

# Computação, Infoproletariado e Liberdade Digital

Ciclo de Formações do Patos à Esquerda

**Vinícius Fonseca Maciel**

2023



Ciclo de

**Formações**



## Licença de Uso

Licença de Uso

### Documento sob a licença GFDL

Copyright (c) 2024 Vinícius F. Maciel.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".



# Table of Contents

## 1 Computação e Software Livre

► Computação e Software Livre

► Infoproletariado e Uberização

► Algoritmos de Destruição em Massa



# Computação

## 1 Computação e Software Livre

- A computação integra as bases da estrutura capitalista e ela é oportunamente utilizada no controle das massas.
- A militância da esquerda revolucionária anticapitalista precisa compreender o universo digital.
- Compreender significa poder estudar e contestar/creditar as configurações sociais que surgem a partir da evolução tecnológica.



# Computação

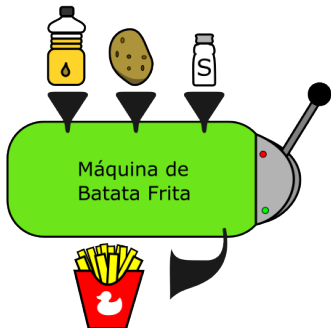
## 1 Computação e Software Livre

- Alguns conceitos importantes sobre computação:
  - Algoritmos
  - Programa
  - Linguagem de programação e código-fonte
  - Código de máquina
  - Sistemas Operacionais



# Computação — Algoritmos

1 Computação e Software Livre



- **Algoritmo:** sequência finita de passos que levam à resolução de um problema.
- **Problema:** fazer uma batata frita.
- Um algoritmo, informalmente, pode ser interpretado neste contexto como uma “receita”.
- Qual é a receita?

Figure: Uma máquina de fazer batata frita



# Computação — Algoritmos

1 Computação e Software Livre

- Para construir algoritmos usamos um conceito fundamentalmente utilizado na orquestração lógica de nossas sinapses em basicamente tudo que fazemos. São elas duas coisas: um conjunto histórico de dados e uma definição de sucesso.
- Na computação usamos a clássica sequência para definir o funcionamento de um algoritmo: *Entrada*  $\rightarrow$  *Processamento*  $\rightarrow$  *Saida*



# Computação — Programa

## 1 Computação e Software Livre

- A tradução do algoritmo para um formato em que uma entidade seja capaz de executá-lo define um **programa**.
- Descobrir o M.M.C. pode se constituir de um mesmo algoritmo tanto para o humano quanto para uma máquina, o que diferencia é a “linguagem” utilizada por eles.
- Software, aplicação, aplicativo e programa de computador são termos similares sensíveis ao contexto.





# Computação — Linguagem & Código-fonte

## 1 Computação e Software Livre

- Linguagem define um **conjunto de palavras** (léxico) que obedecem um **conjunto de regras** (sintaxe) a fim de **definir um sentido** (semântica).
- A **linguagem** entendida por um computador é **artificial**, formalmente definida na matemática.
- Ser um programador de computadores exige a compreensão da linguagem utilizada na escrita de um programa.
- O código-fonte é um arquivo construído com textos redigidos a partir do uso destas linguagens para formar o software.



- [illegible]



# Computação — Linguagem & Código-fonte

1 Computação e Software Livre

- Este é um exemplo de programa, chamado “Greetings” escrito em Python.
- Python é uma linguagem de muito alto nível, em outras palavras, bastante amigável

```
greetings.py  
  
pn = input('Primeiro nome: ')  
sn = input('Sobrenome: ')  
print('Olá,', pn, sn)
```

