesquisa sobre os processos de socialização entre músicos e musicistas (MELLO, 2011), é comum que a divisão sexual do trabalho também se dê a partir de escolhas conscientes de mulheres para setores em que elas possam obter maior estabilidade no vínculo de trabalho e maior flexibilidade em relação à carga de trabalhos domésticos não-remunerados que ainda hoje e mesmo em setores de alta escolaridade persiste em ter uma quase insignificante divisão dessas tarefas com companheiros do sexo masculino.

Finalmente, uma grande barreira se interpõe a mulheres a partir do preconceito explícito em considerar as mulheres como menos capazes de exercer trabalhos de alta tecnologia. Isso acarreta possivelmente dificuldades de contratação, levando em consideração a tendência comum de desvalorização dos diplomas possuídos pelas mulheres: “quando elas têm o mesmo nível de escolaridade dos homens, ascendem globalmente a empregos menos qualificados e, quando exercem as mesmas categorias socioprofissionais, elas são superdiplomadas” (Daune-Richard, 2003). Ainda mais, uma visível consequência se configura nas numerosas desistências ao longo dos anos, desde a formação e entrada no mercado de trabalho até os empecilhos para altas posições na carreira. Nesse sentido, de acordo com uma reportagem estadunidense (LIEN, 2015) os setores de TI dos EUA estão sofrendo uma diminuição em seus números de cargos ocupados por conta de uma expressiva quantidade de mulheres que está abandonando a carreira na área.

Posto isso, podemos enfim observar uma disparidade de rendimentos entre homens e mulheres na *Tabela 7*:

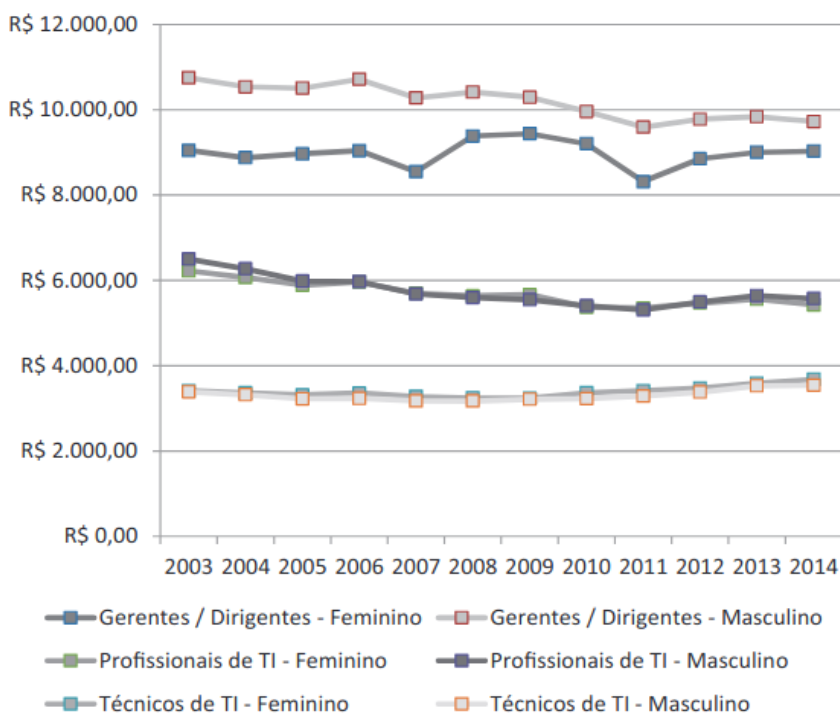
Tabela 7 – Rendimento de desenvolvedores de software (exceto analistas de sistemas) em reais e salários-mínimos de julho de 2010 segundo gênero

		Medidas de Distribuição Central					
		Média	Moda	Quartil Inferior	Mediana	Quartil Superior	Máximo
Rendimento no Trabalho Principal	Masculino	2.462	2.000	1.020	1.800	3.000	30.000
	Feminino	2.126	2.000	851	1.500	2.500	16.000
Rendimento no Trabalho Principal em N° de Salários Mínimos	Masculino	4,83	4	2	3,53	5,88	58,82
	Feminino	4,17	4	1,67	2,94	4,9	31,37

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

No entanto, vale ressaltar que a desigualdade salarial de gênero é mais acentuada nos cargos mais elevados no setor de TI (NUNES, 2016), tal como mostra o *Gráfico 1*, onde os grupos ocupacionais de dirigentes e gerentes apresentam uma maior discrepância entre homens e mulheres na mesma função, enquanto os setores mais baixos possuem rendimentos relativamente mais próximos.

Gráfico 1 – Variação de rendimentos por sexo e grupos ocupacionais do setor de TI: 2003 a 2014 – Brasil



Fonte: NUNES, 2016, construído a partir da RAIS (MTE, online pelo aplicativo Dardo).

3.2.2 Região Geográfica

Antes de chegarmos à questão sobre como uma outra categoria de desigualdade social, a de cor ou raça, impacta sobre o rendimento no trabalho principal, vamos observar uma terceira variável possivelmente intercalada nesse processo: em quais regiões geográficas se concentram os(as) desenvolvedores(as) de *software*. A *Tabela 8* demonstra a predominância das regiões Sudeste e Sul como focos do mercado de trabalho nessa área, por conterem, combinadas, 78,6% dos postos de trabalho.

**Tabela 8 – Desenvolvedores de *software*
(exceto analistas de sistemas) segundo grandes regiões**

Região Geográfica	Frequência	Percentual
Centro-Oeste	6.632	8,1%
Nordeste	8.484	10,3%
Norte	2.439	3,0%
Sudeste	45.310	55,2%
Sul	19.214	23,4%
Total	82.079	100,0%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

No entanto, essa distribuição está ligeiramente fora de proporção em relação à população total do país, onde, segundo dados oficiais do Censo 2010 (IBGE, 2011), a região mais populosa é o Sudeste (42,1%), mas a segunda mais populosa é o Nordeste (27,8%), sendo seguida atrás pelo Sul (14,4%), o Norte (8,3%), e o Centro-Oeste (7,4%). Isto é, a região Nordeste possui uma sub-representação de desenvolvedores(as) de *software* (lembrando que considero apenas aqueles que exercem a função como seu trabalho principal). Um motivo concebível para esse fato é a concentração de polos tecnológicos nessas regiões que concentram o poder econômico do país. É interessante, portanto, que a localização física seja um fator tão relevante para a constituição do mercado de trabalho em uma área em que um número considerável de pessoas trabalha remotamente pela Internet, não necessitando comparecer a um escritório na empresa matriz, tal como será discutido no próximo capítulo.

3.2.3 Cor/Raça

Por fim, podemos deter a atenção sobre a desigualdade racial entre desenvolvedores(a) de *software*. A princípio é necessário observar as proporções de frequência por meio da *Tabela 9*, onde estão dispostas as categorias conforme nomeadas pelo IBGE. Logo em seguida apresento a *Tabela 10*, onde escolhi agrupar essas categorias numa variável binária, em que a categoria “Branca” está agrupando os rótulos originais “branca” e “amarela”, e a categoria “Negra” contém os rótulos “preta”, “parda” e “indígena”.

**Tabela 9 – Desenvolvedores de *software*
(exceto analistas) por cor ou raça agrupada**

Cor ou Raça	Frequência	Percentual
Branca	61.119	74,46%
Preta	3.034	3,70%
Amarela	1.906	2,32%
Parda	16.011	19,51%
Indígena	9	0,01%
Total	82.079	100,00%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

A escolha dessa metodologia se inspira em estudos que focam na distinção étnica brasileira de um ponto de vista mais dualista (TELLES, 2003; HASENBALG, 2005), ou seja, não levando em conta as discussões sobre as distinções identitárias entre “pardos” e “pretos”. A categoria foi então denominada como “negra”, em vez de “não-branca”, seguindo a tendência mais comum das pesquisas de raça e cor no mercado de trabalho, apesar de que o enfoque identitário ainda não foi explorado qualitativamente o suficiente.

**Tabela 10 – Desenvolvedores de *software*
(exceto analistas de sistemas) por cor ou raça agrupada**

	Frequência	Percentual
Branca	63.026	76,79%
Negra	19.053	23,21%
Total	82.079	100,00%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Dessa forma, torna-se mais claro que o conjunto de “brancos” é majoritário no setor, quando nos recordamos de que no conjunto da população brasileira um pouco mais da metade poderia ser considerada “negra”. Ainda que essa desproporção seja comum entre ocupações mais valorizadas, como as que exigem maior escolaridade, por conta de obstáculos diretos e indiretos no acesso ao estudo, esse não foi um fator com significância suficiente para influenciar na desproporção, já que a diferença entre o número de pessoas com diploma de nível superior e médio entre as cores/raças não varia tanto nesta categoria ocupacional. Uma explicação que creio ser mais plausível neste caso é a comparação com as proporções de raça/cor entre as regiões geográficas, já que há um maior número de desenvolvedores(as) nas regiões Sudeste e Sul, como mostrado anteriormente, que por sua vez possuem uma proporção maior de “brancos”, ao contrário da região Nordeste, por exemplo, o que pode ser observado na *Tabela 11*. Com isso, é demonstrável que a sub-representação de “negros” seja consequência da desproporção geográfica.

Tabela 11 – Cor ou Raça da População Brasileira por Grande Região Geográfica

Região	Cor ou Raça Agrupada			
	“Branca”		“Negra”	
	Frequência	% da Linha	Frequência	% da Linha
Centro-Oeste	6.046.866	43,0%	8.008.714	57,0%
Nordeste	16.119.855	30,4%	36.954.871	69,6%
Norte	3.862.865	24,4%	11.998.191	75,6%
Sudeste	45.055.249	56,1%	35.287.229	43,9%
Sul	21.641.799	79,0%	5.744.108	21,0%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Posto isso, resta analisar como o rendimento é afetado pela variável de cor/raça. Utilizando a variável de cor/raça agrupada na *Tabela 12*, torna-se evidente mais um fator de desigualdade, já que o conjunto dos(as) desenvolvedores(as) “negros” possuem desvantagens de renda em relação aos “brancos” dos setores inferiores aos superiores; uma disparidade de renda que é ligeiramente mais acentuada que aquela que acontece entre homens e mulheres.

Tabela 12 – Rendimento de desenvolvedores de *software* (exceto analistas) em Reais e Salários Mínimos de Julho de 2010 segundo cor/raça agrupada

		Medidas de Distribuição Central					
		Média	Moda	Quartil Inferior	Mediana	Quartil Superior	Máximo
Rendimento no Trabalho Principal	Branca	2.535	2.000	1.020	1.900	3.000	30.000
	Negra	2.000	1.500	950	1.500	2.500	16.000
Rendimento no Trabalho Principal em N° de Salários Mínimos	Branca	4,97	4	2,00	3,73	5,88	58,82
	Negra	3,92	3	1,86	2,94	4,90	31,37

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Vem à mente então a pergunta se as diferenças de peso geográfico também não estão desequilibrando essa relação entre a cor/raça e a renda, já que o número de pessoas está ponderado por esse fator. Para explorar isso, antes é preciso lembrar de que como há uma terceira variável na análise ainda não sabemos o suficiente para concluir qual a direção da relação: se é a desigualdade geográfica é o principal fator da diferença de rendimentos (influenciando também a cor/raça) ou se é a cor/raça o fator preponderante de discriminação (o que ocorre simultaneamente à questão espacial). Assim sendo, podemos observar na *Tabela 13* que de fato a localização dos desenvolvedores(as) entre as grandes regiões geográficas impacta sobre a variação de rendimentos do trabalho.

Tabela 13 – Rendimento de desenvolvedores de *software* (exceto analistas) por Grande Região Geográfica

Região	Medidas de Distribuição Central					
	Média	Moda	Quartil Inferior	Mediana	Quartil Superior	Máximo
Centro-Oeste	2.396	2.000	1.020	1.800	2.900	30.000
Nordeste	1.892	510	700	1.500	2.100	18.000
Norte	2.105	510	800	1.500	2.500	11.800
Sudeste	2.671	2.000	1.200	2.000	3.200	30.000
Sul	2.070	2.000	1.000	1.500	2.500	25.000

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Logo, resta saber se a desigualdade de renda em relação à cor/raça é um efeito exclusivo da localização geográfica, ou se também é concomitante a outras influências em cada região. Por isso, isolarei a análise a seguir para os casos

referentes aos habitantes do Sul e Sudeste, para assim ver se as desigualdades se limitam às diferenças entre as grandes regiões, ou se elas se reproduzem nos diferentes contextos. Na *Tabela 14* está demonstrada a reprodução dos padrões de desigualdade de renda entre “brancos” e “negros” levando em conta o contexto do Sul e Sudeste, onde presumivelmente haveria melhores condições de possibilitar postos de trabalho de alto rendimento para um maior número de pessoas. Podemos ver neste caso apenas uma ligeira diminuição do *gap* racial.

Tabela 14 – Rendimento de desenvolvedores de *software* (exceto analistas) do Sul e Sudeste em Reais e Salários Mínimos de Julho de 2010 segundo cor/raça agrupada

		Medidas de Distribuição Central					
		Média	Moda	Quartil Inferior	Mediana	Quartil Superior	Máximo
Rendimento no Trabalho Principal	Branca	2.564	2.000	1.100	2.000	3.000	30.000
	Negra	2.133	1.500	1000	1.600	2.700	13.000
Rendimento no Trabalho Principal em N° de Salários Mínimos	Branca	5,03	4	2,16	3,92	5,88	58,82
	Negra	4,18	3	1,96	3,14	5,29	25,49

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

A partir daí, algumas outras hipóteses surgem para tentar explicar esse fenômeno, como a de que seriam as diferenças do acesso à escolaridade entre “brancos” e “negros” que fundamentaria um privilégio de acessos a determinadas posições na ocupação ou cargos de maior rendimento. Tendo isso em mente, a *Tabela 15* demonstra que existe de fato uma proporção levemente desfavorável aos “negros” em comparação com os “brancos” no número de pessoas com nível de instrução mais baixo; no entanto, a diferença não chega a ser tão expressiva quanto se poderia supor inicialmente, no que diz respeito ao número de pessoas com o diploma de nível superior.

Tabela 15 – Escolaridade de desenvolvedores de *software* (exceto analistas) do Sul e Sudeste segundo cor/raça agrupada

Cor ou Raça Agrupada	Nível de Escolaridade							
	Sem instrução		Fundam. Completo		Médio Completo		Superior Completo	
	Frequência	% da Linha	Frequência	% da Linha	Frequência	% da Linha	Frequência	% da Linha
Branca	1.546	2,9%	2.938	5,5%	23.110	43,2%	25.897	48,4%
Negra	723	6,7%	815	7,6%	4.670	43,5%	4.527	42,2%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Complementando esse raciocínio, já que a diferença de rendimentos por raça parece ser influenciada pela escolaridade, ou seja, a capacidade de inserção em posições de diferentes valores no mercado de trabalho, cabe pôr em prova mais uma variável nessa conjunção: a relação direta entre o rendimento e as ocupações relacionadas ao desenvolvimento de *software*, tal como descrito na *Tabela 16*:

Tabela 16 – Frequência de desenvolvedores de *software* (exceto analistas) do Sul e Sudeste em cada ocupação segundo cor ou raça agrupada

Cor ou Raça Agrupada	Ocupações							
	Desenvolvedores de Programas e Aplicativos [...]		Desenvolvedores de Páginas de Internet [...]		Programadores de Aplicações		[...] Não Classificados Anteriormente	
	Frequência	% da Linha	Frequência	% da Linha	Frequência	% da Linha	Frequência	% da Linha
Branca	14.617	27,2%	2.803	5,2%	34.830	64,8%	1.461	2,7%
Negra	2.370	21,9%	719	6,6%	7.389	68,3%	336	3,1%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Aqui é possível perceber que entre os(as) desenvolvedores(as) “brancos(as)” e “negros(as)” há também uma pequena margem de diferença proporcional entre as ocupações do setor. As ocupações que identifiquei anteriormente como possuindo mais características de precariedade no trabalho são aquelas com a participação relativamente maior de “negros”: os(as) desenvolvedores(as) de páginas de Internet (Web) e multimídia; e os(as) desenvolvedores(as) não classificados anteriormente.

Além disso, não é apenas o número de desenvolvedores(as) “branco(as)” e “negros(as)” que varia em cada uma das ocupações, o que supostamente influenciaria na diferença das médias entre as duas categorias raciais; mas a própria renda é diretamente desproporcional quando enfocamos cada uma das ocupações.

Ou seja, a desigualdade não é simplesmente causada por diferenças de formação ou de posicionamento em ocupações menos valorizadas, mas ela se reproduz em praticamente todas as instâncias. É possível examinar na *Tabela 17* que nas quatro ocupações os “brancos” têm um rendimento maior do que os “negros”, inclusive em um *gap* mais amplo do que quando observamos os rendimentos de forma mais geral anteriormente. Tendo isso em vista, concluo que a questão da diferença entre rendimentos se compõe a partir de múltiplos fatores sociais, sejam paralelos ou confluentes.

Tabela 17 – Rendimento Médio em Reais de 2010 de desenvolvedores de *software* (exceto analistas) do Sul e Sudeste em cada ocupação segundo cor ou raça agrupada

Cor ou Raça Agrupada	Média de Rendimento no Trabalho Principal por Ocupações			
	Desenvolvedores de Programas e Aplicativos [...]	Desenvolvedores de Páginas de Internet [...]	Programadores de Aplicações	[...] Não Classificados Anteriormente
Branca	3467	2028	2214	2914
Negra	2581	1382	2075	1844

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Por conseguinte, pudemos ver a reprodução de processos sociais mais amplos no setor específico do desenvolvimento de *software*, o que pode significar uma capacidade de influência mais branda da cultura *hacker* em causar mudanças nos padrões de trabalho, já que algumas barreiras sociais se manifestam claramente, como o acesso limitado do campo às mulheres e às pessoas “negras”. No próximo capítulo explorarei os limites e potenciais das especificidades do *software* livre em fazer frente aos processos sociais mais amplos da “sociedade informacional” contemporânea.

4 Perfis do Desenvolvimento de *Software*

Este capítulo explora os resultados principais do *survey* virtual conduzido por mim com o objetivo de defrontar a hipótese central da pesquisa: a relação hipotética entre uma identidade ou socialização *hacker* e as especificidades de uma cultura do trabalho no desenvolvimento de *software* livre. Com esse intuito, as próximas seções tratam de esclarecer algumas características do trabalho no desenvolvimento de *software* e quais são os aspectos comuns dessa ocupação, considerando tanto as características pessoais quanto as condições de trabalho e observando de que forma podem ser interpretadas sociologicamente.

Em seguida, a análise se concentra nas especificidades do desenvolvimento de *software* livre e de código aberto, comparando características encontradas entre aqueles(as) que não desenvolvem SLCA. Com base nessas considerações, finalizarei o estudo com um exame de mais algumas relações hipotéticas entre os valores políticos, a autoidentificação ou não com a noção de *hacker* e em que medida os diferentes valores sociais no campo influenciam a organização coletiva do trabalho.

4.1 Socialização e Preferências Pessoais

Ao comparar os resultados das variáveis exploradas a partir do Censo 2010 no capítulo anterior com as frequências correspondentes no *survey* podemos observar que as mesmas tendências populacionais se repetem: 90,1% de homens (N=237) em relação a 26 mulheres (N=26); partindo do mesmo agrupamento de cor/raça adotado anteriormente, há 71,5% de população “branca” (N=178) contra 28,5% de “negra” (N=71); e ainda 64,8% dos/das respondentes residem nas regiões Sul e Sudeste (N=160) enquanto 35,2% dos demais residindo nas regiões Norte, Nordeste e Centro-Oeste (N=87). Com isso é possível considerar que o *survey* demonstrou uma boa representatividade em relação à população de desenvolvedores de *software* no Brasil.

Assim, podemos seguir para a observação da frequência dos principais grupos religiosos do Brasil entre o grupo estudado. Tal como exposto na *Tabela 18*, há uma distribuição incomum de pessoas “sem religião” entre os(as) desenvolvedores(as) de

software, representando a maioria desse grupo (52%), considerando que na população brasileira a parcela correspondente chega apenas a 8%, incluindo ateísmo, agnosticismo e demais pessoas que possuem alguma espécie de religiosidade sem se orientar por uma instituição religiosa em particular.

Tabela 18 – Religião agrupada de desenvolvedores(as) de *software* em comparação com proporção das religiões na população brasileira

Religião	Desenvolvedores(as) de Software 2015		População Brasil 2010
	Frequência (N)	Percentual (%)	Percentual (%)
Católica Apostólica Romana	51	20,9%	64,6%
Evangélicas	36	14,8%	22,2%
Espírita (Kardecista)	11	4,5%	2,0%
Umbanda e Candomblé	4	1,6%	0,3%
Sem Religião	127	52,0%	8,0%
Outras religiosidades	15	6,1%	2,7%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015; Censo Demográfico 2010 (IBGE, 2012).

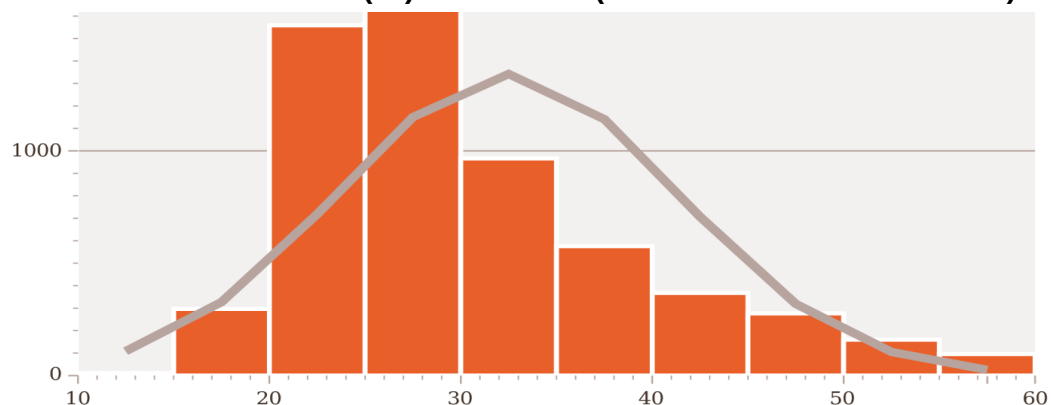
Notas: Nesta tabela foram considerados apenas os casos válidos: ignorando a opção do Censo “não sabe/não declarou” que teve valor de 0,1%, enquanto no *survey* 19 casos (7,2% do total) não responderam a questão.

Para ser mais preciso, a categoria “sem religião” agregou os seguintes valores originalmente no *survey*: “acredito em Deus, mas não tenho uma religião” com 19,4% do total de casos válidos; “ateísmo ou agnosticismo” com 28,1%; e “sem definição” com 2%. Isso aponta para uma diferença importante na socialização de desenvolvedores e desenvolvedoras de *software*, pois a religião não foi um fator importante na trajetória de vida da maioria desse grupo. Levando em conta que a quase totalidade da população brasileira é socializada inicialmente em alguma religião, deduzo que é proeminente no grupo de desenvolvedores(as) uma iniciativa de questionar ou de não se adaptar a um modo de vida e de pensamento proposto durante o que Berger e Luckman (2005) consideram como socialização primária (predominantemente a infância).

Também se destaca a preponderância dos jovens na ocupação: de acordo com os microdados do Censo 2010, as medidas de tendência central da idade se concentram na faixa dos 20 aos 30 anos, com a média em 30 anos, a mediana em 28 e a moda em 25. É possível observar no *Gráfico 2* como a distribuição de

frequência é assimétrica a partir da curva normal sobreposta. Por um lado, podemos compreender isso como um resultado da importância da socialização antecipada desde a infância com a computação, em que é muito comum as narrativas de vida deses(as) profissionais convergirem seu início com um contato emocional com o primeiro computador na infância ou início da adolescência, perceptível tanto em entrevistas realizadas por mim quanto na “história de vida típica” construída por Gabriella Coleman (2013) a partir de sua coleção de entrevistas.

Gráfico 2 – Distribuição de faixas etárias entre desenvolvedores(as) de *software* (exceto analistas de sistemas)



Fonte: Microdados do Censo 2010. Elaboração própria.

Nota: foram desconsiderados neste gráfico, apenas para melhor visualização, os casos com mais de 60 anos, correspondendo a 0,73% do total.

Por outro lado, esse fenômeno se contrapõe à premissa de que muitas ocupações baseadas em um nível de ensino superior possuem um peso relativamente maior para faixas etárias de 30 e 40 anos (levando em conta o tempo em que é necessário investir até conseguir a *expertise* e as experiências profissionais anteriores). Temos então duas circunstâncias plausíveis por trás disso: uma possibilidade é que apenas nos últimos anos essa ocupação teria expandido postos no mercado de trabalho (pelo menos no contexto brasileiro), oferecendo então mais oportunidades recentemente, o que traria um grande número de novatos¹⁶. Empreender uma comparação histórica para verificar um deslocamento geracional seria difícil pois a família ocupacional “desenvolvedores e analistas de programas e aplicativos (software) e multimídia” que selecionei para as análises não

¹⁶ Pressupondo aqui que, pelo menos, a maioria das pessoas com mais de 40 anos tampouco não teria iniciado o trabalho em desenvolvimento de *software* recentemente.

havia sido incluída no Censo 2000, por exemplo. Já outra possibilidade, ainda que menos provável de acontecer tão massivamente, é a relação com a saída de pessoas da carreira, o que é apontado qualitativamente em algumas observações sobre a precariedade no trabalho em TI.

Um dos apontamentos da pesquisa de Maria Aparecida Bridi e Benilde Motim (2014) é que a contratação sem registro é mais frequente entre os(as) trabalhadores(as) mais jovens do setor da indústria de informática, e que apenas após “alguma experiência e idade, eles mesmos procuram garantir alguma segurança para planejar seu futuro e, assim, buscarem contratos formais e/ou melhores salários.” (BRIDI; MOTIM, 2014, p. 27)

Bárbara Castro (2014) também relata a tendência de que os/as mais jovens do setor de TI geralmente valorizem os contratos flexíveis, que exigem trabalho intenso e pouco descanso, por conta do mais alto rendimento; até o momento em que a saúde comprometa sua capacidade de continuar nesse ritmo, ou quando decidem ter filhos (caso especificamente mais dificultoso para as mulheres), quando esses(as) trabalhadores(as) começam a valorizar um emprego estável.

Nesse contexto, cabe apontar como se qualificam as diferenças no estado civil dos(as) desenvolvedores(as) de *software*. Ao observar as distribuições de frequência das variáveis relacionadas à nupcialidade (estado conjugal, natureza da união conjugal e estado civil) notei pequenas variações entre os resultados de meu *survey* – que aponta para uma ligeira maioria de solteiros(as) (48,6%) em relação aos casados(as) (45,2%) – e os dados do Censo 2010¹⁷ – onde a diferença inicialmente foi mais significativa, com 63,5% de solteiros(as) contra 32,1% de casados(as). Além disso, o *survey* indicou que 55,5% das pessoas residia com um(a) cônjuge ou companheiro(a) e que 36,9% não viviam e nunca tinham vivido com algum(a) cônjuge; ao passo que no Censo apenas 43,4% de pessoas vivam com um(a) cônjuge em contraponto com 49,5% de respondentes que nunca viveram com cônjuge. Essas flutuações de valores sugerem que há um número considerável de pessoas que vivem em companhia de cônjuge sem estarem casadas civil ou

17 Aqui partindo do banco de dados que desconsidera a ocupação de analistas de sistemas.

religiosamente, o que é atestado no Censo com a informação de que 32% das pessoas que vivem com cônjuge estão sob o regime de união consensual.

Em alternativa a essa explicação, resta uma impressão de que outros fatores possam influenciar essa variação, e por essa razão, retornei a uma análise comparativa entre as ocupações propostas pelo Censo¹⁸. A *Tabela 19* denota uma relação entre as diferentes ocupações e o comprometimento na relação conjugal: os integrantes dentre os “analistas de sistemas” e os “desenvolvedores e analistas não-classificados anteriormente”, grupos considerados mais estáveis no capítulo anterior apresentaram uma maioria de pessoas que vivem com cônjuge; enquanto isso, os “desenvolvedores de páginas de internet e multimídia” e “programadores de aplicações” apresentaram uma maioria de indivíduos que nunca viveram com cônjuge; e por fim, a ocupação de “desenvolvedores de programas e aplicativos” restou com os valores relativamente equilibrados.

Tabela 19 – Estado Conjugal e Ocupações

Ocupação	Vive em companhia de cônjuge ou companheiro(a)?		
	Sim	Não, mas viveu	Não, nunca viveu
Analistas de Sistemas	143.008	19.458	76.694
% da linha	59,80%	8,14%	32,07%
Desenvolvedores de Programas [...]	9.582	1.488	10.047
% da linha	45,38%	7,05%	47,58%
Desenvolvedores de Páginas [...]	1.986	303	2.467
% da linha	41,76%	6,37%	51,87%
Programadores de Aplicações	22.769	3.856	27.360
% da linha	42,18%	7,14%	50,68%
[...] Não Classificados Anteriormente	1.284	167	769
% da linha	57,85%	7,51%	34,63%
Total	178.629	25.272	117.338
% da linha	55,61%	7,87%	36,53%

Fonte: Microdados do Censo 2010 ponderados por peso amostral. Elaboração própria.

Nota: A diferença entre soma de parcelas e seus totais ocorre devido ao arredondamento.

Observemos agora a quais pessoas ou instituições os(as) respondentes atribuíram seus contatos iniciais com a computação. Nesta variável, o objetivo foi mapear a influência de alguns fatores sobre a socialização inicial dos(as) desenvolvedores(as) de software sobre a escolha dessa carreira. E à medida que

¹⁸ Incluindo as cinco ocupações da família COD 251, tal como no início do capítulo anterior.

algumas instâncias possuem uma maior frequência de influência atribuída elas se tornam uma fonte comum de características entre as pessoas de um campo. Por isso, alguns espaços sociais clássicos como a família e a escola se tornam fundamentais para um grande número de teorias e escolas sociológicas (cf. BERGER; LUCKMANN, 2005).

Em nosso caso estudado, a família foi responsável pelo primeiro contato com a computação para quase metade (44%) das pessoas que responderam o questionário, tal como exposto com mais detalhes na *Tabela 20* a seguir. Em companhia de outros fatores como amigos (38%) e escola (24%), é possível entender que para um grande parte dos(das) desenvolvedores(as) de *software* as instituições mais próximas da “socialização primária” têm relevância para a afinidade com a computação. E mais uma vez a relação mais elucidativa se dá com o tema da relativa juventude em que se dá não apenas o primeiro contato com a computação, mas um profundo interesse pelo objeto.

Tabela 20 – Atribuição do primeiro contato com computação

Quais pessoas ou instituições possibilitaram seus contatos iniciais com a computação? ¹⁹	Número de casos (N)	Percentual de respostas (%)	Percentual em relação ao número de respondentes (%)
Família	116	20,7%	44,1%
Amigos	100	17,9%	38,0%
Escola	62	11,1%	23,6%
Faculdade	73	13,0%	27,8%
Empresa	46	8,2%	17,5%
Igreja	1	0,2%	0,4%
Vizinhos	8	1,4%	3,0%
Você Mesmo(a)	143	25,5%	54,4%
Curso de Informática	4	0,7%	1,5%
Internet, Fóruns	4	0,7%	1,5%
Outras Formas	3	0,5%	1,1%
Total	560	100,0%	212,9%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

¹⁹ Esta variável esteve originalmente no questionário no formato de múltipla resposta, com cada valor sendo apresentado em ordem aleatória para cada respondente. Além disso, a opção “Você Mesmo(a)” foi acrescentada após o pré-teste; enquanto as opções “Curso de Informática”, “Internet, Fóruns” e “Outras Formas” foram incluídas após análise das respostas abertas em campo “Outras”.

A seguir, ainda com referência à *Tabela 20*, note que houve uma atribuição menos frequente para ambientes formais ligados ao período do início da carreira propriamente dita: a faculdade (28%) e empresa (17,5%). Elementos como igreja, vizinhos e até mesmo curso de informática chegaram a ter uma representação praticamente irrelevante. Em um primeiro momento pode parecer surpreendente que os cursos de informática não tenham desempenhado influência para a maioria dos(das) respondentes. Para compreender isso, basta reforçar um contraponto com a importância do ambiente familiar e do autoaprendizado na infância e adolescência como formas de iniciação na computação: os cursos de informática básica acabam se voltando para pessoas que não tiveram esse tipo de experiência inicial. Isso se evidencia na *Tabela 21*, em que os cursos técnicos ou profissionalizantes realizados ao longo da carreira recebem um peso relativamente semelhante entre as atribuições de pouca ou de muita importância.

Tabela 21 – Atribuição da importância para formação profissional

	Nada Importante		Pouco Importante		Razoavelmente Importante		Muito Importante		Extremamente Importante	
	N	%	N	%	N	%	N	%	N	%
Curso de Ensino Superior	12	4,6%	27	10,3%	41	15,7%	84	32,2%	97	37,2%
Curso Técnico ou Profissionalizante	44	20,4%	25	11,6%	62	28,7%	55	25,5%	30	13,9%
Cursos de Certificação	50	20,6%	41	16,9%	60	24,7%	51	21,0%	41	16,9%
Cursos oferecidos pela empresa	38	15,9%	44	18,4%	62	25,9%	68	28,5%	27	11,3%
Aprendizado por meio de fóruns, internet ...	0	0,0%	5	1,9%	24	9,3%	77	30,0%	151	58,8%
Aprendizado próprio, por tentativa e erro	0	0,0%	3	1,2%	24	9,3%	48	18,7%	182	70,8%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

É interessante que alguns elementos de formação profissional que em diversas outras ocupações (inclusive na área de TI) seriam consideradas fundamentais ou, pelo menos, bastante relevantes, assumem aqui uma consideração secundária. Entram nessa lista, incluindo o já mencionado curso técnico ou profissionalizante, os

cursos oferecidos pela empresa empregadora e os cursos de certificação, que habilitam a utilização em uma linguagem ou ferramenta de programação específica, por exemplo. Enquanto isso, com um peso razoável na atribuição de grande importância restou o curso de ensino superior, que, conforme discutido anteriormente possui uma valorização ambivalente entre desenvolvedores(as) de software livre: é considerado enriquecedor, porém nem sempre obrigatório, pois o campo é relativamente flexível quanto à formação exigida.

E por fim, o destaque repousa nas categorias “individualistas”, que apontam para a valorização do autodidatismo com a maior frequência entre os(as) desenvolvedores(as): “você mesmo” como uma fonte do contato inicial com a computação assumiu uma frequência de 54% entre os/as respondentes. Essa categoria não foi priorizada por mim durante o planejamento da questão pois o objetivo era estimar a influência de quais outras pessoas ou instituições sobre a escolha e interesse na computação, isto é, quais formas de socialização foram predominantes. Entretanto, já em um pré-teste vários casos sugeriam a importância de “mim mesmo” como resposta possível, e decidi não ignorar esse interesse.

Essa tendência “individualista” se manifesta com mais evidência na *Tabela 21*, em que também são predominantes as categorias “aprendizado por meio de fóruns especializados, internet e outros meios disponíveis” e “aprendizado próprio, por tentativa e erro”. Aqui contudo, não está representado um simples ímpeto pela ideia de *self-made man*²⁰, mas um reconhecimento de que a maior parte do aprendizado dos pormenores técnicos acontece a partir dos momentos a sós em vez de receber estímulos institucionais. Ainda assim, o tópico que indica a aprendizagem a partir de fóruns, documentações e outros recursos na Internet deixa claro que se trata de uma forma de socialização com aqueles mais experientes, mesmo que uma interação assíncrona não seja percebida, o que poderia ser mal-entendida precipitadamente como uma atividade não-social.

²⁰ Expressão geralmente mantida em inglês por remeter às ideias clássicas do liberalismo estadunidense, significando “homem que se fez por conta própria”.

Levando isso em conta, perguntei a respeito da relevância profissional das certificações, dos cursos técnicos e do autoaprendizado a um desenvolvedor de *software* livre:

Em outras áreas, principalmente de software proprietário, que são coisas um pouco mais fechadas, às vezes são softwares caros de você ter, que você não pode ter em casa pra aprender sozinho, por exemplo, eu acho que faz mais sentido, mas em software livre há muito material, muita documentação disponível livremente. Os softwares você pode rodar em casa e fazer os testes e aprender em casa; e as certificações que existem, a maioria eu acabo vendo como algo um pouco meio burocrático, assim que as pessoas fazem nem tanto pra aprender, mas para mostrarem que sabem como é alguma coisa, mostrar que tem aquele nível de conhecimento. Eu não tenho nenhuma certificação, nenhum curso muito formal que eu tenha feito e também é um pouco daquela característica do desenvolvedor do software livre que a maioria costuma ser um pouco autodidata, e procurar aprender sozinho, com o que acha disponível na Internet em vez de pagar pra fazer um curso formal. (Tux)

Essa declaração representa uma forma de *expertise* que não se fundamenta tanto sobre o reconhecimento formal da experiência e do conhecimento acumulado, mas se concentra na prova direta do saber fazer, frequentemente sob uma justificativa meritocrática.

4.2 Condições de Trabalho

Vamos agora tratar das variáveis que exploram as características do ambiente de trabalho e dos perfis laborais que são comuns entre desenvolvedores(as) de *software*. Tomando como referência comparativa dois dos três critérios do IBGE (2012, p. 33) para definir o trabalho principal em ordem de importância (o maior número de horas normalmente trabalhadas por semana; e, em seguida, o maior rendimento mensal habitual), construí a variável “trabalho principal” após observar o conjunto das informações empíricas coletadas. Com isso, 246 respondentes (96,1% do total) foram categorizados(as) como desenvolvedores(as) de *software* enquanto trabalho principal.

Quanto à quantidade de vínculos de trabalho, fica confirmada a tendência apontada na análise do Censo 2010, na qual predomina o perfil de trabalho concentrado em apenas um vínculo, reduzindo assim minha expectativa de encontrar uma carga maior da precarização do trabalho na forma de uma alta

frequência de múltiplos vínculos. A proporção encontrada em meu *survey* foi a seguinte: 166 respondentes (64,8%) com um vínculo; 77 com dois vínculos (30,1%); e 13 pessoas que afirmaram estar sem nenhum vínculo no momento em que responderam (5,1%).

Dentre essas pessoas que possuem mais de um vínculo, 69% (N=53) declararam que todos os seus vínculos de trabalho atuais se relacionam diretamente com desenvolvimento de software, configurando portanto um perfil em que possuem mais de um ambiente de trabalho mas na mesma ocupação principal. Aqueles²¹ que estão envolvidos profissionalmente com outra ocupação simultaneamente informaram também qual é o outro trabalho. Por ser um número reduzido em tal parcela (N=22), houve apenas uma ligeira vantagem (N=7) de desenvolvedores que trabalham em ocupações relacionadas à Tecnologia da Informação: manutenção de computadores; *sysadmin* (administrador de sistemas); administrador de redes; análise de infraestrutura; gerente de TI. Em seguida se destacou o trabalho na docência (N=6), e as demais ocupações se dividiram entre aquelas relacionadas ao ambiente empresarial (“empresário”; “recursos humanos”; “analista de negócio”; “consultoria”; “função administrativa”) e aquelas relacionadas ao comércio e serviços (“bar”; “comerciante”; “festas infantis”; “marketing multinível”).

Já os casos que acusaram possuir nenhum vínculo de trabalho, demonstraram, através de outras variáveis, que estão sim envolvidos com a atividade de desenvolvimento de *software* ou, pelo menos, estiveram em momento anterior. Aliando isso ao fato de ser um número relativamente pequeno (N=13), creio ser razoável supor que estão temporariamente desempregados, e que não se trata de pessoas que apenas escrevem código nos horários vagos e que possuiriam outra ocupação principal totalmente diferente da programação: esse perfil “amador” é razoavelmente comum na comunidade de *software* livre, o que inicialmente me deixou apreensivo quanto à possibilidade da amostra do questionário agregar programadores/desenvolvedores profissionais e amadores.

21 Neste subgrupo houve apenas participação de homens.

Visto que a amostra coletada pelo *survey* conseguiu selecionar trabalhadores especializados em desenvolvimento de *software*, cabe referir-se mais uma vez ao item “posição na ocupação”, confirmando os resultados discutidos anteriormente. Podemos observar na *Tabela 22* que a estrutura empregatícia do setor apresenta características semelhantes ao compararmos os dois bancos de dados. Entre aqueles(as) que declararam trabalhar por conta própria (40 pessoas no total), 18 (45.0%) possuem CNPJ, caracterizando um tipo de trabalhador(a) conhecido(a) no campo como “PJ”.

Tabela 22 – Posição na Ocupação em percentuais comparativos entre Censo 2010 (exceto analistas de sistemas) e Survey 2015

	Censo 2010	Survey 2015
Empregado com carteira de trabalho assinada	66,2%	56,2%
Empregado pelo regime jurídico dos funcionários públicos	2,6%	9,4%
Empregado sem carteira de trabalho assinada	9,7%	11,3%
Conta própria	19,6%	15,2%
Empregador	1,8%	5,9%
Não remunerado	0,1%	2,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015 e Microdados do Censo 2010.

Considerando os(as) desenvolvedores(as) empregados(as) com ou sem carteira de trabalho assinada, tomei por base a classificação elaborada pelo DIEESE (SEBRAE, 2013) de estabelecimentos comerciais segundo seu porte a fim de elaborar uma variável a respeito do tamanho da empresa em que essas pessoas trabalham. Na *Tabela 23* estão designadas as faixas de tamanho, que de acordo com os critérios mencionados acima – com exceção da categoria “apenas eu mesmo” – correspondem a uma noção aproximada da classificação de tamanho. Esse critério do número de empregados não é tão exato quanto aquele que utiliza a receita bruta anual, porém seria inadequado perguntar desse modo aos(às) respondentes, que em geral não possuem funções de gerência na empresa, esperando deles(as) um conhecimento preciso das finanças.

Tabela 23 – Número de funcionários da empresa em que trabalha

Quantidade de funcionário(a)s	Classificação	Frequência (N)	Percentual (%)
Apenas eu mesmo	-	29	11,9%
2 a 9 pessoas	Microempresa	36	14,8%
10 a 49 pessoas	Pequena empresa	51	20,9%
50 a 99 pessoas	Média empresa	24	9,8%
100 ou mais pessoas	Grande empresa	104	42,6%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

As diferenças de frequência entre as categorias de micro e pequenas empresas, e até entre as médias empresas, são praticamente ignoráveis. Com isso o realce do conjunto de respondentes se concentra entre as grandes empresas, apesar de que esta parcela ainda assim não conseguiu atingir uma maioria absoluta, ou seja, mais da metade do total.

Tabela 24 – Local de Trabalho

Onde é seu local de trabalho?	Frequência (N)	Percentual (%)
Apenas na empresa	123	48,4%
Apenas no próprio domicílio (Home Office ou Remoto)	30	11,8%
Parcialmente na empresa e parcialmente no próprio domicílio	81	31,9%
Na instituição do cliente	3	1,2%
Parcialmente na instituição do cliente e parcialmente no próprio domicílio	6	2,4%
Parcialmente na empresa e parcialmente na instituição do cliente	7	2,8%
Em local alugado e compartilhado para trabalho (Coworking)	2	0,8%
Universidade	2	0,8%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Quanto ao local de trabalho, conforme disposto na *Tabela 24*, houve uma maior distribuição na forma convencional de trabalho na própria empresa em que se está contratado. No entanto, não se pode ignorar a quantidade expressiva de opções mais diversificadas de trabalho que surgiram nos últimos anos por reflexo das tendências de flexibilização do trabalho. As categorias dessa variável buscaram, além de expandir as opções oferecidas no Censo 2010, possibilitar uma contextualização em relação a pesquisas internacionais (Ó RIAIN, 2010; TREMBLAY; GENIN, 2009). Por isso, houve uma combinação das opções

pressupostas a partir da literatura: trabalho no local da empresa; em casa (seja no caso de autônomos que montam um chamado *Home Office* [escritório doméstico] ou no trabalho remoto à empresa em que é contratado); na instituição do cliente, de onde surgem três combinações entre essas possibilidades; e como acréscimo, uma tentativa de observar uma tendência recente de alguns empreendedores de trabalhar em local alugado e compartilhado por outros, em vez de arcar com custos de um espaço próprio, o chamado *Coworking*; além disso, a categoria “Universidade” foi acrescentada após análise do questionário, não sendo prevista por essa pré-visualização teórica.

Iniciando a análise nas questões intrínsecas ao processo de desenvolvimento de *software*, busquei questionar de que maneira o trabalho de desenvolvedores(as) se configura em relação à especialização nas atividades delimitadas pelo modelo de desenvolvimento “cascata”, que propõe um sequenciamento linear de etapas inspirado nas indústrias de construção. Apesar deste modelo não ser aplicado de maneira consensual, o que será discutido na próxima seção, é bastante conhecido e citado em livros-texto da engenharia de *software*, e seus elementos-chave podem ser extrapolados para outros métodos de desenvolvimento, com ao menos suas categorias sendo geralmente bem conhecidas no campo. Estou me referindo às categorias dispostas na *Tabela 25*. O objetivo desta variável foi confirmar se há predominância de atividades mais técnicas como a programação e o teste do *software* em relação a atividades que exigem mais interação com outras pessoas como a documentação, o suporte e treinamento, além de proporcionar um posterior detalhamento das distinções entre as especialidades.

Tabela 25 – Atividade predominante de desenvolvimento

Em qual dessas atividades de desenvolvimento de software você trabalha predominantemente?	Frequência (N)	Percentual (%)
Implementação (ou programação)	114	43,3%
Meu trabalho perpassa todas ou quase todas as atividades	78	29,7%
Liderança de projeto	17	6,5%
Design do software (ou projeto)	14	5,3%
Análise de sistemas	13	4,9%
Análise de requisitos	5	1,9%
Teste	5	1,9%

Em qual dessas atividades de desenvolvimento de software você trabalha predominantemente?	Frequência (N)	Percentual (%)
Implantação (ou instalação)	5	1,9%
Manutenção	4	1,5%
Suporte e treinamento	2	0,8%
Depuração (ou correção de erros)	2	0,8%
Documentação	1	0,4%
Outra especialidade	2	0,8%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

De fato, a maioria dos(das) respondentes (43,3%) relatou se concentrar em atividades de programação (às vezes descrita como implementação do código, quando no contexto do modelo cascata), indicando uma maior identificação com a identidade de “desenvolvedor(a) de *software*” como é razoável esperar visto que este era o rótulo descrito no critério de inclusão do questionário.

Em segundo lugar, um grande número de desenvolvedores(as) (29,7%) afirmou que seu trabalho perpassa por todas as atividades, isto é, não se inclui em um processo de divisão do trabalho extremamente detalhada. Isso pode estar relacionado com a frequência de pessoas que trabalham em pequenas empresas; mas também indica que aqueles que trabalham para grandes empresas não seguem tão estritamente as divisões que os livros-texto sugerem, e que porventura são aplicadas com mais frequência em seu contexto original estadunidense. E por fim, cabe notar que as demais atividades com uma frequência relativamente visível são aquelas com maior valorização no campo: liderança (6,5%), análise de sistemas (4,9%) e *design* do projeto (5,3%).

Levando em conta esse contexto, vale a pena conferir com quais pessoas os(as) desenvolvedores interagem no ambiente de trabalho: 158 respondentes (65,8%) expressaram interagir diretamente com o cliente ou usuário final do software durante o processo de desenvolvimento. E quanto à relação hierárquica no ambiente de trabalho, apenas 18,2% (43 casos) relataram não possuir chefe nem gerente; enquanto isso, 72,9% dos(das) desenvolvedores(as) trabalham no mesmo espaço que o(a) chefe; e 8,9 % se comunicam apenas virtualmente com seus superiores.

Essa composição é o plano de fundo para uma questão ainda mais importante, a saber, o reconhecimento das procedências sociais de pressão (ou estresse em

relação à produtividade) no trabalho. A *Tabela 26* apresenta as principais categorias previstas pela literatura do trabalho na área de TI, tal como a relação com os clientes/usuários, com a ênfase de Ó Riain (2010) sobre trabalhadores contratados em uma empresa; e a mesma situação ponderada por Tremblay e Genin (2009), mas a partir de trabalhadores autônomos. Esse vínculo com os usuários do *software* enquanto produto final, é representado muitas vezes por consumidores que requisitaram um produto com demandas específicas, e que por isso, se manifestam também em várias etapas do desenvolvimento diretamente com os programadores. A importância disso, que assumiu no *survey* a segunda fonte de pressão mais frequente, sugere que os(as) desenvolvedores não são apenas trabalhadores análogos a produtores em uma indústria (como proposto em algumas metodologias de desenvolvimento), mas também possuem características de prestadores de serviços, devido à necessidade de interação, mesmo que por vezes indesejada.

Tabela 26 – Atribuição da pressão por resultados no ambiente de trabalho

De quem você considera sofrer pressão para entregar resultados com rapidez?	Número de Casos (N)	Percentual de respostas (%)	Percentual de respondentes (%)
Chefe ou gerente	134	33,8%	52,8%
Líder da equipe	62	15,6%	24,4%
Colegas	46	11,6%	18,1%
Consumidor/Usuário	118	29,7%	46,5%
Contribuidores de código externos ao projeto	11	2,8%	4,3%
Si próprio	11	2,8%	4,3%
Não há pressão de ninguém	12	3,0%	4,7%
Prazos do Projeto	3	0,8%	1,2%
Total	397	100,0%	156,3%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

A imposição do chefe ou gerente da empresa é a forma mais direta e tradicional de controle do tempo e das decisões de um grupo de trabalho, portanto não surpreende ter sido a mais frequente. O chamado “líder de equipe” é uma categoria existente apenas em alguns contextos, coexistindo com o chefe, que está hierarquicamente superior, mas se diferenciando por ser praticamente um membro de uma equipe de trabalho, designado para coordenar especificamente um projeto dessa equipe, e sendo portanto uma relação mais próxima, cotidiana, e a princípio

menos coercitiva. A maneira como um(a) desenvolvedor(a) interage com seus(suas) colegas de trabalho pode variar bastante, desde um ambiente de incentivo à produtividade quase competitiva até um ambiente de grande descontração e ajuda mútua, tal como relatado entre meus entrevistados. Sobre isso, Ó Riain cita um caso interessante sobre como o ambiente de trabalho entre colegas pode influenciar a recepção de um cliente, tomando um viés de gênero:

Tal como um gerente de projetos experiente notou,[:] “Homens se comportam muito melhor quando há mulheres em volta! Com certeza, em todos os casos! [...] Eles agem com mais maturidade, é mais fácil de lidar com diferentes situações. Não chega a ser o dobro da maturidade, é uma coisa complementar... pois eu nunca estive em uma equipe com uma vasta maioria de mulheres. Em uma equipe de back-office²² com apenas homens, você tem uma atmosfera grosseira de vestiário. E você não consegue simplesmente desligar isso quando o cliente aparece. É uma atmosfera mais cordial com mulheres lá, você não precisa desligá-la”. (Ó Riain, 2010, p. 345, tradução minha).