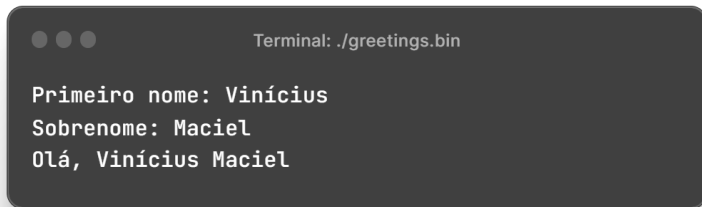
Figure: Código em Python



# Computação — Linguagem & Código-fonte

## 1 Computação e Software Livre

- Esta é a execução do programa “Greetings”.
- O usuário preencheu seu nome (entrada) e recebeu uma mensagem “Olá, Vinícius Maciel” (saída).



```
Terminal: ./greetings.bin  
  
Primeiro nome: Vinícius  
Sobrenome: Maciel  
Olá, Vinícius Maciel
```

Figure: Execução de programa



# Computação — Linguagem & Código-fonte

1 Computação e Software Livre

- Quanto menor o “nível” da linguagem, mais “difícil” pode ser sua compreensão humana, porém mais “fácil” para o computador.
- Próximo: mesmo programa escrito na linguagem C (médio nível)



# Computação — Linguagem & Código-fonte

## 1 Computação e Software Livre

```
greetings.c

#include <stdio.h>
#define MAX_LENGTH_BUF 128

void read_input(char* str)
{
    int i = 0;
    fgets(str, MAX_LENGTH_BUF-1, stdin);
    while(str[i] != 0xA && str[i] != 0x0) i++;
    str[i] = 0x0;
}

int main()
{
    char pn[MAX_LENGTH_BUF];
    char sn[MAX_LENGTH_BUF];
    printf("Primeiro nome: ");
    read_input(pn);
    printf("Sobrenome: ");
    read_input(sn);
    printf("Olá, %s %s\n", pn, sn);
    return 0;
}
```



# Computação — Linguagem & Código-fonte

1 Computação e Software Livre

- Próximo: mesmo programa escrito na linguagem NASM, que é um tipo de Assembly (baixo nível)



# Computação — Linguagem & Código-fonte

## 1 Computação e Software Livre

```
greetings.asm

global _start

;----- defines -----
%define SYS_READ    3
%define SYS_WRITE   4
%define STDIN       0
%define STDOUT      1
%define SYS_EXIT    1
%define BUFFER_MAX  128
;-----

section .bss
    pn: resb BUFFER_MAX
    sn: resb BUFFER_MAX

section .text

print_space:
    mov rcx, space
    mov rdx, 1
    jmp print_text.perform_print
print_newline:
    mov rcx, newline
    mov rdx, 1
    jmp print_text.perform_print
```





# Computação — Linguagem & Código-fonte

## 1 Computação e Software Livre

```
print_text:
    mov rsi, rcx
    mov rdx, 0x0
    .count:
        lodsb
        cmp al, 0xA
        je .perform_print
        cmp al, 0x0
        jz .perform_print
        inc rdx
        jmp .count
    .perform_print:
        mov rax, SYS_WRITE
        mov rbx, STDOUT
        int 0x80
        ret

read_text:
    mov rax, SYS_READ
    mov rbx, STDIN
    mov rdx, BUFFER_MAX
    int 0x80
    ret

print_greetings:
    mov rcx, hello
    call print_text
    mov rcx, pn
    call print_text
    call print_space
```



# Computação — Linguagem & Código-fonte

## 1 Computação e Software Livre

```
    call print_space
    mov rcx, sn
    call print_text
    call print_newline
    ret

_start:
    mov rcx, fname
    call print_text
    mov rcx, pn
    call read_text
    mov rcx, lname
    call print_text
    mov rcx, sn
    call read_text
    call print_greetings

_exit:
    mov rax, SYS_EXIT
    xor rbx, rbx
    int 0x80

section .data
    fname:      db "Primeiro nome: ", 0
    lname:      db "Sobrenome: ", 0
    hello:      db "Olá, ", 0
    space:      db " ", 0
    newline:    db 0xA, 0
```



# Computação — Linguagem & Código-fonte

1 Computação e Software Livre

- Próximo: linguagem de máquina (código binário)