Já os “contribuidores de código externos ao projeto” se referem a uma espécie de agentes específicos ao desenvolvimento de *software* livre e de código aberto, em que o processo de desenvolvimento passa por pessoas que estão fora da empresa/fundação em que se concentra a iniciativa, e por isso, sofre influência de outros contribuidores.

Ademais, acrescentei algumas outras categorias vindas da sugestão de respondentes em um campo aberto: no caso, as opções não-previstas “si próprio”, “não há pressão de ninguém” e “prazos do projeto”. A categoria “si próprio” acentua o assunto do “individualismo” proposto anteriormente, já que a imposição por mais trabalho não é vista primordialmente como tendo uma origem social “externa”. Mas de um ponto de vista sociológico, há de se completar que o que está em jogo é uma complexa internalização dos valores previamente socializados, tal como sintetiza a dialética da identidade “para si” e “para outro” de Dubar (2005). Esse tipo de manifestação é ainda mais forte na declaração de alguns de que “não há pressão de ninguém”: essa última classe tem a peculiaridade de retratar um perfil de quem se habituou a trabalhar em um regime intenso e deseja isso por conta própria, uma característica comum entre os neófitos do campo.

²² Os serviços de suporte de infraestrutura e TI de uma empresa, separada da fachada pública do negócio.

5 Cultura do Trabalho no *Software* Livre

Antes de iniciar a análise principal da pesquisa devo aqui reservar um espaço para a discussão de algumas questões metodológicas relativas ao meu *survey*, que não foram detalhadas anteriormente, dizendo respeito principalmente à amostragem, a fim de enriquecer brevemente a caracterização das pessoas pesquisadas. Creio não ser possível afirmar com segurança que a amostra coletada pelo *survey* apresenta uma proporção de desenvolvedores(as) de *software* livre de código aberto (SLCA) e de desenvolvedores(as) de *software* não-livre (SNL) semelhante à proporção da população, pois a questão pode ter sido interpretada de forma ambígua pelos(as) respondentes, e não encontrei um parâmetro de comparação confiável para sobre a proporção entre os dois grupos. É possível sim observar que obtive uma amostra representativa de desenvolvedores(as) de *software* em geral, ao compararmos as características demográficas com aquelas apresentadas por esse segmento populacional no Censo 2010.

Considerando isso, a comparação entre os dois subgrupos – desenvolvedores(as) que só lidam com SLCA (seja profissionalmente ou por *hobby*) e os/as que só lidam com *software* não-livre – priorizou inicialmente a construção de uma variável em escala, reconhecendo os(as) desenvolvedores(as) que atuam em ambas possibilidades simultaneamente entre as duas categorias, e também priorizando o tipo de *software* que está desenvolvendo atualmente, mesmo que tenha trabalhado com o outro no passado. No entanto, a transformação dos valores coletados para essa escala construída não foi insignificante: 27 casos afirmaram não desenvolver *software* livre ou *software* de código aberto e em outra variável os mesmos declararam também não desenvolver *software* proprietário ou *software* com código fechado²³. Acredito que uma justificativa plausível para essa interpretação dos(as) respondentes(as) a respeito do que imaginei ser uma oposição evidente demonstra que essas identidades são frequentemente exógenas ao campo, conforme discuto na próxima seção. Ainda assim, essa estranha constatação exigiu decisões difíceis sobre como classificar esses casos ou até mesmo se deveriam ser

²³ Além disso, três casos que afirmaram ter desenvolvido SLCA anteriormente e não desenvolver *software* proprietário considerei como desenvolvedores de *software* livre; outros três casos que responderam ter desenvolvido *software* proprietário anteriormente e não desenvolver SLCA atualmente considerei como desenvolvedores de *software* proprietário.

desconsiderados. Não levei adiante a última opção pois o conjunto das demais respostas desses casos parecia demonstrar que se tratavam realmente de desenvolvedores. Com isso, um forte critério de ponderação foi o da licença do *software*: licença com *copyleft* forte (GPL), licença de código aberto permissiva (p. ex: Licença MIT ou X11, Licença BSD, Licença Apache, Licença Artística) ou outros tipos de licença aberta com *copyleft* parcial (p. ex. Licença Pública Mozilla, Licença PHP) significa que se desenvolve *software* livre ou de código aberto; já uma licença comercial ou proprietária (p. ex: contrato do tipo *End-User License Agreement*) implica o desenvolvimento de *software* não-livre. Essa operação redundou na variável exposta por meio da *Tabela 27*:

Tabela 27 – Envolvimento profissional de desenvolvedores(as) com *software* livre, *software* não-livre ou ambos

Desenvolve (ou está envolvido em projeto) ²⁴	Frequência (N)	Percentual (%)
Apenas Software Livre e Software de Código Aberto	60	23,7%
Tanto Software Livre quanto Software Não-Livre	85	33,6%
Apenas Software Não-Livre	108	42,7%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Tendo por base as informações empíricas organizadas por esta variável, recompus os dois subgrupos de desenvolvedores em duas variáveis: aqueles que trabalham com SLCA e aqueles que trabalham com SNL. Assim, aqueles que desenvolvem SLCA e SNL simultaneamente foram embutidos nas duas variáveis, isto é, não se trata de uma divisão populacional, mas conceitual, permitindo isolar mais claramente as características desses dois grupos assim reconhecidos. Essa recategorização das respostas coletadas originalmente é apresentada na *Tabela 28*:

Tabela 28 – Categorização de desenvolvedores(as) de *software* quanto ao envolvimento com *software* livre ou não-livre

Desenvolve (ou está envolvido em projeto)	Frequência (N)	Percentual (%)
Software Livre e Software de Código Aberto	138	52,5%
Software Não-Livre	188	71,5%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

²⁴ Tanto nesta tabela quanto daqui em diante a variável ignora os casos ausentes (N=10).

Por fim, devo acrescentar que um possível viés tenha se constituído por causa do grande número de questões do *survey*, que diversos participantes se queixaram numa seção de comentários. Isso talvez tenha acarretado o fato de metade das respostas ao questionário se constituírem como incompletas (seja aquelas que responderam pouquíssimas questões ou nenhuma, seja respostas até a metade ou quase até o final), o que me levou a excluir a maior parte desses casos (exceto um pequeno número que desistiu apenas de responder as duas últimas seções). Com isso, a suposição não verificável é de que pessoas que possuem relativamente menos tempo livre (tal como pais ou mães com crianças pequenas, pessoas que possuem mais de um vínculo, etc) estejam sub-representadas no *survey*.

5.1 Singularidades do *Software* Livre

Antes de mais nada devo enfatizar que não existe uma identidade coletiva de “desenvolvedor de *software* não-livre” ou “desenvolvedor de *software* proprietário”. Através dos discursos é utilizada genericamente a ideia de “desenvolvedor”, e mais especificamente a noção de “desenvolvedor de *software* livre”. A oposição só surge a partir da definição do que é *software* livre, e daí, uma necessidade de delimitar o que não é *software* livre. Conforme exposto no capítulo *Contrapontos Teóricos*, a noção de *software* vai sendo construída historicamente: inicialmente até mesmo a oposição entre *software* e *hardware* teve de ser conceitualizada pela IBM nos anos 1960 a fim de que programas escritos em um computador pudessem ser utilizados em outros da mesma família (EISCHEN, 2000). Mas ainda nos anos 1970 grande parte dos computadores podiam ser utilizados sem obstrução de senhas entre usuários, e com o código disponível, de forma que o que Richard Stallman mais tarde conceitualizou como *software* livre era o tipo de computação predominante.

Esse tipo de fenômeno de delimitação por oposição é um exemplo do argumento proposto por várias teorias das ciências sociais, em que é a diferença (a relação com um *outro* reconstruído) que permite a constituição da identidade de um grupo delimitado. Isto é, a identidade não deriva simplesmente de uma essência das características em comum de um grupo, mas sim da contínua relação intersubjetiva entre as pessoas.

Becker (2008) ilustra com clareza como o conceito de *outsider* – o tipo de pessoa mais propensa a romper com normas sociais – varia de acordo com posturas teóricas. Uma visão predominante até meados do século XX era de que os *outsiders* são pessoas que de fato realizam comportamentos desviantes. Sem embargo, Becker apresenta uma perspectiva inspirada no interacionismo, na qual o rótulo é mais importante do que o ato praticado em si. Assim, trata-se de considerar pessoas que são rotuladas como desviantes, mesmo que não cometam os atos descritos pelos outros; ao passo que também há pessoas que cometem tais atos sem chegarem a ser rotuladas. Uma perspectiva similar sobre este assunto é proposta por Dubar (2005), conforme exposto anteriormente: trata-se de uma teoria sintetizante da construção das identidades sociais a partir de uma dialética entre as definições *para si* e aquelas *para o outro*, ponderando entre as explicações interpretativistas e estruturalistas.

Essas observações deixam evidente o quão complexas são as construções identitárias. E dessa maneira, cabe ressaltar empiricamente como a oposição identitária entre desenvolvimento de *software* livre e *software* proprietário se efetiva a partir do discurso do movimento *software* livre. Ao rever a *Tabela 27*, notamos um número expressivo (33,6%) de pessoas envolvidas simultaneamente com os dois modelos de desenvolvimento: com código aberto e fechado. Isso aponta para uma relativização da oposição entre as duas categorias no mercado de trabalho, ou a um esforço de mesclar pragmaticamente o *software* de código fechado e aberto de acordo com as necessidades corporativas. Se entendermos que o movimento do *software* livre, tal como defendido pela FSF e pelo projeto GNU, valoriza a tendência ao *copyleft*²⁵ (na qual um *software* livre só pode ser derivado em outro *software* livre) então, é razoável supor que o modelo de código aberto defendido pela OSI (onde se espera mais permissividade para que *software* proprietário seja integrado com *software* livre ou de código aberto) conquistou maior predominância mercadológica (por enquanto, levando apenas em conta o número de trabalhadores especializados).

²⁵ De acordo com o projeto GNU, “O *copyleft* diz que qualquer um que distribui o software, com ou sem modificações, tem que passar adiante a liberdade de copiar e modificar novamente o programa” (FSF, 2015).

Considerando em um primeiro momento apenas os(as) respondentes que trabalham simultaneamente com SLCA e SNL, poderemos contextualizar algumas tensões do campo. Seguindo a tendência apontada anteriormente, a *Tabela 29* indica uma relativa fragilidade ocupacional do setor de SLCA, já que a maioria (68%) desses desenvolvedores “híbridos” não sustenta seu trabalho a partir do desenvolvimento de SLCA.

Tabela 29 – Peso do trabalho com *software* livre na composição da renda pessoal entre desenvolvedores que trabalham simultaneamente com SLCA e SNL

O quanto de sua renda pessoal é obtida através de seu trabalho com <i>software</i> livre?	Frequência (N)	Percentual (%)
Nenhuma ou bem pouca parte	51	68,0%
Menos da metade da minha renda	7	9,3%
Metade da minha renda	1	1,3%
Mais da metade da minha renda	3	4,0%
Toda ou quase toda minha renda	13	17,3%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

A seguir, apresento na *Tabela 30* uma escala comparativa do tempo de trabalho que desenvolvedores “híbridos” dedicam ao SLCA, onde é possível notar frequências análogas à da escala da *Tabela 29*. Essa consistência proporcional de tempo e rendimento indica uma segmentação no mercado de trabalho que confirma a especialização dos setores de SLCA e SNL como “trabalhos principais”²⁶ distintos, mesmo entre esses desenvolvedores que estão envolvidos com ambos paradigmas.

Tabela 30 – Tempo relativo de trabalho dedicado ao *software* livre entre desenvolvedores que trabalham simultaneamente com SLCA e SNL

Quanto do seu tempo de trabalho semanal você habitualmente dedica ao <i>software</i> livre (em relação ao tempo que dedica ao <i>software</i> proprietário)?	Frequência (N)	Percentual (%)
A maior parte do tempo me dedico ao <i>software</i> livre	12	15,8%
Dedico aproximadamente um tempo igual ao <i>software</i> livre e ao <i>software</i> proprietário	11	14,5%
A maior parte do tempo me dedico ao <i>software</i> proprietário	53	69,7%
Total	76	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

²⁶ As variáveis relativas à comparação do tempo e rendimento são inspiradas nos critérios do IBGE para o trabalho principal, tal como exposto anteriormente na seção *Condições de Trabalho*.

Dessa forma, a *Tabela 30* reafirma a preponderância do trabalho com SNL enquanto uma prioridade de tempo, o que significa muito provavelmente que essa parcela da amostra está envolvida com *software* livre apenas em suas horas livres e sem o retorno financeiro. Essas características são comumente representadas na figura do desenvolvedor de *software* livre voluntário, que contribui para projetos enquanto um *hobby*, um perfil que é incentivado por projetos, principalmente aqueles de menor dimensão. Cabe frisar também que existe uma fraca participação de casos nas categorias intermediárias da escala, isto é, não há uma flexibilização expressiva de pessoas que trabalham com SLCA e SNL no mesmo emprego, por exemplo, o que seria representado por uma divisão quase inseparável do rendimento obtido com SNL e SLCA, além de um tempo de trabalho semelhante. No caso da pequena parcela que se concentra no SLCA e atua com pouco tempo e rendimento no SNL, é difícil estabelecer uma interpretação além da hipótese de rotatividade de funções dentro de uma companhia, em que desenvolvedores trabalham com SLCA, mas podem eventualmente interagir com projetos de SNL.

Vamos então continuar a explorar as características de desenvolvedores de SLCA em contraposição aos demais para reanalisar algumas já mencionadas, e em seguida, ressaltar novas variáveis. Na *Tabela 31*, retomo a variável de posição na ocupação para demonstrar mais um indício de que o trabalho especializado em SNL é mais propenso a oferecer estabilidade empregatícia do que o SLCA.

Tabela 31 – Posição na ocupação segundo tipo de licença do *software* desenvolvido

Posição na Ocupação	Software Livre e Código Aberto		Software Não-Livre	
	N	% da coluna	N	% da coluna
Empregado(a) com carteira de trabalho assinada	58	43,3%	120	64,5%
Empregado(a) pelo regime jurídico de funcionário(a)s público(a)s	14	10,4%	10	5,4%
Empregado(a) sem carteira de trabalho assinada	19	14,2%	19	10,2%
Conta própria	28	20,9%	22	11,8%
Empregador(a)	11	8,2%	13	7,0%
Não remunerado(a)	4	3,0%	2	1,1%
Total	134	100,0%	186	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

A principal diferença nessa comparação está nos vínculos empregatícios com ou sem carteira de trabalho assinada. Desenvolvedores(as) envolvidos(as) com SNL estão concentrados(as) no vínculo de trabalho formal, com 64% da categoria, enquanto apenas 43% dos(as) desenvolvedores(as) de SLCA possuem o mesmo tipo de contratação. Isso também é indicado pela proporção relativa ligeiramente maior de pessoas empregadas sem carteira de trabalho entre SLCA ainda que a frequência absoluta seja a mesma.

A anteriormente referida fragilidade do mercado de trabalho especializado em desenvolvimento de SLCA no Brasil se expressa também pela maior participação relativa de indivíduos na condição de trabalho por conta própria. Isso porque vale levar em conta que essa categoria hoje entendida e estimada socialmente a partir do termo empreendedorismo está fortemente ligada aos valores do vigente neo-liberalismo, já que permite uma flexibilidade de contratação na forma de prestadores de serviços terceirizados, frequentemente enquanto *freelancers* PJ.

Esse tipo de flexibilidade das formas de contratação encontra prevalência no campo de desenvolvimento de *software* na forma de trabalhadores na condição de pessoa jurídica, “CLT Flex” ou cooperativas de trabalho (BRIDI; BRAUNERT, 2015). Tal como exposto previamente, esse tipo de vínculo pode ser valorizado pelos(as) próprios trabalhadores(as), principalmente no início da carreira, por remeter a uma ideia de liberdade de horários e de método de trabalho, ainda que sob cargas de trabalho por dia mais pesadas.

Outro aspecto interessante apresentado na *Tabela 31* é a ligeira sobreposição relativa de funcionários(as) públicos(as), devido à valorização do setor público pelo SL. Um dos motivos para isso redundou na Instrução Normativa nº 1 de 2011, que estabeleceu para o serviço público federal o objetivo de preferir a utilização de SL, embasado em discursos da autonomia técnica nacional e da inclusão digital. Outro motivo importante é o da redução de custos, já que a maioria dos SLCA pode ser usado sem custos: mesmo quando o instalador/executável não está disponível gratuitamente, existe a garantia legal de se poder compilar (a partir do código-fonte) sua própria versão, seja modificando-a ou não. Com isso, o setor público forma um pequeno setor do mercado de trabalho em que estão disponíveis vagas destinadas exclusivamente ao desenvolvimento de SLCA.

A partir das diferenças nas formas de contratação ditas acima, é possível compreender o maior percentual de pessoas que trabalham sem chefe ou gerente entre desenvolvedores de SLCA em comparação com aqueles que programam SNL, como visto na *Tabela 32*, supondo a relação com o maior percentual de pessoas trabalhando por conta própria, já que possuem proporções similares.

Tabela 32 – Tipo de relação com ou ausência de chefe ou gerente

Tem chefe?	SLCA		SNL	
	N	%	N	%
Não tenho chefe nem gerente	34	28,1%	24	13,9%
Nós nos comunicamos apenas virtualmente	12	9,9%	18	10,4%
Nós trabalhamos no mesmo espaço	75	62,0%	131	75,7%
Total	121	100,0%	173	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Ainda que seja plausível a partir de pressupostos teóricos imaginar que desenvolvedores de SLCA, devido à valorização genérica da ideia de “liberdade”, tenham maior rejeição para trabalhar sob métodos pouco flexíveis ditados por um chefe que não está diretamente presente no ambiente cotidiano, essa explicação não me parece suficiente. Podemos observar, por exemplo, que o ambiente de flexibilidade de comunicação exclusivamente virtual com a chefia não é expressivamente mais frequente entre desenvolvedores de SLCA. Não obstante, ao ignorar a pequena diferença entre os dois segmentos, a participação estatística do subgrupo total de pessoas que tem apenas um contato virtual com o chefe é considerável, quando se compara com essa possibilidade em outras ocupações.

Quanto à carga de trabalho semanal, posta em comparação na *Tabela 33*, já em uma primeira vista as duas distribuições estatísticas não se mostram díspares, pois o quartil superior, a mediana e a moda são idênticas, sem contar o fato de que as duas últimas medidas citadas são as 40 horas semanais regulamentadas. Ainda assim, a média já indica uma pequena diferença de 38 horas de trabalho de desenvolvedores de SLCA contra 41 de desenvolvedores de SNL, o que pode estar relacionado com o maior desvio padrão da distribuição entre o SLCA, ou seja, com uma ligeira variação mais abrangente de frequências neste segmento.

Tabela 33 – Horas por semana de trabalho de desenvolvedores segundo tipo de licença do *software* desenvolvido

Desenvolve	Em seu trabalho como desenvolvedor(a), quantas horas você trabalha habitualmente por semana?						
	Média	Moda	Quartil Inferior	Mediana	Quartil Superior	Desvio Padrão	Frequência Total (N)
SLCA	38	40	30	40	45	17	134
SNL	41	40	40	40	45	14	185

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Tendo em vista essa constância entre os dois segmentos de *software* no que diz respeito às horas trabalhadas com remuneração, é nítido o contraste em relação à comparação entre as horas dedicadas a atividades voluntárias ligadas à computação, exposto a seguir, na *Tabela 34*. Nesta variável, as duas categorias não apresentam uma distribuição de frequência normal, apesar de o desvio padrão possuir um valor absoluto menor do que na variável anterior. Com uma grande diferença entre desenvolvedores de SLCA e SNL, fica confirmada a expectativa de que o maior envolvimento com o *software* livre se relaciona com um maior número de horas dedicadas a projetos sem retorno financeiro imediato. Predomina entre desenvolvedores de SLCA a média, moda e mediana de 10 horas semanais com atividades que vão desde a contribuição de código em outros programas até a divulgação do *software* livre em eventos. Enquanto isso, desenvolvedores de SNL chegam a possuir o maior número de informantes com nenhuma hora semanal nesse tipo de envolvimento.

Tabela 34 – Horas por semana de atividades voluntárias de desenvolvedores segundo tipo de licença do *software* desenvolvido

Desenvolve	Quantas horas por semana em média você se dedica a trabalho não remunerado relacionado a outros projetos, grupos ou comunidades?						
	Média	Moda	Quartil Inferior	Mediana	Quartil Superior	Desvio Padrão	Frequência Total (N)
SLCA	10	10	4	10	10	8	121
SNL	7	0	0	5	10	8	165

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Continuando o exame das particularidades do desenvolvimento de SLCA, convém reparar como se segmentam as distinções conceituais entre o *software* livre,

tal como defendido pela FSF, e o código aberto, proposto pela OSI, uma discussão já efetuada anteriormente neste texto. A *Tabela 35* oferece esse panorama identitário a partir da comparação entre aqueles que desenvolvem apenas SLCA e os que também desenvolvem SNL (já que os desenvolvedores exclusivos de SNL não tiveram acesso a essa variável).

Tabela 35 – Identificação com conceito de *software* livre, código aberto ou não-identificação entre desenvolvedores de *software* livre e código aberto

Você se considera parte da comunidade de software livre ou da comunidade de código aberto?	Apenas SLCA		Tanto SLCA quanto SNL	
	N	% da coluna	N	% da coluna
Eu me vejo como parte da comunidade de software livre	22	52,4%	31	38,8%
Eu me vejo como parte da comunidade de código aberto	6	14,3%	19	23,8%
Utilizo software livre/aberto, mas não me importo com a distinção	14	33,3%	30	37,5%
Total	42	100,0%	80	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

O que vale ressaltar aqui é principalmente a variação entre as pessoas que se reconhecem dentro da comunidade de código aberto: há maior participação dentro dos desenvolvedores que também se envolvem com SNL, uma relação que podemos embasar na tendência do conceito de código aberto em buscar a flexibilidade mercadológica em relação à ideia mais politizada de *software* livre. Por essa razão, há também uma maior concentração de desenvolvedores identificados com o *software* livre entre aqueles que apenas desenvolvem SLCA. Ainda assim, não podemos ignorar o peso daqueles que ignoram essa distinção político/linguística, cerca de um terço em ambas as amostras.

5.2 Identidade *Hacker*

Partimos então para um dos ângulos da pesquisa, a relação entre o *software* livre e o conceito original de *hacker*, seja formulado como uma identidade, cultura ou ética *hacker*. Apesar destes últimos três conceitos possuírem fundamentações sociológicas muito distintas, eles confluem para um *ethos* rico de significados compartilhados entre os membros dessa subcultura. Um dos documentos fundadores dos princípios *hacker* foi um ensaio (THE MENTOR, 1986) escrito por

um *hacker* da segurança da informação logo após sua prisão, onde ele apresentou na forma de um manifesto os pontos comuns à sociabilidade *hacker*, legitimando uma atividade de *hacking* para além do simples desejo de explorar outras pessoas em benefício próprio. Esse documento foi bastante influente entre entusiastas do campo desde os anos 1980 na sustentação do que viria a ser conhecida como ética *hacker*. A ideia de um manifesto *hacker* foi levada adiante por Wark (2004), que se aprofunda nas implicações filosóficas envolvidas pela conceitualização histórica de *hacker* ao mesmo tempo em que faz uma chamada para a ação de uma nova geração de *hackers*.

Outra influência histórica do campo foi o livro *Hackers: Heroes of the computer revolution*²⁷ (LEVY, 1994). Nesta obra Steven Levy reconstrói as noções aceitas tacitamente entre *hackers* conhecidos, e sintetiza a ética *hacker* a partir dos seguintes termos:

1. O acesso a computadores (e a qualquer coisa que possa lhe ensinar algo sobre a maneira como o mundo funciona) deve ser ilimitado e total. Sempre respeitar o princípio da participação ativa!²⁸ [...]
2. Todas as informações devem ser livres. [...]
3. Desconfie de autoridades; promova a descentralização. [...]
4. *Hackers* devem ser julgados pelo seu *hacking*, não por critérios fictícios como diplomas, idade, raça ou posição. [...]
5. Você pode criar arte e beleza em um computador. [...]
6. Os computadores podem mudar sua vida para melhor. (LEVY, 1994, tradução minha)

Além dessa formulação mais abstrata, a ética *hacker* também foi adaptada para argumentações mais técnicas, implementadas a contextos de segurança empresarial, tal como exemplifica Tori (2008). E por fim, é oportuno incluir as considerações de Thomas (2002) para a composição teórica da ideia de cultura *hacker*. Este autor realizou também um apanhado histórico sobre o assunto, partindo

²⁷ O título da tradução brasileira inclui referências a personalidades da computação: “Os Heróis da Revolução: como Steve Jobs, Steve Wozniak, Bill Gates, Mark Zuckerberg e outros mudaram para sempre as nossas vidas”. Isso apesar de que o autor sequer considera Steve Jobs como um *hacker* e que o livro tenha sido escrito décadas antes do Facebook de Zuckerberg.

²⁸ Termo original: *Hands-on Imperative!*

desde aquelas primeiras formulações conceituais, passando pelas representações midiáticas até as implicações jurídicas e sociotécnicas no mundo do início do século XXI. Tomando por base essas noções indicativas de uma subcultura específica, minha pesquisa enfoca na capacidade de que esses elementos culturais sejam componentes da construção identitária a partir de padrões de socialização em comum, tal como me embasa na teoria de Dubar (2005).

Porém, antes de constatar a identidade nos termos sociológicos, tratarei da questão como uma autoidentificação com o conceito de *hacker* e com a cultura *hacker*. O motivo de separar esse reconhecimento ideológico em duas variáveis se justifica pelo meu interesse em examinar o grau de afinidade com que os diferentes tipos de desenvolvedores(as) podem se relacionar. O pressuposto foi que perceber-se como parte da cultura *hacker* pode ser uma maneira mais suave de assumir respeito ou pelo menos conhecimento daquele conceito histórico de *hacker*.

Isso se atesta na *Tabela 36*, em que, independentemente de se referir a desenvolvedores de SLCA ou SNL, existe maior percentual relativo de autoidentificação com a cultura *hacker* do que de a si mesmo enquanto *hacker*. Quanto às diferenças na comparação, houve também uma confirmação da expectativa de que desenvolvedores(as) de SNL teriam menos afinidade com a identidade e cultura *hacker*, com 28% da categoria se considerando como *hacker*. Ao passo que entre desenvolvedores(as) de SLCA temos quase a metade (42%) se identificando com esse conceito; e a maioria (62%) chega a se declarar possuidora de vínculo com cultura *hacker*. Assim, temos um indício da continuidade contemporânea da relação histórica entre o desenvolvimento de *software* livre e a cultura *hacker*.

Tabela 36 – Autoatribuição como *hacker* ou parte da cultura *hacker* entre desenvolvedores segundo tipo de licença do *software* desenvolvido

		SLCA		SNL	
Você se considera de alguma forma parte da cultura <i>hacker</i> ?	Sim	83	61,9%	89	47,8%
	Não	51	38,1%	97	52,2%
	Total	134	100,0%	186	100,0%
Você se considera um <i>hacker</i> ?	Sim	57	42,5%	53	28,5%
	Não	77	57,5%	133	71,5%
	Total	134	100,0%	186	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Para compreender com mais detalhes como essa representação se dinamiza entre desenvolvedores de *software*, vamos observar algumas declarações das pessoas que responderam o questionário sobre o que consideram ser *hacker*. A *Tabela 37* mostra as principais categorizações dessa noção.

Tabela 37 – Definições de *hacker* entre desenvolvedores de *software*

Qual a característica principal que define um <i>hacker</i> ?	Frequência (N)	Percentual (%)
Curiosidade	34	19,8%
Maestria em computação, expertise	19	11,0%
Segurança e invasão de sistemas	19	11,0%
Conhecimento ou estudo	18	10,5%
Autodidata, aprendizado por experiência	10	5,8%
Persistência, dedicação	10	5,8%
Criatividade, improvisação, inovação	7	4,1%
Raciocínio lógico, solução de desafios	7	4,1%
Modificações, melhorias	6	3,5%
Pejorativo	5	2,9%
Não se interessa ou não sabe	4	2,3%
Software livre ou compartilhar conhecimento	4	2,3%
Subversão	4	2,3%
Criminoso, desonesto	3	1,7%
Definição autorrecursiva	3	1,7%
Distinção com cracker, atitude ética	3	1,7%
Engajamento social	3	1,7%
Outras definições	13	7,6%
Total	172	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

É preciso notar que originalmente esta era uma variável aberta para preenchimento no questionário, o que ocasiona duas consequências práticas: primeiro, a codificação desses termos feita em um momento posterior torna a questão mais dependente de minha interpretação/compreensão do que outras variáveis²⁹; e em segundo lugar, o número de casos ausentes foi bem maior do que as variáveis expostas tanto imediatamente antes quanto depois desta, totalizando 91 casos no total ou 31,6% da amostra. Isso significa que as pessoas deliberadamente escolheram não responder esse tipo de pergunta não-objetiva. Outro ponto a se acrescentar é que codificação desses trechos abertos frequentemente acarreta mais de um “código” ou “significado” na fala de uma mesma pessoa. Porém, para se obter uma análise quantitativa mais eficaz escolhi atribuir apenas um código por pessoa, o que poderia acarretar algum nível de reducionismo.

Posto isso, comento sobre essas descrições. A categoria mais citada como pressuposto da atividade *hacker* foi a “curiosidade”, na maior parte das vezes apenas mencionando essa palavra, mas em um dos casos há este trecho particularmente representativo da noção: “Alguém que fuça no que usa e tem curiosidade de entender como as coisas funcionam e como modificá-las”. Outra categoria que apareceu com muita frequência foi a de “conhecimento ou estudo”, que também foi referida sem maiores explicações. A noção de “autodidata” possui uma nuance extra em relação ao mero acúmulo de horas estudadas, pois se conecta com a experiência de aprender por “tentativa e erro”, de buscar “experimentações” e “fazer coisas com o conhecimento aprendido”. E também está próxima semanticamente à categoria de “persistência e dedicação”, por pressupor o estudo e prática constante da programação nos trechos: “Resiliência predominante, não ir dormir enquanto não terminar aquele assunto novo ou resolver aquele 'bug'” e “ignorar o que te bloqueia”.

A “criatividade” contém, além de menções apenas à própria palavra: “Utilizar um sistema para outros fins que não os que foram inicialmente definidos”; “Se aproveitar de brechas para utilizar algo de uma forma que não tinha sido planejado para fazer”; “Habilidade e criatividade de encontrar caminhos ainda não explorados”.

²⁹ O que pode ser compensado pela verificação pública do banco de dados da pesquisa, tal como o código fonte de um *software* livre ou de código aberto.

E assim há uma diferença quase sutil com a noção de “melhorar ou modificar” um sistema: “Modificar e manipular não trivialmente ou não autorizadamente sistemas de computação”; “ler > pensar > discutir > fazer > aprender > reinventar/melhorar”; “engajamento em descobrir falhas em sistemas, e assim reportá-las”; “Interesse e capacidade de fazer intervenções em coisas, sejam elas reais ou abstratas”.

A partir desse ponto, podemos constatar a aproximação com as propriedades do *software* livre, que chega a ser mencionado diretamente algumas vezes, enquanto que em outras respostas há uma alusão a algumas de suas qualidades intrínsecas: “Descentralização”, “liberdade de informação”, “compartilhar conhecimento” e “vontade de compartilhar”. As definições autorrecursivas, um artifício humorístico comum no *software* livre em referência ao projeto GNU, está representado por: “Ser um hacker... Porque o hacker tem característica própria”, “hackear” e “Se dar a liberdade de não responder essa pergunta”.

O entendimento de “engajamento social” perpassa por: “A capacidade de utilizar recursos tecnológicos para oferecer melhorias para o ambiente ou a comunidade a minha volta”, “senso de cidadania e de justiça e ética”, “sempre tentar ajudar as pessoas”. Já a ideia de “subversão” abarcou: “inquietação com a zona de conforto”, espírito livre”, “perfil questionador”, “capacidade de modificar o status quo”, “destemido”. Essas últimas duas categorias derivam das noções originárias de ser capaz de alterar o sistema segundo sua necessidade/curiosidade (o *Hands-on Imperative*), porém, elas também seguem por um caminho semântico que abandona a necessidade de referência à computação.

Em oposição a essa forma de pensar, foi bastante citada a concepção de “maestria em computação, expertise”, que se configura de maneira discrepante às noções de cultura *hacker*, se concentrando, então, em reservar o título de *hacker* para um especialista da área de computação, simplesmente alguém que possui muitos conhecimentos técnicos. Por esse caminho, a noção de *hacker* se exprime na categoria ainda mais específica de “segurança e invasão de sistemas”, que também foi comum: “pessoas que conhecem muito de segurança a ponto de poder explorar vulnerabilidades do sistema”; “alguém com conhecimentos técnicos e engajamento em descobrir falhas em sistemas, e assim reportá-las”.

Uma categoria que não consta na *Tabela 37* foi de “paixão ou diversão”, pois nunca se manifestou de maneira exclusiva em uma resposta, de modo que outro atributo se sobrepôs. Essa categoria conteve as seguintes descrições: “Gostar de aprender fazendo”; “Desmontar e montar coisas apenas por diversão” e “*Happy coding*”. Esse tipo de qualidade é mencionada em algumas definições teóricas de *hacking*, como em Himanen (2001).

Em outro extremo, as definições pejorativas incluíram: “muito tempo vago e falta de mulher”, “vagabundos”, “idiotice” e “prepotência”. Além desses retornos mais taxativos, houve uma resposta curiosa, que reflete sobre o caráter mais representacional do conceito: “Vejo a palavra ‘hacker’ como as palavras ‘geek’ e ‘nerd’. São termos vagos para pessoas que se interessam por tecnologia ou ‘cultura pop’. Atualmente essas denominações estão na moda”. Há ainda outra resposta interessante, porém não agrupada nas categorias principais: “Espero que, um dia, o mundo deixe de usar o termo hacker. Qualquer indivíduo que saiba programar um computador pode ser considerado hacker, segundo as definições que vemos por aí.”

Tabela 38 – Definições de *hacker* segundo autoatribuição como *hacker*

Qual a característica principal que define um hacker?	Você se considera um hacker?	
	Percentual Relativo à Coluna (%)	
	Não	Sim
Curiosidade	12,3%	31,8%
Maestria em computação, expertise	13,2%	7,6%
Segurança e invasão de sistemas	17,0%	1,5%
Conhecimento ou estudo	18,0%	10,5%
Autodidata, aprendizado por experiência	4,7%	7,6%
Persistência, dedicação	4,7%	7,6%
Criatividade, improvisação, inovação	1,9%	7,6%
Raciocínio lógico, solução de desafios	4,7%	3,0%
Modificações, melhorias	3,8%	3,0%
Subversão	3,8%	0,0%
Distinção com cracker, atitude ética	1,9%	1,5%
Engajamento social	0,0%	4,5%
Soma Total da Amostra Considerada	86%	86,2%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Nota: Algumas das categorias nas linhas foram retiradas apenas para facilitar a leitura do gráfico. Os valores percentuais ao grupo não foram alterados.

Agora indo além dessas descrições, é propício relacioná-las com a autoatribuição enquanto *hacker*, a fim de delimitar quais se baseiam numa suposta experiência própria e quais são representações das quais o sujeito não se identifica, o que apresento na *Tabela 38*. Para facilitar a análise realizei algumas transformações nesta tabela: a classificação “não se interessa ou não sabe”, a atribuição de “criminoso ou desonesto” e as classificações pejorativas se concentraram apenas entre pessoas que não se identificam como *hackers* e que tampouco se sentem próximas da cultura *hacker*, tal como esperado. A categoria “outras definições” não apresenta variação, já que também evidentemente é composta por um agrupamento que não tem nenhum significado em comum. Também foram desconsideradas as categorias (“definição autorrecursiva” e “*software* livre ou compartilhar conhecimento”) que possuem poucos casos como resposta, não permitindo assim uma comparação com suficiente relevância estatística.

À vista disso, é possível perceber algumas diferenças nessa comparação. As categorias que estabelecem o *hacker* como um especialista, “Maestria em computação, expertise” e “Segurança e invasão de sistemas” são bem mais comuns entre pessoas que não consideram a si mesmas como *hackers*, o que parece significar que há nesse grupo uma sensação de desidentificação com esse nível avançado de especialidade. Algumas características não apresentaram variância relevante entre os dois segmentos: “conhecimento ou estudo”, “modificações, melhorias” e “raciocínio lógico, solução de desafios”.

Surpreendentemente, a categoria “subversão” não denotou nenhuma presença entre aqueles que se consideraram *hackers*, sendo talvez uma maneira mais idealizada ou genérica de expressar a sua peculiaridade. Por outro lado, a categoria de “engajamento social” só possuiu manifestação entre os desenvolvedores que se reconhecem como *hackers*. Outros atributos relacionados à maneira como uma pessoa pode se tornar um *hacker* também foram relativamente mais comuns entre os identificados como *hackers*: “autodidata, aprendizado por experiência”, “criatividade, improvisação, inovação”, “curiosidade” e “persistência, dedicação”.

Tendo em vista essas questões, chegamos enfim na relação entre a autoatribuição identitária, seja como *hacker* seja como integrante da cultura *hacker*,

e o grau de envolvimento com o ativismo no *software* livre. O interesse em analisar esse tópico é compreender até que ponto os ideais *hacker* – que perpassam tanto a construção histórica da fundamentação do *software* livre quanto as motivações entre desenvolvedores na atualidade – podem assumir uma influência na maneira em que se efetiva um tipo de engajamento político das tecnologias. Na *Tabela 39*, é possível constatar de maneira geral que há uma associação crescente em se envolver no ativismo de *software* livre e se considerar como *hacker*.

Tabela 39 – Interesse pelo ativismo de *software* livre segundo autoatribuição como *hacker* ou parte da cultura *hacker*

Qual o seu nível de ativismo dentro do movimento software livre?	Você se considera parte da cultura hacker? (Sim)		Você se considera um hacker? (Sim)	
	N	% da coluna	N	% da coluna
Não me interessa pelo software livre	1	0,8%	2	2,7%
Tenho contato eventual com software livre, mas não sou ativista	23	19,3%	13	17,8%
Apenas divulgo o software livre em meus círculos sociais, sou mais uma(a) usuário do que um(a) ativista	37	31,1%	17	23,3%
Fui um ativista no passado, mas atualmente não sou mais	15	12,6%	7	0,0%
Participo de listas de discussão, promovo palestras e ajudo a organizar eventos	19	16,0%	12	16,4%
Sou um(a) ativista engajado(a), pois sou membro de um projeto, além de atividades como organizar eventos, escrever sobre software livre em sites, blogs, etc.	24	20,2%	22	30,1%
Total	119	100,0%	73	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Por fim, é necessário acrescentar que algumas relações entre variáveis, especialmente no caso da comparação entre desenvolvedores de SLCA e SNL, não apresentaram relevância estatística satisfatória, apesar da expectativa em descobrir alguma correlação. Esse tipo de relação verificou-se, por exemplo, entre as variáveis de sexo e raça em comparação quanto ao desenvolvimento de SLCA ou SNL. Quanto à diferença na proporção de gênero, não há praticamente nenhuma alteração no campo do SLCA; e no que diz respeito à variável cor/raça, apenas há uma ligeira variação na relação entre “pardos” e “pretos”, o que infelizmente aponta para a rigidez das formas de desigualdade social.

5.3 Motivações e Cultura do Trabalho

Nesta seção final, vou me deter sobre dois temas fundamentais para esta pesquisa (levando em considerando a comparação entre os perfis de desenvolvedores de SLCA e de SNL): a motivação para trabalhar na computação e alguns elementos da cultura do trabalho extraídos a partir da literatura. Esses dois temas compuseram dois grandes grupos de variáveis em que os/as respondentes avaliaram com base em uma escala de tipo Likert (com três categorias de resposta).³⁰ Assim, inicio a exposição desses resultados com a descrição de uma escala de satisfação em relação ao trabalho como desenvolvedor(a), na *Tabela 40*.

Tabela 40 – Nível de satisfação com o trabalho atual

Como você se considera em relação a seu vínculo de trabalho atual?		
	N	%
Eu apenas sou bem pago, faço este trabalho principalmente pelo dinheiro	17	7,3%
Não gosto muito do trabalho atual, se houver uma oportunidade, trocarei	27	11,6%
É interessante, mas posso imaginar ter outros vínculos de trabalho bons também	112	48,3%
É o que sempre quis fazer, eu realmente amo meu trabalho atual	76	32,8%
Total	232	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Essa tabela representa as características gerais de desenvolvedores(as) de *software*, pois em uma análise prévia a diferença de níveis de satisfação entre programadores de SLCA e de SNL não foi significativa. Assim, é possível constatar uma predominância da boa estima do trabalho desempenhado atualmente, com cerca de um terço dos/das respondentes reconhecendo um sentimento de forte identificação. A baixa vinculação desses trabalhadores à ocupação por fins instrumentais (“faço este trabalho principalmente pelo dinheiro”) será um fator de interpretação relevante em algumas das categorias referidas a seguir.

³⁰ Incluí duas variáveis de “controle” apenas com o propósito de ter uma ideia sobre a atenção ou dedicação em responder precisamente as perguntas apresentadas no final do questionário, levando em conta a extensão do questionário proposto (foram apresentadas até 87 perguntas que redundaram em 195 variáveis). Dentre as respostas válidas, 93,7% dos casos (N=209) na primeira variável, e 98,1% (N=212) na segunda marcaram corretamente a opção indicada. Esse teste indicou um ótimo nível de atenção e de compreensão do que está sendo proposto, já que poucos indicaram as opções fora do que a questão enunciava.

5.3.1 Motivações para o Desenvolvimento de Software

Dessa forma, a *Tabela 41* exhibe as proporções de concordância com afirmações relativas a tipos de motivação para o trabalho de desenvolvimento de *software*. Aqui temos um panorama de quais temas são mais ou menos comuns no campo: os predominantes são “aprender novas habilidades”, “realização de sonho”, “compartilhar conhecimentos” e “diversão em programar”. Isso demonstra uma influência indireta daqueles valores *hacker* apresentados na seção anterior, onde a paixão por programar, aprender e compartilhar extrapolam o âmbito dos desenvolvedores de *software* livre apenas.

Tabela 41 – Motivações para o trabalho de desenvolvedores de *software*

Avalie as seguintes motivações para realizar seu trabalho no desenvolvimento de software, de acordo com a importância em sua vida:	Porcentagem (%) segundo a linha		
	Não concordo	Concordo Parcialmente	Concordo Totalmente
Realização de um sonho: sempre desejei trabalhar na área de tecnologia	4,4%	34,1%	61,6%
Ascensão econômica e profissional	13,9%	51,3%	34,8%
Contribuir com o desenvolvimento da sociedade	10,7%	41,1%	48,2%
Aprender e desenvolver novas habilidades	2,6%	21,7%	78,0%
Ganhar reputação entre outros programadores	35,4%	39,0%	25,6%
Diminuir o poder de grandes companhias de software	72,6%	16,9%	10,5%
Resolver um problema que não poderia ser resolvido por outro software existente	11,3%	39,6%	49,1%
Compartilhar meus conhecimentos e habilidades	4,9%	31,1%	64,0%
Aumentar minhas oportunidades de emprego	13,2%	46,7%	40,1%
Vender produtos ou serviços	46,8%	31,9%	21,3%
Orgulho em fazer parte de uma comunidade	24,2%	42,2%	33,6%
Obrigação por uma circunstância profissional	72,7%	20,8%	6,5%
Diversão em programar	5,3%	21,7%	73,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Já os motivos mais rejeitados são a “obrigação por circunstância profissional” (que se justifica pelo alto número de pessoas que se identificam satisfatoriamente com a ocupação) e “diminuir o poder de grandes companhias de *software*”, uma opção mais ideologizada. Algumas dessas categorias foram inspiradas na tipificação proposta pela pesquisa de Augusto (2003), para que fosse possível comparar os resultados obtidos naquela década. Ao contrário do realizado em minha pesquisa,

ele incluiu também colaboradores voluntários (não-remunerados), a fim de isolar perfis de motivação, socialização e interesse no *software* livre. Nessa ocasião, a tipificação redundou em três perfis mais comuns: a de “hobista” (um perfil mais ideológico do *software* livre e próximo da cultura *hacker*); a de “profissional” (relacionado com a *expertise* específica, e mais afinado com as noções de *código* aberto como uma expressão meramente técnica); e um grupo intermediário de “estudiosos” (que buscam aprofundar conhecimentos sem distinguir entre esses vieses). O pesquisador também identificou um pequeno grupo de “obrigados”, pessoas que utilizavam *software* livre por coerção do trabalho. Uma hipótese que pode explicar algumas das diferenças encontradas em minha pesquisa seria o meu foco de estudar apenas os desenvolvedores profissionais, o que excluiria da amostra os “hobistas” e “estudiosos”, tornando minha amostra mais próxima da distinção mais embaçada entre o uso político e o técnico do SLCA.

Antes de apresentar quais das categorias da *Tabela 41* possuem relevância na comparação entre as duas amostras de interesse da pesquisa (SLCA e SNL), complementarei a discussão sobre a relação entre as diversas motivações, com o objetivo de compreender a construção social do trabalho desde o ponto de vista subjetivo. Para isso, proponho a seguir uma análise de caráter exploratório a respeito das relações entre essas motivações entre si, a fim de construir uma tipologia ideal e apontar caminhos a serem explorados em seguida.

De natureza comparável à análise fatorial, foi utilizada aqui a *análise de componentes principais* (FIELD; MILES; FIELD, 2012), uma técnica multivariada de análise estatística, cujo propósito principal é a redução de informações empíricas e sua sumarização. Ela analisa as relações entre múltiplas variáveis e tenta explicá-las em termos de suas dimensões subjacentes comuns (*componentes*), onde são apresentadas as relações mais relevantes do ponto de vista estatístico no contexto de interdependência entre todas as variáveis escolhidas. Dessa forma, por ser uma análise meramente estatística, a análise de componentes principais pode ser considerada uma forma de validar nexos explicativos, já que o ato de computar esses dados não inclui nenhuma compreensão dos significados das variáveis. Estabelecer as relações semânticas, isto é, compreender o significado, ou o que explica aquelas relações, fica a cargo do pesquisador. E nesse sentido, levei em

consideração para a interpretação o conjunto dos resultados descritivos apresentados anteriormente.

O método de extração escolhido analisou a matriz de covariância (em vez da matriz de correlação) já que as variáveis selecionadas possuem uma escala em comum. Além disso, escolhi um método de rotação oblíquo (Oblimin direto com delta 0), por haver uma pressuposição teórica de que algumas das variáveis devem se correlacionar, já que compartilham significados próximos (isto é, não pressuponho que a variância das categorias sejam independentes entre si). A vantagem da rotação é aumentar o escopo das cargas para além do primeiro fator, já que houve na matriz original um número excessivo de variáveis “relevantes” no primeiro componente principal (ou “fator”). Essa estratégia permite ainda, em nosso caso específico, auxiliar na formação de tipologias analíticas ideais: o que por sua vez pode ser entendido como uma operacionalização aproximada, ou inspirada rudemente, na noção de tipos ideais proposta por Weber (2006).

A princípio, em uma análise preliminar utilizando o método comentado acima, não houve força de associação suficiente entre as variáveis relacionadas à motivação e aquelas relacionadas à cultura do trabalho. Isto é, a análise dos componentes principais distinguiu esses dois temas em componentes totalmente isolados entre si, (apesar de possuírem uma escala similar) e por isso, decidi apresentar análises separadas sobre os dois temas. Porém, mais do que um simples entrave metodológico, isso já aponta um resultado frustrante para a hipótese central: não há correlação significativa entre uma motivação (estabelecida por padrões de socialização “anterior”) e formas específicas de cultura do trabalho. Desde já isso complica a (re)interpretação dos fenômenos tratados até aqui.

Além disso, para possibilitar a comparação desejada pela hipótese, separei as amostras de desenvolvedores de SLCA e de SNL, como se viessem de fontes diferentes. As análises, então, atenderam às duas medidas de adequação amostral: a de Kaiser-Meyer-Olkin (que, por convenção, aceita como satisfatórios valores acima de 0,5) revelou um índice de 0,706 para a amostra de desenvolvedores de SLCA e de 0,791 para a amostra de desenvolvedores de SNL; e o teste de Bartlett, indicou a rejeição da hipótese nula (p menor que 0,01), validando a adequação do

modelo. A *Tabela 42* expõe apenas a variância da amostra explicada pela análise empreendida, enquanto a *Tabela 43* contem os componentes principais.

Tabela 42 – Variância total explicada da análise das variáveis de motivações

Componentes Reescalados	Somadas de extração de carregamentos ao quadrado					
	SLCA			SNL		
	Total	% da variância	% cumulativa	Total	% da variância	% cumulativa
1	3,270	25,152	25,152	3,530	27,151	27,151
2	1,566	12,045	37,197	1,566	12,048	39,199
3	,977	7,517	44,714	,952	7,324	46,523
4	,986	7,587	52,301	1,321	10,162	56,685

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Tabela 43 – Matriz padrão dos escores de componentes da análise de motivações segundo tipo de desenvolvimento de software

Variáveis relacionadas à motivação pelo trabalho como desenvolvedor(a)	Componentes Reescalados							
	SLCA				SNL			
	1	2	3	4	1	2	3	4
Sempre desejei trabalhar na área de tecnologia [vocação]	,229	,103	,163	-,251	,270	-,369	,113	,246
Ascensão econômica e profissional [ascensão]	,244	-,335	,271	-,522	-,121	-,740	-,174	,072
Contribuir com o desenvolvimento da sociedade [sociedade]	,628	,231	-,116	,091	,706	,111	,127	-,080
Aprender e desenvolver novas [habilidades]	,330	-,078	,074	-,163	,473	-,252	,006	,383
Ganhar reputação entre outros programadores [reputação]	-,226	,204	,977	-,134	,075	-,778	,121	-,148
Diminuir o poder de grandes companhias de software [descentralizar]	,144	,816	,141	,238	,347	-,035	,103	-,539
Resolver um problema que não poderia ser resolvido por outro software [solução]	,159	,605	-,126	-,422	,737	,113	-,290	-,030
Compartilhar meus conhecimentos e habilidades [compartilhar]	,529	,224	,146	-,016	,731	-,031	-,122	-,050
Aumentar minhas oportunidades de emprego [oportunidade]	,220	,001	,142	-,704	-,075	-,688	-,127	-,193
Vender produtos ou serviços [venda]	-,126	,076	-,096	-,868	,158	-,144	-,918	,043
Orgulho em fazer parte de uma comunidade [orgulho]	,915	-,033	-,090	-,052	,574	-,148	-,048	-,469
Obrigação por uma circunstância profissional [obrigação]	,009	,311	,032	-,050	-,085	-,169	-,048	-,740
Diversão em programar [diversão]	,211	-,087	,311	,172	,641	-,169	,117	,183

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Nota: Para facilitar a leitura da tabela foram destacadas as cargas com valor acima de 0,5. A matriz SLCA teve a rotação convergida em 17 iterações, e a matriz SNL em 14 iterações, ambas pelo método Oblimin com normalização de Kaiser.