# Computação — Código de Máquina

## 1 Computação e Software Livre

```
greetings.bin

0000225c: 00000000 00000000 00000000 00000000 00000000 00000000 .....
00002262: 00000000 00000000 00000000 00000000 00000000 00000000 .....
00002268: 00000000 01101110 01100001 01101101 01100101 00101110 .name.
0000226e: 01100001 01110011 01101101 00000000 01110000 01101110 asm.pn
00002274: 00000000 01110011 01101110 00000000 01110000 01110010 .sn.pr
0000227a: 01101001 01101110 01110100 01011111 01110011 01110000 int_sp
00002280: 01100001 01100011 01100101 00000000 01110000 01110010 ace.pr
00002286: 01101001 01101110 01110100 01011111 01101110 01100101 int_ne
0000228c: 01110111 01101100 01101001 01101110 01100101 00000000 wline.
00002292: 01110000 01110010 01101001 01101110 01110100 01011111 print_
00002298: 01110100 01100101 01111000 01110100 00000000 01110000 text.p
0000229e: 01110010 01101001 01101110 01110100 01011111 01110100 rint_t
000022a4: 01100101 01111000 01110100 00101110 01100011 01101111 ext.co
000022aa: 01110101 01101110 01110100 00000000 01110000 01110010 unt.pr
000022b0: 01101001 01101110 01110100 01011111 01110100 01100101 int_te
000022b6: 01111000 01110100 00101110 01110000 01100101 01110010 xt.per
000022bc: 01100110 01101111 01110010 01101101 01011111 01110000 form_p
000022c2: 01110010 01101001 01101110 01110100 00000000 01110010 rint.r
000022c8: 01100101 01100001 01100100 01011111 01110100 01100101 ead_te
000022ce: 01111000 01110100 00000000 01110000 01110010 01101001 xt.pri
000022d4: 01101110 01110100 01011111 01100111 01110010 01100101 nt_gre
000022da: 01100101 01110100 01101001 01101110 01100111 01110011 etings
000022e0: 00000000 01011111 01100101 01111000 01101001 01110100 ._exit
000022e6: 00000000 01100110 01101110 01100001 01101101 01100101 fname
```



# Computação — Código de Máquina

1 Computação e Software Livre

- O computador entende apenas as **instruções binárias**.
- **Compilador**: As outras linguagens (assembly, médio e alto nível) são **traduzidas** para conjuntos de valores binários.
- É possível, com dificuldade, **reverter** estes valores binários para uma linguagem de maior alto nível e assim **tentar** reproduzir um código-fonte próximo do original.



# Computação — Código de Máquina

## 1 Computação e Software Livre

- **Problema:** compiladores deixam o código mais “bagunçado”, isso torna difícil o trabalho de **engenharia reversa**.
- Este problema, entretanto, é **conveniente para os distribuidores de software proprietário**.





# Computação — Sistemas Operacionais

## 1 Computação e Software Livre

- O sistema operacional é um software especial que **controla o acesso** de outros softwares aos recursos oferecidos pelo computador: processadores, memória, unidades de armazenamento, periféricos...
- Permite com que **programas executem concomitantemente**: você consegue ouvir música enquanto escreve um documento.
- Sem os sistemas operacionais, os computadores são inúteis.







# Software Não Livre

## 1 Computação e Software Livre

- Todo software pode ter como meio de distribuição sua versão “pronta”, i.e. compilada, para o funcionamento nos computadores.
- Houve um tempo em que se comprava o hardware e, como “**brinde**”, vinham os softwares juntos ao código-fonte para que o usuário pudesse **modificar**.
- Comunidade de usuários de computador era pequena. Era comum programadores escreverem códigos para seus equipamentos e **compartilhá-los entre universidades e corporações**. (Collective (2009))



# Software Não Livre

## 1 Computação e Software Livre

- Martin Goetz registra a **primeira patente de software** em 19 de junho de 1968 (Torres (2013))
- IBM **inova** na cobrança separada de recursos em 23 de junho 1969. (Johnson (1998))





# Software Não Livre

1 Computação e Software Livre

Alguns efeitos da falta de liberdade:

- A clássica luta contra os **gerenciadores de direitos digitais** (DRM).
  - Restrições de cópia, geolocalização, integridade do dispositivo, versões obrigatórias, ... (GNU (2023))
- **Restrição** de aproveitamento de recurso físico por software.
  - ex. “DLC” de carro de luxo.
- *Spywares* e *Backdoors* injetados **propositalmente**.
- **Roubo** de dados e **escravização** do usuário.
- A **obsolescência programada** via software.