Tendo por base as indicações dos escores relevantes (aqueles com valor acima de 0,5) na *Tabela 43*, elaborei uma proposta de rotulação para esses componentes principais no *Quadro 1* abaixo. Os sinais negativos neste caso puderam ser ignorados, já que os únicos escores relevantes que possuíam valor negativo estão em um componente (cada coluna da tabela) em que não existem outros escores relevantes com valor positivo, em cuja correlação se contrastariam. Isto é, os valores negativos podem ser considerados positivos desde que todos os sinais do componente (ou fator) sejam invertidos (DUNTEMAN, 1989, p. 38). Esses escores apresentaram valor negativo provavelmente porque se contrastam à força do primeiro componente, indicando mais uma escala com os extremos *altruísta* e *empresarial*.

Quadro 1 – Síntese dos rótulos tipificados para a análise de componentes de motivações entre desenvolvedores de software

	SLCA		SNL	
	Rótulo	Variáveis Relevantes	Rótulo	Variáveis Relevantes
1	<i>Altruísta comunitário</i>	[orgulho]; [sociedade]; [compartilhar]	<i>Altruísta comunitário</i>	[solução]; [compartilhar]; [sociedade]; [diversão]; [orgulho]
2	<i>Profissional</i>	[descentralizar]; [solução]	<i>Busca por reconhecimento</i>	[reputação]; [ascensão]; [oportunidade]
3	<i>Busca por reconhecimento</i>	[reputação]	<i>Empresarial</i>	[venda]
4	<i>Empresarial</i>	[venda]; [oportunidade]; [ascensão]	<i>Profissional</i>	[obrigação]; [descentralizar]

Nota: As variáveis estão dispostas segundo a ordem da maior carga para a menor.

Minha interpretação é que não houve diferenças significantes entre as duas amostras, já que as características gerais rotuladas para cada componente principal apenas mudaram a ordem em que apareceram, impactando mais na frequência que cada uma explica da amostra do que numa mudança da estrutura. Isso indica uma rejeição da hipótese de distinções estruturais entre as motivações ocupacionais de desenvolvedores(as) de *software* livre e código aberto e de desenvolvedores(as) de *software* não-livre. A título de exemplo, o tipo *altruísta comunitário* não deveria

aparecer como primeiro componente na amostra de SNL numa interpretação forte da hipótese.

Além disso, essa síntese aponta para uma potencial associação entre as duas categorias, ou seja, uma validação de que não há uma partição rígida entre esses dois grupos no campo; separação essa que se torna então ou ideal ou esporádica. Em outras palavras, parece cada vez mais apropriado observar este campo ocupacional como uma escala entre dois extremos, tal como explicitado na *Tabela 27*. Por um lado, isso decorre em reconhecer a influência ubíqua do mercado em flexibilizar e se apropriar de conceitos antes mais politizados a fim de incentivar uma aplicação técnica, o que significa, por exemplo, a vitória prática da noção de código aberto sobre a de *software* livre. Por outro lado, também devemos considerar (ou lembrar) a influência indireta que a cultura *hacker* causou não apenas sobre desenvolvedores de *software* livre, mas também sobre toda a categoria de desenvolvedores e programadores, já que é possível observar aí os valores de compartilhamento, de trabalhar em benefício da sociedade, dentre outros.

Cabe agora uma breve exposição sobre cada um dos componentes propostos, relacionando-os a outros elementos empíricos ou teóricos. O fator motivacional problematizado há pouco foi citado por ser justamente o primeiro componente agrupado pela análise: o *altruísmo comunitário*.

Apesar de parecer um tanto quanto idealizada no contexto em que estamos construindo, a noção de altruísmo é conceitualizada a partir das discussões organizacionais do desenvolvimento de SLCA. Baytiyeh e Pfaffman (2010) consideram o “altruísmo” como a motivação principal dessa classe de trabalhadores, tratada especificamente como uma vontade de compartilhar e produzir conhecimento para um bem social, além da valorização do aprendizado próprio. Isso acontece em oposição à motivação financeira, que naquela pesquisa não impactou significativamente para os motivos de contribuir com a programação de *software* livre. O ideal altruísta se manifesta no seguinte trecho de um desenvolvedor de *software* livre entrevistado por mim:

O que eu mais gosto [no meu trabalho] é de poder contribuir, eu ser pago pra contribuir para alguma coisa em que eu acredito [...], porque dá muito orgulho de poder sobreviver fazendo aquilo que você acredita. Tão poucas pessoas conseguem fazer isso, tão poucas pessoas, então eu acho que tenho que agradecer todos os dias. (Freedo)

Como contraponto à síntese dos componentes na comparação SLCA-SNL, oferecerei algumas tabelas onde houve uma correlação com significância estatística relevante na comparação desejada. Já entre aquelas variáveis em que não há correlação, o teste manual em que comparei as variáveis demonstra surpreendentemente proporções muito próximas das características de desenvolvedores de SLCA com aquelas de desenvolvedores de SNL em praticamente todas as variáveis, o que seria enfadonho reproduzir na íntegra aqui³¹. Nessa análise complementar, as correlações são mais bem visualizadas utilizando a escala que permite a comparação entre quem só desenvolve SLCA, quem só desenvolve SNL e quem desenvolve *software* por ambos os modelos.

Assim, podemos observar nas *Tabelas 44 e 45* que houve uma maior identificação com os motivos do componente *altruísmo comunitário* entre desenvolvedores de SLCA (e em um grau relativamente intermediário se manifestaram desenvolvedores “híbridos”, confirmando o comportamento de uma escala). Não apresentei em uma tabela, mas a variável “orgulho em fazer parte de uma comunidade” apresentou uma dinâmica semelhante às outras duas deste componente. Devido à aparente contradição entre o enfraquecimento da hipótese e essa correlação significativa, é preciso se atentar à forma que se dá a articulação desses elementos explicativos: a análise dos componentes indicou apenas que a estrutura geral das motivações passa pela mesma tipologia entre todos(as) desenvolvedores(as) do campo, mas dentro o caso específico dos valores *altruístas* há uma relevância maior (embasada na significância estatística destas correlações) entre os desenvolvedores de SLCA, indo de acordo com o caminho teórico que estive construindo até aqui.

³¹ As demais variáveis que apresentaram uma correlação estatística, mas que escolhi não trazer para esta explicação foram: “Diversão em programar” e “Sempre desejei trabalhar na área de tecnologia”, na qual aqueles que desenvolvem tanto SLCA quanto SNL tiveram predominância positiva.

Tabela 44 – Concordância com o motivo “contribuir com o desenvolvimento da sociedade” segundo tipo de desenvolvimento de *software*

[Motivo] Contribuir com o desenvolvimento da sociedade	Desenvolve					
	Apenas SLCA		SLCA e SNL		Apenas SNL	
	N	%	N	%	N	%
Concordo Totalmente	27	62,8%	39	53,4%	37	39,8%
Concordo Parcialmente	14	32,6%	27	37,0%	44	47,3%
Não Concordo	2	4,7%	7	9,6%	12	12,9%
Total	43	100,0%	73	100,0%	93	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Tabela 45 – Concordância com o motivo “compartilhar meus conhecimentos e habilidades” segundo tipo de desenvolvimento de *software*

[Motivo] Compartilhar meus conhecimentos e habilidades	Desenvolve					
	Apenas SLCA		SLCA e SNL		Apenas SNL	
	N	%	N	%	N	%
Concordo Totalmente	31	70,5%	51	69,9%	52	55,9%
Concordo Parcialmente	13	29,5%	18	24,7%	34	36,6%
Não Concordo	0	0,0%	4	5,5%	7	7,5%
Total	44	100,0%	73	100,0%	93	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Acrescento a essas considerações mais um trecho de entrevista elucidativo:

Olha, a colaboratividade que a gente tem é uma coisa que eu acho fantástica. Eu acho que, sem dúvida, o que eu gosto [no trabalho] mais é aquela coisa de estar com problemas e ter sempre várias pessoas dispostas a ajudar esse problema. Eu acho que o que menos gosto [no trabalho] é quando isso não acontece, muitas vezes acontece das pessoas esconder informação ou não poder, por razões quaisquer, liberar essa informação, em alguns casos atrapalhar intencionalmente o desenvolvimento. (Geeko)

Em seguida, o componente *profissional* se refere a uma manifestação da expertise, do saber acumulado (não apenas técnico, mas também apoiado em princípios abstratos, formais e em consensos tácitos) por trabalhadores especializados e que desejam sempre se superarem, apresentando novos resultados. Entre os desenvolvedores de SNL houve uma associação do componente com a variável “obrigação por uma circunstância profissional”, o que indica envolvimento com uma área ou método que não é preferida por sua própria

motivação ou paixão, em detrimento de uma disposição para ascender na carreira, em aceitar o que o mercado valoriza.

Apesar de essa não ter sido a interpretação que relacionei originalmente com as variáveis durante a preparação do questionário, me parece que a associação das variáveis “resolver um problema que não poderia ser resolvido por outro software existente” e “diminuir o poder de grandes companhias de software” nesse componente têm em comum o desenvolvimento do ideal de profissionalismo. É admissível relacionar a noção de “descentralizar” o poder de grandes companhias com a aspiração de se opor à regulação profissional gerida pelo mercado. O sentido exposto aqui parte do contexto argumentativo na qual Freidson (2001) propõe a tipologia das instâncias reguladoras do profissionalismo em conflito: o mercado, o Estado e o terceiro setor (por meio de entidades como conselhos de classe, entidades de categoria, etc).

Isso se constitui ainda que levemos em conta as dificuldades e tensões desse campo ocupacional em definir sua área de atuação de acordo com os requisitos tradicionais, numa arena política onde há uma divisão entre aqueles que defendem uma regulamentação formal da profissão da Tecnologia da Informação (baseada na expertise específica) e os que acreditam na autorregulação do mercado como uma melhor opção (NUNES; MELLO; SOUZA, 2015).

Desta maneira, é possível que a variável relacionada à descentralização do poder de companhias de *software* tenha sido interpretada também a partir da relação com oferecer um produto (*software*) inovador no mercado, tendo por base a associação das duas variáveis. O que reforça isso é a observação de que a variável “descentralização” possui um número menor de identificação do que a variável “solução”.

Ainda assim, a *Tabela 46* demonstra que os(as) desenvolvedores(as) de SLCA possuem bem mais afinidade com a descentralização, tal como esperado.

Tabela 46 – Concordância com o motivo “Diminuir o poder de grandes companhias de software” segundo tipo de desenvolvimento de software

[Motivo] Diminuir o poder de grandes companhias de software	Desenvolve					
	Apenas SLCA		SLCA e SNL		Apenas SNL	
	N	%	N	%	N	%
Concordo Totalmente	12	27,9%	7	9,7%	3	3,3%
Concordo Parcialmente	5	11,6%	16	22,2%	13	14,0%
Não Concordo	26	60,5%	49	68,1%	74	82,2%
Total	43	100,0%	72	100,0%	90	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

A ânsia por solucionar desafios técnicos foi um elemento que eu não havia previsto dentre a estruturação das definições de *hacker* configuradas na *Tabela 37*. Esse tema reaparece com a importância de ressaltar a imersão na programação em detrimento de outras atividades relacionadas com o desenvolvimento, como podemos observar no trecho a seguir:

— *Você se lembra de alguma coisa que pode ser recente ou no passado que desmotivou?*

— [...] Nessa atividade específica o que mais me desmotiva é quando alguém tentar botar algum freio em mim “isso você não deve fazer” ou tenta colocar burocracia naquilo que eu vou desenvolver. Quando eu vou desenvolver software não preciso de papel e não é um negócio tipo: prédio que você vai subir, planta que alguém tem que assinar aquela planta e nem nada, e não está colocando o risco na vida de ninguém e tal. Se der um problema simplesmente dá um passo atrás e volta. Acho quando as pessoas começam a burocratizar demais começa a dar problemas. Incomodo com essas coisas assim. Se eu pudesse simplesmente sentar e trabalhar e escrever código oito horas por dia seria ótimo. Mas infelizmente tem essas coisas burocráticas e tem que tratar isso e é um pé no saco. (Phoenix)

Continuando para o terceiro componente principal aludido, a *busca por reconhecimento* se concentrou em ambas as amostras (SLCA e SNL) na variável “ganhar reputação entre outros programadores”. A partir daqui podemos notar motivações mais instrumentais do que ideológicas, pois há uma maior relação com o estabelecimento de uma posição de destaque na carreira, por exemplo.

Stewart (2005) concatena a noção de reputação ao status social, admitindo-a como um importante elemento da dinâmica de produção de SLCA, ainda que porventura não seja percebido com tanta clareza. Isso porque os membros de uma

comunidade tendem a avaliar uns aos outros a partir das referências sociais disponíveis, o que acaba restringindo a capacidade de mobilidade de status dessas pessoas. O autor cita o efeito Mateus³² cunhado por Merton (1968)³³ como uma síntese desse tipo de dinâmica em que possuir reputação diminui obstáculos e aumenta incentivos para um dado agente, enquanto que possuir pouca ou nenhuma reputação causa uma forte tendência de não ter o trabalho reconhecido.

Se lembrarmos a definição de Eric Raymond³⁴ de um *hacker* como aquele(a) que simplesmente é reconhecido como tal por outro(a) *hacker*, podemos enfim esclarecer como a reputação, um tópico frequentemente mencionado pelos(as) desenvolvedores de SLCA, é fundamental, ademais ao mérito do código mais bem escrito. A reputação, além de um motivador do *self*, se constitui como um elemento curricular informal.

Por um lado, na comunidade de *software* livre, é possível a busca por reputação apenas como um valor em si mesmo, como a apresentação do *self* de um programador de grandes capacidades, em especial nos casos em que um desenvolvedor pode contribuir voluntariamente para outros projetos também. Por outro lado, podemos levar em consideração a reputação como um fator profissional, ao observar que, entre os desenvolvedores de SNL, a variável “reputação” esteve relacionada também às ideias de “ascensão” (econômica e profissional) e de “aumentar minhas oportunidades de emprego”, fortalecendo assim a interpretação exposta acima.

Por fim, o perfil *empresarial* se refere à forma mais direta pelo interesse econômico da atividade. Em ambas as amostras foram relacionadas as variáveis: “vender produtos ou serviços”, “aumentar minhas oportunidades de emprego”, “ascensão econômica e profissional” e “resolver um problema que não poderia ser resolvido por outro software existente”. Eis uma interpretação interessante de um desenvolvedor de *software* livre sobre a questão das diferenças empresariais:

32 “Pois a quem tem, mais será dado, e terá em grande quantidade. Mas a quem não tem, até o que tem lhe será tirado.” (Mt 25:29, BÍBLIA, 2001).

33 Robert Merton utilizou essa parábola para exemplificar a dinâmica de aprofundamento das desigualdades e é particularmente mais citada hoje nos contextos educacionais.

34 Citado na página 28.

— *O que distingue um desenvolvedor de software livre de um desenvolvedor software proprietário?*

— Eu tenho alguns amigos que desenvolvem software proprietário, inclusive um deles trabalha na Osmium³⁵ [...]. Basicamente o que distingue é o viés social. É o viés de você preocupar em termos se isso aqui é bom para o mundo, bom para a sociedade, e não só se isso aqui que estou fazendo é bom pra empresa e eu estou ganhando dinheiro. [...] Eu lembro que um pouco depois que ele entrou [na Osmium], a gente estava conversando pela internet [sobre o patch que] eu tinha enviado pro GDB, como é software livre as modificações que a gente vê são abertas, pode procurar na internet que estão todas as modificações que eu enviei lá, ninguém fecha nada. Aí eu estava conversando com ele e achei a modificação que eu tinha feito, que eu tinha achado legal, eu falei poxa! Dá uma olhada aí e enviei o link, [e] ele falou: infelizmente eu não posso olhar, não posso abrir, porque eu assinei um contrato com a empresa que me proíbe ver código livre, código aberto, porque senão a gente pode de certa forma ficar contaminado com aquilo, depois utilizar um [algoritmo] no software da Osmium, isso pode gerar depois um conflito, pode gerar até um processo porque é GPL que é uma licença de software livre. [...] Com o código [em] GPL, você precisa abrir todo o código que utiliza aquela parte. Então a Osmium morre de medo disso. Aí existem contratos dizendo que você não pode. Em contrapartida, [...] quando entrei na Panama³⁶, eu recebi um papel que era o contrato da Panama e no papel dizia o seguinte você está liberado, estou ciente [...] que estou liberado para contribuir para qualquer projeto de software livre que você queira, mesmo que esse projeto vá contra os interesses da empresa. [...] Então a Panama não pode me demitir se eu contribuir para um projeto de software livre que é concorrente de outra que eles utilizam para distribuição. Então isso muda as mentalidades das empresas. (Freedo)

5.3.2 Cultura do Trabalho no Desenvolvimento de Software

Esta última seção apresenta os resultados obtidos da segunda análise de componentes, a respeito das variáveis ligadas diretamente à noção de cultura do trabalho. Inevitavelmente a conceitualização de cultura do trabalho só é mensurável como uma variável latente, isto é, o questionário precisa conter um conjunto de diferentes variáveis sobre o tema, que então em conjunto representam essa variável subjacente. Porém, antes de nos determos sobre a análise multivariada, vale a pena observar as frequências descritivas de cada variável, a fim de ter uma noção geral de quais questões são preponderantes e quais se revelam com uma influência reduzida. Vide a *Tabela 47*:

³⁵ Codinome de uma empresa multinacional de *software* em modelo proprietário.

³⁶ Codinome de uma empresa internacional especializada em *software* livre e código aberto.

Tabela 47 – Percepção de elementos da cultura e do ambiente de trabalho entre desenvolvedores de software

Avalie as afirmações em relação a seu ambiente de trabalho a partir da escala:	Percentual relativo à linha		
	Não concordo	Concordo Parcialmente	Concordo Totalmente
Predomina um espírito de trabalho em equipe entre meus colegas. [equipe]	8,8%	46,8%	44,4%
A equipe de desenvolvimento possui liberdade para fazer seu trabalho da forma como prefere. [liberdade]	15,9%	44,0%	40,1%
As decisões sobre a execução do projeto são de responsabilidade de toda a equipe. [decisão]	21,5%	42,9%	35,6%
Os horários de trabalho são estabelecidos conforme a necessidade da empresa ou dos clientes. [horários]	32,1%	39,0%	29,0%
A maneira como se deve realizar o trabalho constantemente é mudada. [reformulação]	30,5%	48,1%	21,4%
Os colegas de trabalho comentam seus problemas pessoais uns com os outros. [intimidade]	16,7%	60,1%	23,2%
Há um ambiente agradável de trabalho. [agradável]	7,9%	42,1%	50,0%
Os funcionários trabalham bem, mesmo na ausência do chefe ou gerente. [autonomia]	13,2%	42,0%	44,9%
A organização coletiva do trabalho é eficiente. [eficiente]	21,7%	48,3%	30,1%
O trabalho frequentemente é entediante. [entediante]	55,6%	31,9%	12,6%
Frequentemente há pressão para entregar resultados em pouco tempo. [agilidade]	16,1%	50,2%	33,7%
Tenho uma separação bem definida entre horas de trabalho e horas livres. [jornada]	20,0%	43,2%	36,7%
A qualidade do trabalho final é priorizada em relação à agilidade. [qualidade]	22,9%	55,6%	21,5%
É importante saber se relacionar com pessoas de outras áreas de Tecnologia de Informação. [sociável]	1,4%	16,0%	82,7%
Para realizar meu trabalho é importante buscar conhecimentos com outros profissionais ou empresas. [intercâmbio]	7,5%	41,3%	51,2%
Sinto que estou competindo com colegas de minha empresa ou equipe. [competição]	59,7%	32,8%	7,5%
Minha empresa oferece uma estabilidade que me permite continuar empregado(a) por muitos anos. [estabilidade]	23,8%	37,1%	39,1%
Meu salário varia de acordo com minha [produtividade].	65,4%	18,2%	16,4%
O software que desenvolvo é direcionado para uso específico de um cliente, e não para o público genérico. [direcionado]	24,4%	27,8%	47,8%
Para terminar um projeto no prazo, é mais importante improvisar do que cumprir o planejamento inicial. [improviso]	46,5%	40,8%	12,8%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Nota: Foi considerada como ausente a opção de resposta disponível “Não se aplica ao meu caso”.

Em quase todas essas perguntas, a opção aqui ignorada “Não se aplica ao meu caso” possuiu uma proporção relativamente inferior tanto aos casos não respondidos quanto à opção válida correspondente com o menor percentual: uma média de 11,5 respostas ou 4,3% do percentual total. As duas exceções que motivam essa explicação foram as variáveis referentes à *Competição* (“sinto que estou competindo com colegas de minha empresa ou equipe”) e à *Produtividade* (“meu salário varia de acordo com minha produtividade”).

De qualquer forma, a variável mais frequentemente considerada foi a [sociável] (“é importante saber se relacionar com pessoas de outras áreas de Tecnologia de Informação”), que, no entanto, não foi relacionada a nenhum dos componentes principais indicados em seguida. Isso talvez signifique que essa é uma opinião/representação comum de se concordar, mas que não implica necessariamente em uma prática de socialização específica, o que indica que a variável deveria ter sido melhor formulada para indicar mais claramente que eu estava interessado em compreender uma prática de interação e não uma opinião do que é bom.

A variável com menos concordância apontada pela *Tabela 47* é a [produtividade] (“meu salário varia de acordo com minha produtividade”). E é curioso aperceber na *Tabela 48* que suas frequências podem ser comparadas com algumas categorias precarizadas da posição na ocupação: os regimes de trabalho “[por] conta própria” e “empregado sem [carteira de] trabalho assinada” estão relacionados com a variação do salário conforme a produtividade. Nessa última correlação, o menor índice de “não se aplica ao meu caso” está entre as pessoas que trabalham por conta própria, ao mesmo tempo que essa categoria possui a maior porcentagem relativa de concordância total com a situação de variabilidade do rendimento de acordo com a produtividade.

Tabela 48 – Concordância com o elemento “meu salário varia de acordo com minha produtividade” segundo posição na ocupação

Posição na Ocupação	Concordo Totalmente		Concordo Parcialmente		Não concordo		Não se aplica ao meu caso	
	N	%	N	%	N	%	N	%
Não remunerado	0	0,0%	1	25,0%	2	50,0%	1	25,0%
Conta Própria	14	41,2%	8	23,5%	7	20,6%	5	14,7%
Empregador	4	30,8%	5	38,5%	7	7,7%	3	23,1%
Empregado(a) sem carteira de trabalho	3	13,0%	2	8,7%	13	56,6%	5	21,7%
Empregado(a) com carteira de trabalho	3	2,5%	11	9,0%	71	58,2%	37	30,3%
[...] funcionário público	2	8,7%	2	8,7%	10	43,5%	9	39,1%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Seguindo então para a análise de componentes principais, a *Tabela 49* apresenta a variância total explicada pelos componentes considerados, com autovalor (*eigenvalor*) maior que 1. O método de extração e de rotação foi o mesmo da análise anterior. O índice de Kaiser-Meyer-Olkin da amostra de SLCA indicou o valor de 0,687, e a de SNL indicou 0,711, ambos bons indicadores desse tipo de relevância estatística, enquanto o teste de Bartlett foi mais uma vez favorável. Enfim, a *Tabela 50* apresenta a matriz comparada com todos os componentes.

Tabela 49 – Variância total explicada da análise de variáveis de cultura do trabalho

Componentes Reescalados	Somadas de extração de carregamentos ao quadrado					
	SLCA			SNL		
	Total	% da variância	% cumulativa	Total	% da variância	% cumulativa
1	3,400	18,888	18,888	3,252	18,105	18,105
2	2,034	11,299	30,187	2,132	11,844	29,949
3	1,457	8,094	38,281	1,375	7,638	37,587
4	1,524	8,467	46,748	1,285	7,138	44,725
5	1,095	6,084	52,833	1,091	6,062	50,787
6	1,080	5,998	58,831	1,045	5,804	56,591

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Tabela 50 – Matriz padrão dos escores de componentes da análise de cultura do trabalho segundo tipo de desenvolvimento de *software*

Variáveis relacionadas à cultura do trabalho	Componentes Reescalonados											
	Software Livre e Código Aberto						Software Não-Livre					
	1	2	3	4	5	6	1	2	3	4	5	6
[equipe]	,802	-,032	,040	-,206	-,037	,028	,738	-,063	,040	-,021	-,042	-,095
[liberdade]	,378	-,153	-,132	,190	-,243	,584	,251	,067	,544	,017	,023	-,388
[decisão]	,675	,020	,161	,125	-,277	-,137	,449	-,095	,409	-,048	,067	-,179
[flexível]	-,061	,110	,079	,189	-,877	-,045	,001	-,480	,110	-,031	-,940	-,049
[reformulação]	-,183	,284	-,277	-,194	-,333	-,020	-,034	,752	,089	-,198	,108	,064
[intimidade]	,356	,362	-,159	-,267	-,307	,225	,546	,560	,019	-,013	-,104	-,028
[agradável]	,473	-,176	,088	,340	,038	-,024	,278	-,387	,185	-,103	-,072	-,039
[autonomia]	,762	,023	,127	-,137	,125	,105	,625	-,050	-,173	-,012	-,145	-,145
[eficiente]	,781	,084	-,356	-,026	,148	,087	,711	-,029	,000	-,004	,221	,180
[entediante]	-,243	,363	-,074	-,147	,051	,594	-,288	,553	-,056	,217	,077	-,339
[agilidade]	,064	,748	,182	,066	-,046	,045	,148	,446	,099	,471	-,162	,323
[jornada]	,118	,006	,439	-,109	-,219	-,030	,116	,018	,065	,021	,333	-,464
[qualidade]	,051	-,159	-,092	,189	-,053	-,194	,179	-,235	,140	-,253	-,020	-,160
[competição]	,094	-,045	,079	,179	,094	,639	,078	,200	-,265	,217	-,174	-,296
[estabilidade]	-,025	-,654	,215	-,361	-,228	,511	-,037	-,026	-,019	-,019	-,173	-,828
[produtividade]	-,205	,176	-,010	,894	-,136	,206	-,211	,104	,917	,084	-,113	,100
[direcionado]	-,110	,205	,857	,068	,141	,089	-,015	-,295	,075	,960	,060	-,069
[improvisado]	-,018	,664	,038	,000	-,056	,006	-,024	,574	,009	-,010	-,118	-,043

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Nota: Os rótulos das variáveis foram abreviados e a versão completa da questão está na *Tabela 47*. Para facilitar a leitura da tabela foram destacadas as cargas com valor acima de 0,4. A matriz SLCA teve a rotação convergida em 14 iterações, e a matriz SNL 13 iterações ambas pelo método Oblimin com normalização de Kaiser.

Tal como justificado na análise empreendida na seção anterior, apresento no *Quadro 2* uma síntese interpretativa do significado de cada componente, agrupando os nomes das variáveis relacionadas em cada componente principal.

Quadro 2 – Síntese dos rótulos tipificados para a análise de componentes de cultura do trabalho entre desenvolvedores de software

	SLCA		SNL	
	Rótulo	Variáveis Relevantes	Rótulo	Variáveis Relevantes
1	<i>Equipe proativa</i>	[equipe]; [eficiente]; [autonomia]; [decisão]; [agradável]	<i>Equipe proativa</i>	[equipe]; [eficiente]; [autonomia]; [intimidade]; [decisão]
2	<i>Pressão pelo prazo</i>	[agilidade]; [improviso]; -[estabilidade]	<i>Pressão pelo prazo</i>	[reformulação]; [improviso]; [intimidade]; [entediante]; [agilidade]
3	<i>Direcionado ao cliente</i>	[direcionado]; [jornada]	<i>Liberdade de decisão</i>	[produtividade]; [liberdade]; [decisão]
4	<i>Freelancer</i>	[produtividade]	<i>Direcionado ao cliente</i>	[direcionado]; [agilidade]
5	<i>Horário flexível</i>	[flexível]	<i>Horário flexível</i>	[flexível]
6	<i>(Resíduo?)</i>	[competição]; [entediante]; [liberdade]; [estabilidade]	<i>Estabilidade</i>	[estabilidade]; [jornada]

Nota: As variáveis estão dispostas segundo a ordem da maior carga para a menor.

Essa análise, ao contrário da anterior, já aponta para algumas diferenças entre as duas amostras, ainda que não se configurem como totalmente distintas. Os dois primeiros componentes principais (*equipe proativa* e *pressão pelo prazo*) são os que explicam uma porção maior da variância (19% e 11%, tal como na *Tabela 49*) e possuem pouca diferença entre as duas amostras, o que me fez considerá-las como possuindo o mesmo valor semântico.

A partir do terceiro componente principal, há diferenças na ordem em que os componentes aparecem na extração. O componente *direcionado ao cliente* possui essa característica, mas podemos considerar que é o mesmo em ambas pois apresenta as mesmas variáveis constituintes. Enquanto isso, dois componentes apontam para a distinção entre as duas amostras: a *liberdade de decisão* só aparece entre os desenvolvedores de SNL, pois essas características (em especial a de “[decisão]” no primeiro componente) já estão subsumidas nos outros componentes; e a de *freelancer* entre desenvolvedores de SLCA, talvez porque a categoria de trabalho por conta própria ser mais frequente entre esses desenvolvedores, tal como visto na *Tabela 31*. E por fim, o fator *estabilidade* aparece com mais clareza entre o SNL, já que na amostra de SLCA há um

agrupamento de várias variáveis com pouca relação semântica entre si, o que me faz conjecturar que talvez se trate mais de um resíduo estatístico dos outros componentes do que uma relação significativa para a interpretação sociológica.

Por conseguinte, temos à disposição uma tipologia de perfis distintos idealmente sobre a cultura do trabalho de desenvolvedores de *software*: o primeiro tipo (*equipe proativa*) é um modo de trabalho em que predomina uma equipe comunicativa, eficiente, autônoma, com responsabilidade para decisões, e que pode inclusive gerar um ambiente de trabalho mais agradável ou de proximidade entre os colegas; no segundo tipo (*pressão pelo prazo*) predomina a referência ao prazo, com influência da agilidade, improviso e reformulação do projeto, e que dentro o SLCA há uma contraposição com o modelo da estabilidade de carreira; o tipo *direcionado ao cliente* indica uma cultura organizacional, como o nome indica, em que o planejamento e implementação da programação de *software* é pautada nas necessidades de um cliente específico, cujas interações com os desenvolvedores são mais intensas; outro tipo organizacional é aquele em que o direcionamento do cliente e/ou das instâncias superiores da empresa são ainda mais influentes ao ponto de flexibilizar os horários de trabalho dos desenvolvedores; a única forma restante que se aproxima desse ritmo de trabalho é o de *freelancer*, em que também a composição do rendimento é determinada pelo ritmo de produção; e por contraponto, um último tipo claramente identificável é o do trabalho *estável* em uma empresa, em que um(a) desenvolvedor(a) pode se manter por muitos anos, o que geralmente ocorre em algumas grandes empresas internacionais de *software* e também em empresas públicas nacionais, devido ao vínculo como funcionário(a) público.

Posto isso, vamos observar mais alguns detalhes desses componentes relacionados à cultura do trabalho. O primeiro componente, *equipe proativa*, é o que reúne mais elementos citados na literatura sobre cultura do trabalho no SLCA, portanto nos tomará mais fontes de explicação. Sua característica primordial é a da percepção de um espírito de trabalho em equipe entre seus participantes, o que, apesar de estar presente nas duas amostras, é mais comum entre desenvolvedores(as) de SLCA, tal qual expresso na *Tabela 51*.

Tabela 51 – Concordância com o elemento “predomina um espírito de trabalho em equipe” segundo tipo de desenvolvimento de *software*

Predomina um espírito de trabalho em equipe entre meus colegas	Desenvolve					
	Apenas SLCA		SLCA e SNL		Apenas SNL	
	N	%	N	%	N	%
Concordo Totalmente	20	57,1%	38	56,7%	32	35,6%
Concordo Parcialmente	13	37,1%	24	35,8%	49	54,4%
Não Concordo	35	5,7%	5	7,5%	9	10,0%
Total	35	100,0%	67	100,0%	90	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

É razoável supor que a própria maneira de organizar o trabalho com o código aberto exija uma colaboratividade maior quando o programa se fundamenta efetivamente na forma *bazar* de desenvolvimento (RAYMOND, 2012). Cabe acrescentar que o próprio estilo da liderança de um projeto de código aberto impacta na motivação dos desenvolvedores em contribuir (LI; TAN; TEO, 2012), tanto nas motivações “intrínsecas” (subjetivas, emocionais ou interpretativas) quanto nas “extrínsecas” (organizacionais, materiais ou estruturais).

Quais princípios então atestam por uma boa equipe de desenvolvimento de *software*? Sánchez (2011) seleciona as seguintes características como as principais determinantes da inovação em empresas de *software* livre: um ambiente de trabalho em que predomina o incentivo e não punição por falhas; a autonomia dos empregados em tomar decisões; a valorização da criatividade, colaboração e compartilhamento de conhecimento, dentre outras. Dessa forma, esse ambiente positivo de trabalho é percebido com maior frequência entre desenvolvedores(as) de SLCA, como podemos observar na *Tabela 52*, o que fez com que essa variável (trabalho agradável) fosse incluída no componente principal *equipe proativa* apenas na amostra desse grupo.

Tabela 52 – Concordância com o elemento “ambiente agradável de trabalho” segundo tipo de desenvolvimento de *software*

Há um ambiente agradável de trabalho	Desenvolve					
	Apenas SLCA		SLCA e SNL		Apenas SNL	
	N	%	N	%	N	%
Concordo Totalmente	22	53,7%	46	66,7%	34	37,0%
Concordo Parcialmente	19	46,3%	17	24,6%	50	54,3%
Não Concordo	0	0,0%	6	8,7%	8	8,7%
Total	36	100,0%	66	100,0%	90	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

No entanto, cabe lembrar também que frequentemente o trabalho em SLCA se dá com interações predominantemente virtuais, com menos interações presenciais com a equipe de colegas. Uma das razões para isso é o menor número relativo de empresas especializadas em SLCA no Brasil proporcionalmente ao número de profissionais especializados disponíveis, o que faz vários trabalharem remotamente para empresas estrangeiras. E talvez por influência desse aspecto, a variável de intimidade tenha constado apenas na amostra de SNL do componente respectivo. O seguinte depoimento atesta para o caráter subjetivo desse tipo de interação virtual com colegas de trabalho:

O que eu menos gosto é a solidão. Sem dúvida nenhuma! [...] As pessoas às vezes olham pra mim e acham: nossa que vida boa e não sei o que lá. Mas, elas não sabem. A gente está sempre insatisfeito com o que a gente tem. O que mais me incomoda é a solidão, morar sozinho. [...] fiquei 10 meses sem conhecer ninguém com quem eu trabalhava. Em 2011 meu gerente foi trocado, aí eu não fui mais em reunião do time, então hoje eu não conheço o meu gerente. Eu converso com ele por texto, por telefone, mas não sei como ele é, isso pra mim é muito estranho. Eu sinto falta do contato com o pessoal, sinto muita falta. As maiorias das pessoas trabalham em casa ou porque elas já são casadas e elas já tem essa infraestrutura pra suprir o contato ou porque elas são muito antissociais. Inclusive eu trabalho com um cara que é assim ele mora no cafundó da República Tcheca e ele fala que mora lá porque não gosta de gente, não gosta de ver gente e fica meses sem ver gente, porque não gosta. Então pra ele é o melhor trabalho do mundo. Eu já não sou assim não tenho, não sou casado, não tenho namorada, não tenho filhos, então eu fico sozinho mesmo. Isso me incomoda. (Fredo)

Quanto ao modelo de trabalho pautado no direcionamento ao cliente, nota-se uma aproximação às características organizacionais descritas por Ó Riain (2010), mais comuns no desenvolvimento de SNL, tal como vemos na *Tabela 53*.

Tabela 53 – Concordância com o elemento “desenvolvimento direcionado para uso específico de um cliente” segundo tipo de desenvolvimento de software

O software que desenvolvo é direcionado para uso específico de um cliente, e não para o público genérico	Desenvolve					
	Apenas SLCA		SLCA e SNL		Apenas SNL	
	N	%	N	%	N	%
Concordo Totalmente	11	28,2%	34	51,5%	45	51,1%
Concordo Parcialmente	13	33,3%	18	27,3%	25	28,4%
Não Concordo	15	38,5%	14	21,2%	18	20,5%
Total	39	100,0%	66	100,0%	88	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Esse contexto é ainda problematizado por (ROLANDSSON; BERGQUIST; LJUNGBERG, 2011): empresas que desenvolvem *software* proprietário foram aos poucos incorporando modelos de código aberto, num sistema de desenvolvimento híbrido. Esse tipo de transformação empresarial parte de tensões externas, a respeito da forma de organizar o trabalho informacional. E para implementar isso, os(as) programadores(as) criam diversas estratégias de adaptação entre as demandas ideológicas da comunidade de SLCA e os interesses do mercado.

É interessante também notar o que apresenta a *Tabela 54*, em que os desenvolvedores de SLCA afirmam se entediar com menos frequência do que os de SNL. Por mais que essa característica isoladamente seja difícil de ser explicada, vale manter em mente que o trabalho no SLCA é na maioria das vezes uma escolha de especialização feita desde o início da formação, sendo alimentada pela paixão ao modelo de programação e compartilhamento referido.

Tabela 54 – Concordância com o elemento “trabalho frequentemente entediante” segundo tipo de desenvolvimento de software

O trabalho frequentemente é entediante	Desenvolve					
	Apenas SLCA		SLCA e SNL		Apenas SNL	
	N	%	N	%	N	%
Concordo Totalmente	2	5,4%	8	12,3%	15	16,5%
Concordo Parcialmente	11	29,7%	17	26,2%	33	36,3%
Não Concordo	24	64,9%	40	61,5%	43	47,3%
Total	37	100,0%	65	100,0%	91	100,0%

Fonte: Questionário aplicado pelo pesquisador no período de janeiro a fevereiro de 2015.

Porventura essa diferença no que é considerado enfadonho se expresse também nas escolhas de flexibilização do modo de trabalhar, porque o trabalho estável em uma só empresa ou projeto é considerado por alguns o elemento entediante. Nos trechos de entrevista a seguir é possível observar como se dá a ambivalência entre fugir da monotonia e seguir em um só ritmo de carreira:

— *Você considera seu trabalho estável?*

— Nunca considereicho que isso é uma das coisas piores que eu não quero. Um trabalho estável não me daria a oportunidade do limite das coisas, ter um trabalho estável é mais burocrático. Eu não gosto de trabalho estáveis, eu não estou num trabalho estável agora e provavelmente eu não estaria do trabalho estável nunca.

— *Você já trabalhou como freelancer?*

— Já por quase um ano e nunca mais. [...] Bom, eu conseguia ganhar muito bem a cada hora que eu trabalhasse, mas eu não conseguia trabalhar muitas horas e isso é muito ruim, porque eu me sentia inútil, eu podia aproveitar muitas horas pra trabalhar e eu ficava parado, procurando o que fazer, procurando cliente e envolve muitas coisas que eu não gostaria de ter que fazer, então envolve burocracia em geral. Os meus primeiros trabalhos foram coisas tipo acertava um preço e era curto pra essa tipo de coisa é bem tranquilo, ao menos que você fica sem trabalho com frequência, você trabalha a semana inteira, a não ser se você fecha um projeto, depois fica uma semana inteira sem trabalhar. Isso pra mim é muito frustrante, eu gosto de trabalhar. Ficar sem trabalhar é uma coisa muito frustrante. (Phoenix)

Por fim, levando em conta toda a construção acumulada, podemos concluir que as condições de trabalho parecem ser influenciadas com mais força pelos elementos estruturais do que pelos intersubjetivos. Deste modo, mesmo que não seja viável estabelecer conexões causais a partir de análise do *survey*, já que não temos certeza da direção das relações, podemos supor que as diferenças entre SLCA e SNL se devem a fatores estruturais do mundo do trabalho que provavelmente não se originam numa diferença de sociabilidade anterior. Ainda assim, é razoável considerar que após os anos de formação e imersão no trabalho, uma sociabilidade e interiorização dos valores *hacker* se constitui.

6 Considerações Finais

Ao longo da pesquisa foi possível perceber uma preponderância de algumas formas de socialização: um aprendizado com tecnologias digitais desde a juventude, o que demonstra nos discursos uma paixão para além do trabalho; uma valorização (ou, pelo menos, uma boa adaptação) em relação ao trabalho à distância; um predomínio de homens tanto na categoria em geral quanto nas atividades consideradas mais técnicas (refletindo na clássica disparidade de salários), o que se fundamenta nas distinções de socialização de gênero, que por sua vez encoraja os homens a atividades mais técnicas e as mulheres às relações interpessoais; e por fim, um reconhecimento da importância da colaboração (que motiva o *software* livre) também para o ambiente de trabalho, com uma desaprovação de barreiras para a informação.

Foi possível notar a distinção de identidades e motivações entre desenvolvedores de código aberto, que interpretam apenas como uma boa oportunidade no mercado de trabalho, e de *software* livre *stricto sensu*, que possuem fortes convicções políticas, redundando num movimento social. E ainda, notamos que existem de fato diferenças entre desenvolvedores de *software* livre e proprietário, tanto na prática de seu trabalho quanto em suas formações. Malgrado isso, essas diferenças não advêm tanto (ou na maioria dos casos) de condições ideológicas de forma direta: essa discrepância não é constituída por estruturas organizacionais claramente distintas, mas se configura em um espectro de influências com maior ou menor grau.

Em se tratando da discussão central da relação entre a cultura *hacker*, os valores do *software* livre e a cultura do trabalho de desenvolvedores de *software*, foi possível constatar a vitalidade dos valores *hacker* não apenas entre desenvolvedores de *software* livre, mas também entre aqueles que trabalham exclusivamente com *software* não-livre. Nesse sentido, a ética e a cultura *hacker* possuem uma influência difusa sobre toda a classe de desenvolvedores (livre e proprietário), não sendo uma característica identificadora de um padrão de socialização única. Com efeito, os ideais *hacker* se impregnaram na cultura da

informática através da história como uma referência de valores, mais ou menos, praticados.

Ainda assim, as comparações entre os dois modelos indicaram que em geral há uma maior tendência de que os desenvolvedores de SLCA se interessem por motivações mais ideológicas do que econômicas. Não obstante, por vezes ficou nítido que as diferenças não são tão fundamentais e absolutas como se poderia imaginar a partir da hipótese, das construções históricas e teóricas ou de discursos sobre as diferenças no campo. As evidências parecem sugerir que as maiores diferenças entre o SLCA e SNL se devem a condições estruturais, como a disponibilidade de empregos em grandes companhias, a maior quantidade de pessoas por conta própria e/ou sem vínculo formal no SLCA.

Além disso, a análise de componentes principais pôde se aprofundar na análise de minha hipótese central para então concluir que as distinções entre SLCA e SNL são ideais, ou melhor, estão em diferentes matizes de uma escala. A partir dessas considerações, um potencial metodológico futuro aponta para o refinamento da escala dentre os desenvolvedores “híbridos” para distinguir aqueles que trabalham por mais tempo e/ou ganham a maior parte de sua renda em um outro modelo de desenvolvimento. Compreender como se dão as zonas cinzentas de contratação, por exemplo, é uma área de grande interesse da sociologia do trabalho contemporâneo.

Sendo uma análise exploratória, houve apenas indicações de quais associações merecem ainda serem examinadas por outros tipos de análise porventura por outro(a)s pesquisadore(a)s. O banco de dados da pesquisa está publicado na Internet³⁷ com uma licença *copyleft* de utilização e compartilhamento abertos, com duas opções: *Creative Commons Atribuição-Compartilha Igual 4.0*³⁸ ou *Open Database License*³⁹. Há ainda um bom potencial metodológico para aplicação de técnicas avançadas de comparação entre as matrizes da análise de componentes principais (ROMNEY et al, 2000; KRZANOWSKI, 1979; GARCIA,

37 Disponível no endereço eletrônico <https://technica.neocities.org/scientias/datos/dev2015.html>, e futuramente no Consórcio de Informações Sociais <<https://nadd.prp.usp.br/cis>>.

38 <<https://creativecommons.org/licenses/by-sa/4.0/deed.pt-br>>

39 <<http://opendatacommons.org/licenses/odbl/summary/>>

2012) e de outras técnicas de análise multivariada, visto que produzi uma quantidade muito grande de variáveis para o *survey*.

Um dos limites do *survey* foi a exclusão de contribuidores voluntários de código como contraponto aos desenvolvedores de *software* livre contratados para esse fim, por isso, compreender até que ponto esse tipo de contribuição é fundamental para a sobrevivência de um projeto ainda precisa ser melhor explorado. Além disso, outros temas incluídos no *survey* não puderam ser tratados aqui, por exemplo, a relação entre as posições políticas dos desenvolvedores e o impacto nos valores técnicos e na cultura do trabalho.

Por fim, a importância de reconhecer os ideais *hacker* se revela no potencial de nos reconhecermos como *hackers*, de assumir um papel ativo e reinterpretativo em relação às tecnologias, desde os últimos aparelhos digitais até instituições mais abstratas como a linguagem cotidiana, a fim de aprofundar a transformação do cotidiano.

Referências

- APPLICATION PROGRAM. In: **PCMAG**, Encyclopedia. Disponível em: <<https://www.pcmag.com/encyclopedia/term/application-program>>. Acesso em: 22 jun. 2015.
- ASSANGE, Julian et al. **Cypherpunks: liberdade e o futuro da internet**. São Paulo: Boitempo, 2013.
- AUGUSTO, Maurício Pires. **Um estudo sobre as motivações e orientações de usuários e programadores brasileiros de software livre**. Rio de Janeiro, 2003. Dissertação (Mestrado em Administração) – Universidade Federal do Rio de Janeiro.
- BAUMAN, Zygmunt. **O mal-estar da pós-modernidade**. Rio de Janeiro: Jorge Zahar, 1998.
- BAYTIYEH, Hoda; PFAFFMAN, Jay. Open source software: a community of altruists. **Computers in human behavior**, [S.l.], v. 26, n. 6, p. 1345-1354, 2010. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0747563210000828>>. Acesso em: 3 ago. 2014.
- BECKER, Howard Saul. **Outsiders: estudos de sociologia do desvio**. Rio de Janeiro: Jorge Zahar, 2008.
- BERGER, Peter Ludwig; LUCKMANN, Thomas. **A construção social da realidade: tratado de sociologia do conhecimento**. 25. ed. Petrópolis: Vozes, 2005.
- BÍBLIA. Mateus. Português. In: **Bíblia sagrada**. Nova Versão Internacional. International Bible Society, 2001, Cap. 25, vers. 29. Disponível em: <<https://www.bibliaonline.com.br/acf+nvi/mt/25>>. Acesso em: 8 jun. 2015.
- BIG DATA. In: WIKTIONARY: the free dictionary. Wikimedia, 2019. Disponível em: <https://en.wiktionary.org/wiki/big_data>. Acesso em: 10 abr. 2019.
- BOLTANSKI, Luc; CHIAPELLO, Éve. **O novo espírito do capitalismo**. São Paulo: Martins Fontes, 2009.
- BOURDIEU, Pierre. **Contrafogos: táticas para enfrentar a invasão neoliberal**. Rio de Janeiro, Jorge Zahar, 1998.
- _____. **Razões práticas: sobre a teoria da ação**. 9 ed. Campinas (SP): Papyrus, 2008.
- BRIDI, Maria Aparecida; MOTIM, Benilde Lenzi. Trabalho e trabalhadores na indústria de informática. **Contemporânea**, São Carlos, v. 4, n. 2, p. 351-380, 2014. Disponível em: <<http://www.contemporanea.ufscar.br/index.php/contemporanea/article/view/238/113>>. Acesso em: 10 jun. 2016.
- BRIDI, Maria Aparecida; BRAUNERT, Mariana Bettega. O trabalho na indústria de software: a flexibilidade como padrão das formas de contratação. **Caderno CRH (Online)**, Salvador, v. 28, n. 73, p. 199-214, 2015. Disponível em: <<https://doi.org/10.1590/S0103-49792015000100013>>. Acesso em: 13 jun. 2016.

CASTELLS, Manuel. A cultura da Internet. In: _____. **A galáxia da Internet**: reflexões sobre a Internet, os negócios e a sociedade. Rio de Janeiro: Jorge Zahar, 2003. p. 34-55.

_____. **A sociedade em rede**. 8. ed. São Paulo: Paz e Terra, 2005.

CASTRO, Bárbara Geraldo de. **Afogados em contratos**: o impacto da flexibilização do trabalho nas trajetórias dos profissionais de TI. Campinas, 2013. Tese (Doutorado em Ciências Sociais) – Universidade Estadual de Campinas.

_____. Mothering and Fathering in Flexible and Precarious Working Contexts: The Brazilian IT Sector Case. **International Journal of Gender, Science and Technology**, Milton Keynes, v. 6, n. 1, 2014. Disponível em: <<http://genderandset.open.ac.uk/index.php/genderandset/article/view/288>>. Acesso em: 01 fev. 2016.

CATHO. Cargo de Analista de Sistemas. 2015. Disponível em: <<http://www.catho.com.br/profissoes/analista-de-sistemas/#o-que-faz-um-analista-de-sistemas>>. Acesso em: 30 out. 2015.

CENSO Demográfico 2010: Resultados Preliminares da Amostra. Rio de Janeiro: IBGE, 2011. Disponível em: <http://www.ibge.gov.br/home/estatistica/populacao/censo2010/resultados_preliminares_amostra/notas_resultados_preliminares_amostra.pdf>. Acesso em: 01 out. 2015.

COLEMAN, Gabriella. **Coding freedom**: the ethics and aesthetics of hacking. Princeton (NJ): Princeton University Press, 2013. Disponível em: <<https://gabriellacoleman.org/Coleman-Coding-Freedom.pdf>>. Acesso em: 29 jun. 2024.

COLLIER, David. El método comparativo: dos décadas de cambios. In: SARTORI, Giovanni; MORLINO, Leonardo (Comp.). **La comparación en las ciencias sociales**. Madrid: Alianza Editorial, 2004. p. 51-79.

CUCHE, Denys. A noção de “cultura de empresa”. In: _____. **A noção de cultura nas ciências sociais**. 2. ed. Bauru: EDUSC, 2002. p. 209-224.

DAUNE-RICHARD, Anne-Marie. Qualificações e representações sociais. In: MARUANI, Margaret; HIRATA, Helena (Org.). **As novas fronteiras da desigualdade**: homens e mulheres no mercado do trabalho. São Paulo: SENAC, 2003, p. 65-76.

DUBAR, Claude. **A socialização, construção das identidades sociais e profissionais**. São Paulo: Martins Fontes, 2005.

DUNTEMAN, George H. **Principal Components Analysis**. Sage University Papers Series on Quantitative Applications in the Social Sciences, 07-69. Newbury Park (CA): Sage, 1989.

EISCHEN, Kyle. **Information Technology**: History, Practice and Implications for Development. Working Papers, Center for Global, International and Regional Studies. Santa Cruz: University of California, 2000. Disponível em: <<https://escholarship.org/uc/item/951709tx>>. Acesso em: 18 abr. 2016.

- FIELD, Andy; MILES, Jeremy; FIELD, Zoë. Exploratory factor analysis. In: _____. **Discovering Statistics using R**. 1st ed. London: SAGE, 2012. p. 1888-2043.
- FLICK, Uwe. O processo da pesquisa qualitativa. In: _____. **Introdução à pesquisa qualitativa**. 3. ed. Porto Alegre: Atmed/Bookman, 2009a. p. 95-101.
- _____. Pesquisa qualitativa e quantitativa. In: _____. **Introdução à pesquisa qualitativa**. 3. ed. Porto Alegre: Atmed/Bookman, 2009b. p. 39-49.
- FONTAINE, Annie. **La culture du travail de rue**: une construction quotidienne. Montréal, 2011. Thèse (Phd en service social) – Université de Montréal.
- FREIDSON, Eliot. **Professionalism, the Third Logic**: on the Practice of Knowledge. Chicago: The University of Chicago Press, 2001.
- FSF, Free Software Foundation. O Que é “Esquerdo de Cópia” (Copyleft)? In: **O Sistema Operacional GNU**. 03 mar. 2015. Disponível em: <<https://web.archive.org/web/20160416102054/http://www.gnu.org/licenses/copyleft.pt-br.html>>. Acesso em: 02 maio 2016.
- GOUNET, Thomas. **Fordismo e toyotismo**: na civilização do automóvel. São Paulo: Boitempo, 2002.
- HACKER. In: **DICIO**, Dicionário online de português. Porto: 7Graus, 2016. Disponível em: <<http://www.dicio.com.br/hacker>>. Acesso em: 10 jun. 2016.
- HARAWAY, Donna. Saberes localizados: a questão da ciência para o feminismo e o privilégio da perspectiva parcial. **Cadernos Pagu**, Situando diferenças, Campinas (SP), n. 5, p. 7-41, 1995. Disponível em: <<https://ieg.ufsc.br/storage/articles/October2020/31102009-083336haraway.pdf>>. Acesso em: 29 jun. 2024.
- HARVEY, David. **O neoliberalismo**: história e implicações. São Paulo: Loyola, 2008.
- HASENBALG, Carlos. **Discriminação e desigualdades raciais no Brasil**. 2. ed. Belo Horizonte: Editora UFMG; Rio de Janeiro: IUPERJ, 2005.
- HELOANI, Roberto. **Gestão e organização no capitalismo globalizado**: história da manipulação psicológica no mundo do trabalho. São Paulo: Atlas, 2003.
- HIMANEN, Pekka. **A ética dos hackers e o espírito da era da informação**. Rio de Janeiro, Campus, 2001.
- HOUAISS, Antônio; VILLAR, Mauro. **Dicionário Houaiss da Língua Portuguesa**. Rio de Janeiro: Objetiva, 2009.
- IBGE. Sinopse do Censo Demográfico 2010. Rio de Janeiro: IBGE, 2011. Disponível em: <<http://biblioteca.ibge.gov.br/visualizacao/livros/liv49230.pdf>>. Acesso em: 04 jan. 2016.
- _____. Censo Demográfico 2010: Características gerais da população, religião e pessoas com deficiência. Rio de Janeiro: IBGE, 2012. Disponível em:

<http://biblioteca.ibge.gov.br/visualizacao/periodicos/94/cd_2010_religiao_deficiencia.pdf>. Acesso em: 27 jan. 2016.

INFORMANT. Qual a diferença entre engenheiro, programador e desenvolvedor? In: **INFORMANT**: Sonhe alto. 18 mar. 2014. Disponível em: <<https://web.archive.org/web/20150923221625/http://www.informant.com.br/blog/2014/03/18/qual-a-diferenca-entre-engenheiro-programador-e-desenvolvedor>>. Acesso em: 22 jun. 2015.

JODELET, Denise. Representações sociais: um domínio em expansão. In: JODELET, Denise (Org.). **As representações sociais**. Rio de Janeiro: Eduerj, 2002, p.17-44.

KING, Garry; KEOHANE, Robert; VERBA, Sidney. **Designing social inquiry**: scientific inference in qualitative research. New Jersey: Princeton University Press, 1994.

KRZANOWSKI, Wojtek J. Between-Groups Comparison of Principal Components. **Journal of the American Statistical Association**, [S.l.], v. 74, n. 367, p. 703-707, Sep. 1979. Disponível em: <<http://www.jstor.org/stable/2286995>>. Acesso em: 01 jun. 2016.

LEVY, Steven. **Hackers**: Heroes of the Computer Revolution. New York: Dell/Doubleday, 1994.

LI, Yan; TAN, Chuan-Hoo; TEO, Hock-Hai. Leadership characteristics and developers' motivation in open source software development. **Information & Management**, n. 5, v. 49, p. 257-267, 2012. Disponível em: <<http://dx.doi.org/10.1016/j.im.2012.05.005>>. Acesso em: 27 abr. 2015.

LIBRESOFT. **FLOSS 2013 survey** [dataset]. versão "1.0" 02 abr. 2014. Disponível em: <<https://web.archive.org/web/20160316215717/http://floss2013.libresoft.es/results.en.html>>. Acesso em: 29 mar. 2016.

LIEN, Tracey. Why are women leaving the tech industry in droves? **Los Angeles Times**. Los Angeles, Feb. 2015. Seção Business. Disponível em <<http://www.latimes.com/business/la-fi-women-tech-20150222-story.html#page=1>>. Acesso em: 11 mar. 2015.

MACEDO, Maicon Costa Borba. **O Mercado de Trabalho em Tecnologia de Informação**: a inserção profissional dos desenvolvedores de software. Porto Alegre, 2011. Dissertação (Mestrado em Sociologia) – Universidade Federal do Rio Grande do Sul.

MARX, Karl. Capítulo XI: Cooperação. In: _____. **O capital**: crítica da economia política. Livro primeiro. Vol I. 19. ed. Rio de Janeiro: Civilização Brasileira, 2002. p. 373-388.

MAZIÈRES, Antoine Bernard Marie. **Uma análise sociopolítica do movimento do software livre e de código aberto**. Campinas (SP), 2009. Dissertação (Mestrado em Ciência Política) – Universidade Estadual de Campinas.

MELLO, Matheus Guimarães. O trabalho em serviços musicais: gênero, representações de desvalorização e múltiplos perfis de atuação In: NUNES, Jordão Horta (Org.). **A seu dispor!**: sociologia do trabalho em serviços. Goiânia: Editora da PUC Goiás, 2011, p. 131-159.

MERTON, Robert K. The Mathew Effect in Science. **Science**, New Series, v. 159, n. 3810, p. 56-63, Jan. 1968.

MIGNOLO, Walter D. A colonialidade de cabo a rabo: o hemisfério ocidental no horizonte conceitual da modernidade. In: LANDER, Edgardo (Org.). **A colonialidade do saber: eurocentrismo e ciências sociais. Perspectivas latino-americanas.** Buenos Aires: CLACSO, 2005. Disponível em <<http://bibliotecavirtual.clacso.org.ar/ar/libros/lander/pt/Mignolo.rtf>>. Acesso em 12 nov. 2013.

NUNES, Jordão Horta. “A seu dispor!” – identidade e interação no trabalho em serviços. In: _____ (Org.). **A seu dispor!** Sociologia do trabalho em serviços. Goiânia: Ed. da PUC Goiás, 2011. p. 15-48.

_____. Gênero e raça no trabalho em tecnologia da informação (TI). **Ciências Sociais Unisinos**, São Leopoldo, v. 52, n. 3, p. 383-395, 2016. Disponível em: <https://copiarevistas.unisinos.br/index.php/ciencias_sociais/article/view/csu.2016.52.3.09/5758>. Acesso em: 29 jun. 2024.

NUNES, Jordão Horta; MELLO, Matheus Guimarães; SOUZA, Tatiele Pereira de. Reconhecimento e profissionalização no campo da tecnologia da informação. In: CONGRESSO BRASILEIRO DE SOCIOLOGIA, 17., 2015, Porto Alegre. **Programação Completa: GT20 – 2ª Sessão – 23 de julho.** Disponível em: <<http://automacaodeeventos.com.br/sociologia/sis/inscricao/resumos/0001/R1172-1.PDF>>. Acesso em: 24 jun. 2024.

Ó RIAIN, Seán. The missing customer and the ever-present market: software developers and the service economy. **Work and Occupations**, [Nashville (TN)], v. 37, n. 3, p. 320-348, 2010. Disponível em: <<https://doi.org/10.1177/0730888410373331>>. Acesso em: 16 jan. 2015.

ORTIZ, Matilde Laura Velasco. Un acercamiento al método tipológico en sociología. In: TARRÉS, María Luisa (Org.). **Observar, escuchar y comprender: sobre la tradición cualitativa en la investigación social.** México (D.F.): FLACSO, 2008. p. 289-323.

PAGÈS, Max et al. **O poder das organizações: a dominação das multinacionais sobre os indivíduos.** São Paulo: Atlas, 1987.

PPI. Pirate Parties International. **About PPI**, 2015. Disponível em: <<https://pp-international.net/about-ppi>>. Acesso em: 3 jul. 2015.

RAYMOND, Eric. **A catedral e o bazar.** 1998. Disponível em: <<http://www.dominiopublico.gov.br/download/texto/tl000001.pdf>>. Acesso em: 16 out. 2012.

ROLANDSSON, Bertil; BERGQUIST, Magnus; LJUNGBERG, Jan. Open source in the firm: opening up professional practices of software development. **Research Policy**, v. 40, n. 4, p. 576-587, 2011. Disponível em: <<https://doi.org/10.1016/j.respol.2010.11.003>>. Acesso em: 15 out 2012.

ROMNEY, A. Kimball; et al. Statistical methods for characterizing similarities and differences between semantic structures. **PNAS**, [S.l.], v. 97, n. 1, p. 518-523, 2000. Disponível em: <<https://doi.org/10.1073/pnas.97.1.518>>. Acesso em: 01 jun. 2016.

ROSENFELD, Cinara. Autoempendedorismo: forma emergente de inserção social pelo trabalho. **Revista Brasileira de Ciências Sociais**, v. 30, n. 89, p. 115-197, 2015. Disponível em: <<https://doi.org/10.17666/3089115-128/2015>>. Acesso em: 02 dez. 2015.

ROSENFELD, Cinara; MOSSI, Thays. O que é ser flexível na nova configuração do trabalho? os quadros superiores de TI e a flexibilidade na gestão da empresa, na organização do trabalho e nas relações de trabalho. **Revista ABET**, João Pessoa, v. 10, n. 2, p. 80-95, 2011. Disponível em: <<https://periodicos.ufpb.br/index.php/abet/article/view/15601/8915>>. Acesso em: 15 out. 2012.

SÁNCHEZ, Christian Muñoz. **Cultura organizacional e inovação em empresas mexicanas desarrolladoras de software libre**. México (DF), 2010. Tesis (Maestría en Ciencias en Administración) – Instituto Politécnico Nacional.

SAUTU, Ruth; BONIOLO, Paula; DALLE, Pablo; ELBERT, Rodolfo. La construcción del marco teórico en la investigación social. In: _____. **Manual de metodología**. Construcción del marco teórico, formulación de los objetivos y elección de la metodología. Buenos Aires: CLACSO, 2005. p. 29-81. Disponível em: <<https://biblioteca-repositorio.clacso.edu.ar/handle/CLACSO/14733>>. Acesso em: 28 maio 2013.

SEBRAE (Org.). Anuário do trabalho na micro e pequena empresa: 2013. 6. ed. Serviço Brasileiro de Apoio às Micro e Pequenas Empresas; Departamento Intersindical de Estatística e Estudos Socioeconômicos – Brasília, DF; DIEESE, 2013.

SILVEIRA, Sérgio Amadeu da. Ciberativismo, cultura hacker e o individualismo colaborativo. **Revista USP**, São Paulo, n. 86, p. 28-39, 2010. Disponível em: <<https://www.revistas.usp.br/revusp/article/view/13811/15629>>. Acesso em: 15 out. 2012.

STALLMAN, Richard. **Free software, free society**. 2nd ed. Boston: Free Software Foundation, 2010. Disponível em: <<https://www.gnu.org/doc/fsfs-ii-2.pdf>>. Acesso em: 3 ago. 2014.

STEWART, Daniel. Social Status in an Open-Source Community. **American Sociological Review**, v. 70, n. 5, p. 823-842, Oct. 2005. Disponível em: <<http://www.jstor.org/stable/4145361>>. Acesso em: 15 mar. 2013.

STRAUSS, Anselm; CORBIN, Juliet. **Pesquisa qualitativa: técnicas e procedimentos para o desenvolvimento de teoria fundamentada**. Porto Alegre: Artmed/Bookman, 2008.

TELLES, Edward. Repensando as relações de raça no Brasil. **Teoria e Pesquisa**, São Carlos, n. 42 e 43, p. 131-159, 2003. Disponível em: <<http://www.teoriaepesquisa.ufscar.br/index.php/tp/article/viewFile/59/49>>. Acesso em: 01 jan. 2016

THE MENTOR. The Conscience of a Hacker / The Hacker Manifesto. **Phrack**, v. 1, n. 7, 1986. Disponível em: <<http://phrack.org/issues/7/3.html#article>>. Acesso em: 27 abr. 2015.

THOMAS, Douglas. **Hacker Culture**. Minneapolis: University of Minnesota Press, 2002.

TOLEDO, Enrique de la Garza *et al.* Para um conceito ampliado de trabalho, de controle, de regulação e de construção social da ocupação: os “outros trabalhos”. In: LEITE, Marcia de Paula; ARAÚJO, Angela Maria Carneiro. **O trabalho reconfigurado: ensaios sobre Brasil e México**. São Paulo: Annablume, 2009. p. 123-147.

TORI, Carlos. **Hacking Ético**. Rosario, 2008.

TREMBLAY, Diane-Gabrielle; GENIN, Émilie. Remodelage des temps et des espaces de travail chez les travailleurs indépendants de l'informatique: l'affrontement des effets de marchés et des préférences personnelles. **Temporalités**, n. 10, 2009. Disponível em: <<https://doi.org/10.4000/temporalites.1111>>. Acesso em: 28 jan. 2015.

WALLERSTEIN, Immanuel. **World-systems analysis: an introduction**. Durham: Duke University Press, 2006.

_____. Mudando a geopolítica do sistema-mundo: 1945-2025. In: MARTINS, Carlos Eduardo; VALENCIA, Adrián Sotelo (Org.). **A América Latina e os desafios da globalização**: ensaios em homenagem a Ruy Mauro Marini. Rio de Janeiro, São Paulo: Editora PUC Rio, Boitempo Editorial, 2009.

WARK, McKenzie. **A Hacker Manifesto**. Cambridge (MA): Harvard University Press, 2004.

WEBER, Max. **A "objetividade" do conhecimento nas ciências sociais**. Tradução, apresentação e comentários de Gabriel Cohn. São Paulo: Ática, 2006.

APÊNDICE A – Roteiro de Entrevista



Universidade Federal de Goiás
 Faculdade de Ciências Sociais
**PESQUISA: CULTURA DO TRABALHO
 DE DESENVOLVEDORES DE SOFTWARE LIVRE**



Esta entrevista é parte de minha pesquisa em andamento, que ao final redundará em uma dissertação de mestrado, através do Programa de Pós-Graduação em Sociologia da UFG, que possui um intuito estritamente científico: compreender e analisar a cultura do trabalho de desenvolvedores de *software* livre brasileiros. Agradecemos sua colaboração e garantimos que sua identidade será resguardada e só serão divulgadas informações relacionadas a nomes fictícios. Os resultados da pesquisa serão divulgados somente em eventos e publicações científicas.

Pesquisador: Matheus Guimarães Mello

Orientador: Prof. Jordão Horta Nunes

Estímulo à memória

* Qual a primeira coisa que vem à sua cabeça quando pensa no seu trabalho como desenvolvedor?

* O que é software livre?

→ para você há diferenças entre software livre e código aberto? Quais? Por quê?

(perguntar onde trabalha, com quem trabalha... para depois narrar o dia típico)

Narrativa Episódica

* Você poderia narrar seu dia típico levando principalmente em conta seu trabalho como desenvolvedor? (o que acontece no dia a dia?)

* E também uma semana típica de trabalho? (o que acontece de um dia para outro?)

Trajatória Pessoal / Profissional

* Por que você decidiu estudar computação?

* Conte como foi sua formação na área da computação.

* Dentre o trabalho de desenvolvimento de software livre, quais sub-atividades são possíveis se especializar?

→ alguma delas seria mais central na definição do desenvolvimento de software?

* Por que você se especializou na sua atual área? (como chegou a exercer essa atividade?)

* Dentre todos os tipos de atividades e competências do seu trabalho, com quais você se sente mais realizado?

- * Você gostaria de estar envolvido em outro tipo de atividade na computação? Por quê?
- * O que você pretende fazer nos próximos anos? Quais expectativas ou planos pessoais e profissionais você têm quanto ao futuro?

Identidade Profissional / Ocupacional

- * O que é ser desenvolvedor de software livre?
 - quais as características de alguém que trabalha com software livre?
- * O que distingue um desenvolvedor de software livre de um desenvolvedor de software proprietário?
 - o que distingue um desenvolvedor de software de outros profissionais de TI?
- * Você acha que um diploma de ensino superior ou certificações de cursos em geral são um requisitos para trabalhar na sua área?
 - se sim, quais são mais valorizados?
 - se não, quais competências são necessárias?
- * O que é um hacker?
 - você se considera um hacker?
- * Dentre todas suas identidades (filho, desenvolvedor, brasileiro, de tal religião, etc), qual (ou quais) você mais se identifica? (qual a primeira coisa que lembra quando diz eu sou isso)
- * Como os outros te identificam? (como desenvolvedor, hacker, ativista, trabalhador...)
 - você se sente confortável ou desconfortável com alguma dessas identificações?
- * O que você mais gosta e o que menos gosta em seu trabalho?
- * O que mais motiva seu envolvimento no desenvolvimento de software? (a paixão pelo que faz ou a possibilidade de bons retornos financeiros?)
- * Quais acontecimentos recentes lhe motivou mais a se dedicar bem no trabalho?
 - e quais lhe desmotivou/frustrou?
- * Se você não fosse um desenvolvedor, o que seria ou gostaria de ser?
- * Acredita que seu trabalho é valorizado socialmente?
 - se sim, diga quais
 - se não, de que formas seu trabalho poderia ser mais valorizado?

Condições de Trabalho

- * Você tem mais de um emprego? (de que forma você compõe seu rendimento mensal?)
- * Você considera seu trabalho estável?
- * Você considera sua carga de trabalho pesada?
 - acha que seria necessária uma redução do tempo de trabalho para melhorar sua qualidade de vida?

- * Acha que é necessário para um desenvolvedor exercer várias atividades simultaneamente?
→ quais os pontos positivos e negativos disso?
- * Você trabalha ou já trabalhou como freelancer como sua principal atividade?
- * E em trabalhos fixos, é comum se envolver mais com projetos flexíveis?
- * Descreva a hierarquia dos cargos no seu trabalho.
- * Como funciona a cooperação entre seus colegas de trabalho?

Cultura do Trabalho

- * De quais comunidades de desenvolvimento de software livre já participou?
- * Quais as principais diferenças você percebe entre as comunidades?
→ sobre essas diferenças, por que você acha elas acontecem?
→ no que mais elas podem interferir?
- * Você já presenciou diferenças de ideologia ou posicionamento político influenciando as escolhas técnicas?
- * Quais as diferenças entre um profissional mais experiente e um iniciante na área?
- * Você acredita que a ideia de disponibilizar o código aberto influencia a maneira como o trabalho se organiza em sua empresa/fundação? (como a dinâmica do seu trabalho seria diferente se você trabalhasse apenas com software proprietário?)

Gênero

- * Você acha que o fato de você ser homem/mulher influencia no dia a dia da execução de sua atividade?
- * Para você, por que há mais homens do que mulheres nas áreas da computação?
- * É comum algum tipo de discriminação, seja por diferença de salário, seja por tratamento diferente no dia a dia?

Tempo

- * Como você costuma organizar seu tempo? Quais atividades têm prioridade?
- * Quanto tempo do seu dia você dedica à computação? Fale sobre esse envolvimento que você tem.
- * O que costuma fazer quando não está trabalhando?
- * Seu círculo de amizades envolve principalmente pessoas da computação?
- * O que gostaria de gastar mais tempo fazendo?

Obrigado por contribuir como informante nesta pesquisa.

Gostaria de participar do questionário on-line da próxima etapa da pesquisa? [S/N]

e-mail: _____

Poderia me recomendar alguém para ser entrevistado também?

APÊNDICE B – Estrutura do Questionário

Este questionário é parte de uma pesquisa de mestrado em sociologia, intitulada “Cultura do trabalho de desenvolvedores de software livre”. Seu intuito é estritamente científico, com o seguinte objetivo principal: compreender os processos de socialização e construção de identidades ocupacionais de desenvolvedores de software e sua relação com a cultura e organização do trabalho.

- O questionário é destinado a brasileiro(a)s que trabalham profissionalmente com desenvolvimento de software (livre ou proprietário; gratuito ou comercial), sejam desenvolvedores(as), programadores(as) ou engenheiros(as) de software.
- Sua participação é voluntária, isto é, a qualquer momento você pode recusar-se a responder as perguntas não-obrigatórias ou mesmo desistir de participar, clicando em “sair e apagar o questionário”.
- Desde já agradecemos sua colaboração e garantimos que sua identidade será resguardada, e que as eventuais divulgações de resultados da pesquisa serão limitadas a eventos e publicações científicas.
- Tanto os resultados finais quanto o banco de dados despersonalizado estarão disponíveis a todos(as) colaboradores(as) e publicados sob uma licença aberta para adaptação de outros(as) pesquisadores(as).

Para mais informações, inclusive sobre privacidade, metodologia e objetivos da pesquisa, por favor entre em contato com Matheus Guimarães Mello matheusmello@inventati.org

SITUAÇÃO PESSOAL

1 [sexo] Sexo:

Feminino
Masculino

2 [ufNatal] Estado em que nasceu:

Rondônia
Acre
Amazonas
Roraima
Pará
Amapá
Tocantins
Maranhão
Piauí
Ceará
Rio Grande do Norte
Paraíba
Pernambuco
Alagoas

Sergipe
Bahia
Minas Gerais
Espírito Santo
Rio de Janeiro
São Paulo
Paraná
Santa Catarina
Rio Grande do Sul
Mato Grosso do Sul
Mato Grosso
Goiás
Distrito Federal
_Outro País

3 [naturalizado] Você é brasileiro(a) naturalizado(a)? *

Só responder essa pergunta sob as seguintes condições: A resposta foi '_Outro País' na questão '2 [ufNatal]' (Estado em que nasceu:)

Sim
Não
Ainda não, mas entrei com processo de naturalização

4 [ufReside] Estado em que reside atualmente:

Rondônia
Acre
Amazonas
Roraima
Pará
Amapá
Tocantins
Maranhão
Piauí
Ceará
Rio Grande do Norte
Paraíba
Pernambuco
Alagoas
Sergipe
Bahia
Minas Gerais
Espírito Santo
Rio de Janeiro
São Paulo
Paraná
Santa Catarina
Rio Grande do Sul
Mato Grosso do Sul
Mato Grosso
Goiás
Distrito Federal
_Outro País

5 [idade] Qual sua idade (em anos completos)? *

Por favor, coloque sua resposta aqui:

6 [civil] Qual seu estado civil?

Casado(a)
Desquitado(a) ou separado(a) judicialmente
Divorciado(a)
Viúvo(a)
Solteiro(a)

7 [religiao] Qual a sua religião?

Acredito em Deus, mas não tenho uma religião
Ateísmo ou Agnosticismo
Católica
Protestante Histórica: Luterana, Presbiteriana, Metodista, Batista, Anglicana, Congregacional ou outras
Evangélica Pentecostal: Assembleia de Deus, Congregação Cristã no Brasil, Deus é Amor, Evangelho Quadrangular ou outras
Evangélica Neopentecostal: Universal do Reino de Deus, Igreja Internacional da Graça de Deus, Fonte da Vida ou outras
Adventista
Igreja de Jesus Cristo dos Santos dos Últimos Dias
Testemunha de Jeová
Outra Religião Cristã
Espírita, Espiritualista ou Kardecista
Umbanda, Candomblé ou outra Religiosidade Afro-Brasileira
Judaica
Muçulmana
Hinduísta
Budista
Outra Religião Oriental
Esotérica
Tradição Indígena
Pertencço a várias religiões
Outra

8 [cor] Qual sua cor ou raça?

Branca
Amarela (origem oriental)
Parda
Preta
Indígena

FAMÍLIA

9 [conjuge] Você vive em companhia de cônjuge ou companheiro(a)? *

Sim
Não, mas já vivi antes
Não e nunca vivi

10 [conjugueOcupacaoTexto] Qual a ocupação ou profissão do(a) cônjuge ou companheiro(a)?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Sim' na questão '9 [conjugue]' (Você vive em companhia de cônjuge ou companheiro(a)?)

Por favor, coloque sua resposta aqui:

11 [filhos] Quantos filhos ou filhas você tem? *

Por favor, coloque sua resposta aqui:

DICA: Se não tem filhos ou filhas, digite 0.

12 [filhosidade] Quantos anos tem seu filho ou sua filha mais novo(a)?

Só responder essa pergunta sob as seguintes condições: A resposta foi maior ou igual a '1' na questão '11 [filhos]' (Quantos filhos ou filhas você tem?)

Por favor, coloque sua resposta aqui:

DICA: Se possuir menos de um ano, basta preencher com um decimal aproximado. Por exemplo: 6 meses = 0,5 / 3 meses = 0,25

13 [moradores] Quantas pessoas, incluindo você, moram na sua casa? *

Por favor, coloque sua resposta aqui:

DICA: Não se esqueça de incluir todo(a)s moradore(a)s, incluindo crianças e agregado(a)s. No caso de morar sozinho(a), digite 1.

14 [moraPais] Você mora com pai e/ou mãe?

Só responder essa pergunta sob as seguintes condições: A resposta foi maior ou igual a '2' na questão '13 [moradores]' (Quantas pessoas, incluindo você, moram na sua casa?)

Sim
Não

ESCOLARIDADE E FORMAÇÃO**15 [escolaridade] Qual a sua escolaridade? ***

Sem Instrução Formal ou Fundamental Incompleto
Fundamental Completo
Médio Completo
Superior Completo
Pós-Graduação

16 [estuda] Você está frequentando algum curso (em qualquer um dos níveis de escolaridade)?

Sim
Não

17 [estudaNivel] Qual o nível do curso que está frequentando atualmente?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Sim' na questão '16 [estuda]' (Você está frequentando algum curso (em qualquer um dos níveis de escolaridade)?)

Regular do Ensino Fundamental
Educação de Jovens e Adultos ou Supletivo do Ensino Fundamental
Regular do Ensino Médio
Educação de Jovens e Adultos ou Supletivo do Ensino Médio
Técnico
Superior de Graduação
Especialização de Nível Superior (mínimo de 360 horas)
Mestrado
Doutorado
Outro

18 [estudaTecnicoTexto] Qual o nome do curso técnico sendo cursado?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Técnico' na questão '17 [estudaNivel]' (Qual o nível do curso que está frequentando atualmente?)

Por favor, coloque sua resposta aqui:

19 [cursadoSuperiorTexto] Qual o nome do curso superior já cursado?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Superior Completo' ou 'Pós-Graduação' na questão '15 [escolaridade]' (Qual a sua escolaridade?)

Por favor, coloque sua resposta aqui:

20 [estudaSuperior] Qual o nome do curso superior sendo cursado?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Superior de Graduação' na questão '17 [estudaNivel]' (Qual o nível do curso que está frequentando atualmente?)

Por favor, coloque sua resposta aqui:

21 [tecnico] Você já concluiu algum curso técnico ou profissionalizante?

Sim
Não

22 [cursadoTecnicoTexto] Qual o nome do curso realizado?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Sim' na questão '21 [tecnico]' (Você já concluiu algum curso técnico ou profissionalizante?)

Por favor, coloque sua resposta aqui:

23 [socializa] Quais pessoas ou instituições possibilitaram seus contatos iniciais com a computação? *

Por favor, escolha as opções que se aplicam:

Família
Amigos
Escola
Faculdade
Empresa
Igreja
Vizinhos
Você mesmo(a)
Outra:

24 [escolaridadeMae] Qual a escolaridade de sua mãe?

Não Sei
Sem Instrução Formal ou Fundamental Incompleto
Fundamental Completo
Médio Completo
Superior Completo
Pós-Graduação

25 [escolaridadePai] Qual a escolaridade de seu pai?

Não Sei
Sem Instrução Formal ou Fundamental Incompleto
Fundamental Completo
Médio Completo
Superior Completo
Pós-Graduação

26 [formacao] Avalie a relevância dos seguintes itens para sua formação profissional:

Por favor, escolha a resposta adequada para cada item:

Nada importante Pouco Importante Razoavelmente Importante Muito Importante Extremamente Importante

Curso de Ensino Superior
Curso Técnico ou Profissionalizante
Cursos de Certificação
Cursos oferecidos pela empresa
Aprendizado por meio de fóruns especializados, internet e outros meios disponíveis
Aprendizado próprio, por tentativa e erro

DICA: Se não tiver realizado a formação de algum tópico, considere como "nada importante".

DESENVOLVIMENTO DE SOFTWARE

27 [atividade] Em qual dessas atividades de desenvolvimento de software você trabalha predominantemente?

*

Liderança de projeto
Análise de sistemas
Análise de requisitos
Design do software (ou projeto)
Implementação (ou programação)
Teste
Documentação
Implantação (ou instalação)
Suporte e treinamento
Depuração (ou correção de erros)
Manutenção
Gerência de configuração (p. ex. controle de versão)
Meu trabalho perpassa todas ou quase todas as atividades
Outros

28 [metodo] No processo de desenvolvimento de software há diversos modelos e formas de organizar o trabalho. Quais dos modelos e métodos abaixo se aproximam do seu cotidiano de trabalho:

Modelo em cascata: o processo de desenvolvimento ocorre a partir de fases separadas em que há um processo de especificação completa de requisitos, projeto, construção e teste de sistema.

Desenvolvimento iterativo: intercala as atividades de especificação, desenvolvimento e validação.

Ambos são utilizados e a inserção em um desses modelos depende do projeto no qual estou inserido(a).

Outro tipo:

29 [linguagem] Quais ferramentas e linguagens de programação você utiliza com mais frequência em seu trabalho?

Por favor, escolha as opções que se aplicam:

Assembly
C
C++
C#
Objective C
Java
Python
Perl
PHP
Unix Shell
HTML
LISP
(La)TeX
Pascal
Fortran
Basic
Visual Basic
JavaScript
SQL
XML
Ruby

Smalltalk
Tcl
Make
Scheme
Lua
Hern
Brainfuck
R
Outra:

30 [licença] Qual é a licença predominantemente escolhida para os programas que você desenvolve? *

Licença comercial ou proprietária (p.ex: contrato do tipo End-User License Agreement - EULA)
Licença de código aberto permissiva (p.ex: Licença MIT ou X11, Licença BSD, Licença Apache, Licença Artística)
Outros tipos de licença aberta com copyleft parcial (p.ex. Licença Pública Mozilla, Licença PHP)
Licença com copyleft forte (GPL - GNU General Public Licence)
Outros tipos de licenças livres/abertas (p.ex. Domínio Público)
Sem licença
Múltiplo licenciamento de código aberto/software livre
Múltiplo licenciamento de código aberto/proprietário
Não sei qual é o tipo de licença ou não sei como escolher o licenciamento

**31 [area] Qual a área funcional do programa que você está desenvolvendo no momento?
Por favor, escolha as opções que se aplicam:**

Áudio / Vídeo
Comunicação / Social
Educação
Científico
Escritório / Empresarial
Desenvolvimento
Gráficos
Segurança
Sistema / Utilitário
Jogos
Outra:

SOFTWARE LIVRE

32 [proprietario] Desenvolve ou está envolvido(a) em projeto de software proprietário (código fechado)? *

Sim
Não
Não agora, mas já estive envolvido(a)

33 [proprietariolnicio] Em que ano começou seu trabalho no desenvolvimento de software proprietário (código fechado)?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Sim' ou 'Não agora, mas já estive envolvido(a)' na questão '32 [proprietario] ' (Desenvolve ou está envolvido(a) em projeto de software proprietário (código fechado)?)

Por favor, coloque sua resposta aqui:

34 [proprietarioFim] Em que ano terminou seu envolvimento com software proprietário?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Não agora, mas já estive envolvido(a)' na questão '32 [proprietario]' (Desenvolve ou está envolvido(a) em projeto de software proprietário (código fechado)?)

Por favor, coloque sua resposta aqui:

35 [livre] Desenvolve ou está envolvido(a) em projeto de software livre / software de código aberto? *

Sim
Não
Não agora, mas já estive envolvido(a)

36 [livreInicio] Em que ano começou seu trabalho no desenvolvimento de software livre?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Sim' ou 'Não agora, mas já estive envolvido(a)' na questão '35 [livre]' (Desenvolve ou está envolvido(a) em projeto de software livre / software de código aberto?)

Por favor, coloque sua resposta aqui:

37 [livreFim] Em que ano terminou seu envolvimento com software livre?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Não agora, mas já estive envolvido(a)' na questão '35 [livre]' (Desenvolve ou está envolvido(a) em projeto de software livre / software de código aberto?)

Por favor, coloque sua resposta aqui:

38 [livreIdentidade] Você se considera parte da comunidade de software livre ou da comunidade de código aberto?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Sim' ou 'Não agora, mas já estive envolvido(a)' na questão '35 [livre]' (Desenvolve ou está envolvido(a) em projeto de software livre / software de código aberto?)

Eu me vejo como parte da comunidade de software livre.
Eu me vejo como parte da comunidade de código aberto.
Utilizo software livre/aberto, mas não me importo com a distinção.

39 [ativismo] Qual o seu nível de ativismo dentro do movimento software livre?

Não me interessa pelo software livre..
Tenho contato eventual com software livre, mas não sou ativista.
Apenas divulgo o software livre em meus círculos sociais, sou mais um(a) usuário(a) do que um(a) ativista.
Fui um(a) ativista no passado, mas atualmente não sou mais.
Participo de listas e grupos de discussão, promovo palestras e ajudo a organizar eventos.
Sou um(a) ativista engajado(a), pois sou membro de um projeto, além de atividades como organizar eventos, escrever sobre software livre em sites, revistas ou blogs, etc.

40 [hackerCultura] Você se considera de alguma forma parte da cultura hacker? *

Sim
Não

41 [hackerIdentidade] Você se considera um(a) hacker? *

Sim
Não

42 [hackerDefinicaoTexto] Qual(is) a(s) característica(s) principal(is) que define(m) um(a) hacker?

Por favor, coloque sua resposta aqui:

TEMPO

43 [vinculos] Quantos vínculos de trabalho você tem atualmente? *

Nenhum
Um
Dois ou mais

DICA: Isso inclui outros empregos, trabalhos temporários, trabalhos informais e prestação de serviço, tais como freelancer, que não se relacionam com seu trabalho principal.

44 [vinculosDesenv] Todos os seus vínculos de trabalho atuais se relacionam diretamente com desenvolvimento de software? *

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Dois ou mais' na questão '43 [vinculos] ' (Quantos vínculos de trabalho você tem atualmente?)

Sim
Não

45 [vinculosComplemTexto] Além de seu trabalho como desenvolvedor(a), qual(is) seu(s) outro(s) trabalho(s)?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Não' na questão '44 [vinculosDesenv] ' (Todos os seus vínculos de trabalho atuais se relacionam diretamente com desenvolvimento de software?)

Por favor, coloque sua resposta aqui:

46 [horasDesenv] Em seu trabalho como desenvolvedor(a), quantas horas você trabalha habitualmente por semana? *

Por favor, coloque sua resposta aqui:

47 [tempoSoftware] Quanto do seu tempo de trabalho semanal você habitualmente dedica ao software livre (em relação ao tempo que dedica ao software proprietário)?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Sim' na questão '32 [proprietario] ' (Desenvolve ou está envolvido(a) em projeto de software proprietário (código fechado)?) e A resposta foi 'Sim' na questão '35 [livre] ' (Desenvolve ou está envolvido(a) em projeto de software livre / software de código aberto?)

A maior parte do tempo me dedico ao software livre
Dedico aproximadamente um tempo igual ao software livre e ao software proprietário
A maior parte do tempo me dedico ao software proprietário

48 [horasComplem] Quantas horas por semana você trabalha habitualmente em seu(s) outro(s) trabalho(s)? *

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Não' na questão '44 [vinculosDesenv] ' (Todos os seus vínculos de trabalho atuais se relacionam diretamente com desenvolvimento de software?)

Por favor, coloque sua resposta aqui:

49 [horasVoluntar] Quantas horas por semana em média se dedica a trabalho não remunerado relacionado a outros projetos, grupos ou comunidades?

Por favor, coloque sua resposta aqui:

DICA: Participação em listas de discussão, apresentação e organização em eventos de TI, divulgação, suporte ou desenvolvimento de software livre, etc.

50 [lazer] Como você costuma preencher seu tempo fora do trabalho?

Por favor, escolha as opções que se aplicam:

Com a família
Realizando projetos de trabalho
Com amigo(a)s presencialmente
Com amigo(a)s através de comunicação virtual
Com namorado(a) ou esposo(a)
Em atividades religiosas
Em atividades físicas ou esportivas
Em cursos de certificação ou relacionados à tecnologia
Em atividades culturais (show de música, cinema, teatro)
Assistindo TV
Jogando videogame
Em leitura
Surfando na Internet
Descansando sem fazer uma atividade específica
Outros:

CARACTERÍSTICAS GERAIS DO TRABALHO

NOTA: A partir de agora, todas as questões se referem apenas a seu trabalho com desenvolvimento de software

51 [localTrabalho] Onde é seu local de trabalho? *

Se você escolher 'Outro lugar:' por favor especifique a sua escolha no campo de texto.

Apenas na empresa
Apenas no próprio domicílio
Parcialmente na empresa e parcialmente no próprio domicílio
Na instituição do cliente
Parcialmente na instituição do cliente e parcialmente no próprio domicílio
Parcialmente na empresa e parcialmente na instituição do cliente
Em local alugado e compartilhado para trabalho (coworking)
Outro lugar:

52 [deslocaMunicípio] Você se desloca diariamente para outro município para trabalhar?

Sim
Não

53 [empresas] Em quantas empresas você já trabalhou e não trabalha mais?

Por favor, coloque sua resposta aqui:

54 [posicaoOcupacao] Qual a sua posição ocupacional em seu trabalho como desenvolvedor(a)? *

Empregado(a) com carteira de trabalho assinada
Empregado(a) pelo regime jurídico de funcionário(a)s público(a)s
Empregado(a) sem carteira de trabalho assinada
Conta própria
Empregador(a)
Não remunerado(a)

55 [focoEmpresa] Qual o foco da empresa em que você trabalha?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Empregado(a) pelo regime jurídico de funcionário(a)s público(a)s' ou 'Empregado(a) com carteira de trabalho assinada' ou 'Empregado(a) sem carteira de trabalho assinada' ou 'Empregador(a)' na questão '54 [posicaoOcupacao]' (Qual a sua posição ocupacional em seu trabalho como desenvolvedor(a)?)

Empresa especializada em um único software
Empresa com foco em diferentes tipos de software
Empresa com foco na tecnologia da informação em geral
Empresa com foco em outra área, mas com setor de tecnologia da informação
Empresa pública com foco em oferecer software ao governo
Outra:

56 [autonomoCnpj] Seu trabalho por conta própria possui CNPJ?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Conta própria' na questão '54 [posicaoOcupacao]' (Qual a sua posição ocupacional em seu trabalho como desenvolvedor(a)?)

Sim
Não

57 [funcionarios] Quanto(a)s funcionário(a)s estão empregado(a)s na empresa em que você trabalha?

Apenas eu mesmo Apenas eu mesmo
2 a 9 pessoas 2 a 9 pessoas
10 a 49 pessoas 10 a 49 pessoas
50 a 99 pessoas 50 a 99 pessoas
100 ou mais pessoas 100 ou mais pessoas

DICA: Caso trabalhe por conta própria, marque "Apenas eu mesmo".

58 [cliente] Você interage diretamente com o cliente ou usuário final do software durante o processo de desenvolvimento?

Sim
Não

59 [chefe] Seu chefe ou gerente trabalha no mesmo espaço que você, ou se comunicam apenas virtualmente?

Nós trabalhamos no mesmo espaço
Nós nos comunicamos apenas virtualmente
Não tenho chefe nem gerente

60 [pressao] De quem você considera sofrer pressão para entregar resultados com rapidez? *

Por favor, escolha as opções que se aplicam:

Chefe ou gerente
Líder da equipe
Colegas
Consumidor/Usuário
Contribuidores de código externos ao projeto
Outra pessoa:

61 [rendaSoftwareLivre] O quanto de sua renda pessoal é obtida através de seu trabalho com software livre?

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Sim' na questão '32 [proprietario]' (Desenvolve ou está envolvido(a) em projeto de software proprietário (código fechado)?) e A resposta foi 'Sim' na questão '35 [livre]' (Desenvolve ou está envolvido(a) em projeto de software livre / software de código aberto?)

Nenhuma ou bem pouca parte
Menos da metade da minha renda
Metade da minha renda
Mais da metade da minha renda
Toda ou quase toda minha renda

62 [rendaComplem] O quanto de sua renda pessoal é obtida através de seu(s) trabalho(s) não relacionado(s) com desenvolvimento de software

Só responder essa pergunta sob as seguintes condições: A resposta foi 'Dois ou mais' na questão '43 [vinculos] ' (Quantos vínculos de trabalho você tem atualmente?)

- Nenhuma ou bem pouca parte
- Menos da metade da minha renda
- Metade da minha renda
- Mais da metade da minha renda
- Toda ou quase toda minha renda

63 [rendaDomicilio] Qual o rendimento bruto total de seu domicílio (a soma aproximada dos rendimentos de todas as pessoas que moram com você), em reais?

Por favor, coloque sua resposta aqui:

VALORES E CULTURA DO TRABALHO

64 [satisfacao] Como você se considera em relação a seu vínculo de trabalho atual?

- É o que sempre quis fazer, eu realmente amo meu trabalho atual.
- É interessante, mas posso imaginar ter outros vínculos de trabalho interessantes também.
- Eu apenas sou bem pago, faço este trabalho principalmente pelo dinheiro.
- Não gosto muito do trabalho atual, se houver uma oportunidade, irei trocar.
- Eu realmente detesto meu trabalho atual.

65 [motivo] Avalie as seguintes motivações para realizar seu trabalho no desenvolvimento de software, de acordo com a importância em sua vida:

Por favor, escolha a resposta adequada para cada item:

Concordo Totalmente Concordo Parcialmente Não concordo

- Realização de um sonho: sempre desejei trabalhar na área de tecnologia
- Ascensão econômica e profissional
- Contribuir com o desenvolvimento da sociedade
- Aprender e desenvolver novas habilidades
- Ganhar reputação entre outros programadores
- Diminuir o poder de grandes companhias de software
- Resolver um problema que não poderia ser resolvido por outro software existente
- Compartilhar meus conhecimentos e habilidades
- Aumentar minhas oportunidades de emprego
- Esta é uma variável de controle, marque "concordo parcialmente"!
- Vender produtos ou serviços
- Orgulho em fazer parte de uma comunidade
- Obrigação por uma circunstância profissional
- Diversão em programar

66 [trabalho] Avalie as afirmações em relação a seu ambiente de trabalho a partir da escala:

Por favor, escolha a resposta adequada para cada item:

Concordo Totalmente Concordo Parcialmente Não concordo Não se aplica ao meu caso

Predomina um espírito de trabalho em equipe entre meus colegas.
 A equipe de desenvolvimento possui liberdade para fazer seu trabalho da forma como prefere.
 As decisões sobre a execução do projeto são de responsabilidade de toda a equipe.
 Os horários de trabalho são estabelecidos conforme a necessidade da empresa ou dos clientes.
 A maneira como se deve realizar o trabalho constantemente é mudada.
 Os colegas de trabalho comentam seus problemas pessoais uns com os outros.
 Há um ambiente agradável de trabalho.
 Os funcionários trabalham bem, mesmo na ausência do chefe ou gerente.
 A organização coletiva do trabalho é eficiente.
 O trabalho frequentemente é entediante.
 Frequentemente há pressão para entregar resultados em pouco tempo.
 Esta é uma variável de controle, marque "não concordo!"
 Tenho uma separação bem definida entre horas de trabalho e horas livres.
 A qualidade do trabalho final é priorizada em relação à agilidade.
 É importante saber se relacionar com pessoas de outras áreas de Tecnologia de Informação.
 Para realizar meu trabalho é importante buscar conhecimentos com outros profissionais ou empresas.
 Sinto que estou competindo com colegas de minha empresa ou equipe.
 Minha empresa oferece uma estabilidade que me permite continuar empregado(a) por muitos anos.
 Meu salário varia de acordo com minha produtividade.
 O software que desenvolvo é direcionado para uso específico de um cliente, e não para o público genérico.
 Para terminar um projeto no prazo, é mais importante improvisar do que cumprir o planejamento inicial.

POSICIONAMENTO POLÍTICO

67 [politicaArmas] Avalie as seguintes questões a partir de seu posicionamento pessoal:

A posse de armas deve ser proibida, pois ameaça a vida de outras pessoas.
 Possuir uma arma legalizada deve ser um direito do cidadão para se defender.

68 [politicaCriminal]

A maior causa da criminalidade é a falta de oportunidades iguais para todos.
 A maior causa da criminalidade é a maldade das pessoas.

69 [politicaMaconha]

O uso de maconha não deve ser proibido, porque é o usuário que sofre com as consequências.
 O uso de maconha deve ser proibido porque toda a sociedade sofre com as consequências.

70 [politicaHomossexual]

A homossexualidade deve ser aceita por toda a sociedade.
 A homossexualidade deve ser desencorajada por toda a sociedade.

71 [politicaDeus]

Acreditar em Deus torna as pessoas melhores.
Acreditar em Deus não necessariamente torna uma pessoa melhor.

72 [politicaMaioridade]

Adolescentes que cometem crimes devem ser punidos como adultos.
Adolescentes que cometem crimes devem ser reeducados.

73 [politicaSindicatos]

Os sindicatos são importantes para defender os interesses dos trabalhadores.
Os sindicatos servem mais para fazer política do que defender os trabalhadores.

74 [politicaPrivacidade]

O desrespeito à privacidade digital é uma violação inaceitável dos direitos dos cidadãos.
Apenas criminosos e terroristas deveriam se preocupar com a vigilância dos governos na Internet.

75 [politicaPobreza]

Boa parte da pobreza está ligada à falta de oportunidades iguais para que todos possam subir na vida.
Boa parte da pobreza está ligada à preguiça de pessoas que não querem trabalhar.

76 [politicaPrivatiza]

Deve-se evitar privatizar empresas públicas, pois alguns bens devem ser sempre de acesso público e gratuito.
Privatizar empresas públicas deve ser uma opção para melhorar um serviço prestado à população e ajudar na redução de gastos do Estado.

77 [politicaPartidos]

Os partidos políticos são importantes para representação dos diferentes interesses da sociedade.
Os partidos políticos não representam as necessidades da sociedade, apenas de alguns grupos poderosos.

78 [politicalnflacao]

Se não houver outra escolha, reduzir o desemprego deve ter prioridade sobre o controle da inflação.
Se não houver outra escolha, controlar a inflação deve ter prioridade sobre a redução do desemprego.

79 [politicalmpostos]

O Estado deve se responsabilizar com programas sociais, mesmo que por isso os impostos continuem altos.
O Estado deveria cobrar menos impostos, mesmo que reduzindo os programas sociais.

80 [politicaMercado]

O governo deve regular a economia para evitar crises e distribuir os recursos com mais eficiência.
Quanto mais livre o mercado, mais livres serão as pessoas.

81 [politicaClasses]

Uma sociedade ideal não deveria ter classes, ou pelo menos suas riquezas deveriam ser distribuídas mais igualmente.

Uma sociedade livre inevitavelmente possuirá classes, pois a propriedade particular de empresas é o que gera o dinamismo necessário da economia.

82 [politicaPosicaoTexto] Caso você tenha uma posição político-ideológica específica, como você se definiria?

Por favor, coloque sua resposta aqui:

83 [sexualidadeTexto] Como você definiria sua orientação sexual?

Por favor, coloque sua resposta aqui:

DICA: Por exemplo: "heterossexual", "bissexual", "homossexual"...

FINAL

84 [notificar] Gostaria de receber uma notificação assim que o banco de dados despersonalizado e os resultados da pesquisa forem publicados?

Sim
Não

85 [entrevista] Você estaria disposto(a) a conceder uma entrevista (por email ou bate-papo) como continuação desta pesquisa?

Sim
Não

86 [email] Por favor informe seu e-mail para contato:

Só responder essa pergunta sob as seguintes condições:

----- Scenario 1 -----

A resposta foi 'Sim' na questão '84 [notificar] ' (Gostaria de receber uma notificação assim que o banco de dados despersonalizado e os resultados da pesquisa forem publicados?)

----- OR Scenario 2 -----

A resposta foi 'Sim' na questão '85 [entrevista] ' (Você estaria disposto(a) a conceder uma entrevista (por email ou bate-papo) como continuação desta pesquisa?)

Por favor, coloque sua resposta aqui:

DICA: Seu endereço de e-mail NÃO será incluído no banco de dados final, nem divulgado para outros grupos de pesquisa ou terceiros.

87 [comentário] Por favor, sinta-se à vontade para usar este último espaço para fazer algum comentário sobre o questionário ou a pesquisa, se desejar:

Por favor, coloque sua resposta aqui:

Agradecemos por sua colaboração!

Obrigado por ter preenchido o questionário.