# Software Não Livre

1 Computação e Software Livre

Alguns “sabores” da não liberdade:

- **Software proprietário/privativo:** qualquer software que não se tem acesso ao código-fonte, nem tem permissão de redistribuição gratuita.
- **Freeware:** permitem redistribuição gratuita. (sem código-fonte)
- **Shareware:** redistribuição permitida, e.g. trial/demo, mas necessita licença obrigatória. (sem código-fonte)
- **Softwares privados:** desenvolvido para um usuário específico, sem redistribuição. (este usuário *pode* ter acesso ao código-fonte)
- **Software comercial:** Distribuição sob licenciamento de venda. É até possível vender uma distribuição de um software livre, e.g Red Hat. (*pode* ou *não* ter código-fonte)



# Software Livre

1 Computação e Software Livre

## Liberdades

- 0. A liberdade de **executar** o programa como você desejar, para qualquer propósito.
- 1. A liberdade de **estudar** como o programa funciona, e adaptá-lo às suas necessidades. Para tanto, acesso ao código-fonte é um pré-requisito.
- 2. A liberdade de **redistribuir** cópias de modo que você possa ajudar outros.
- 3. A liberdade de **distribuir** cópias de suas versões modificadas a outros. Desta forma, você pode dar a toda comunidade a chance de beneficiar de suas mudanças. Para tanto, acesso ao código-fonte é um pré-requisito.



# Software Livre

## 1 Computação e Software Livre

- Se o **usuário tem todas as quatro liberdades** garantidas em um determinado software, aquele software é livre, e se o usuário se **compromete** a usar software livre, então ele é **livre** (Collective (2009)).
- Ser um software “de graça” não o torna um software livre.
- Hoje é possível usar um sistema computacional totalmente livre!
  - Desde firmwares e gerenciadores de boot até os softwares utilitários do dia a dia.



# Software Livre

1 Computação e Software Livre

- “O Software Livre é o primeiro movimento de luta pela **libertação do ciberespaço.**”
- Surgiu da **indignação** dos “hackers” do laboratório de IA do MIT nos anos 80. Em destaque: “Richard Stallman”.
- Os “hackers” do MIT eram conhecidos por quebrar a exclusividade de uso dos recursos computacionais da instituição e sempre conseguir **democratizar o acesso** ao que precisavam.



# Software Livre

## 1 Computação e Software Livre

- Symbolics monopoliza as máquinas Lisp do laboratório de IA do MIT e controla a atmosfera colaborativa da comunidade “hacker”.
- Stallman em **retaliação à empresa Symbolics**, que decidiu acabar com o compartilhamento livre de códigos dentro do laboratório de IA, reprogramou todas as soluções da empresa e continuou compartilhando o código entre os seus. (Collective (2009))



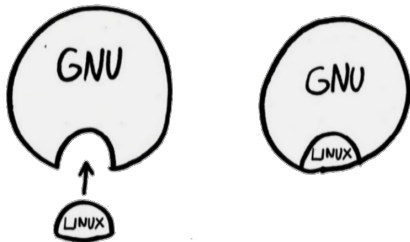




# Software Livre

## 1 Computação e Software Livre

- Quase todo software importante do meio da década de 80 é proprietário.
- Em 1985, Stallman lança o manifesto GNU e cria *Free Software Foundation*(FSF).
- GNU desenvolve utilitários e desenvolve a casca (shell) que faz a interface usuário-sistema operacional.
- GNU = GNU's Not Unix
- Faltava o núcleo (*kernel*) que integraria um Sistema Operacional completo.





# Software Livre — Linux

## 1 Computação e Software Livre

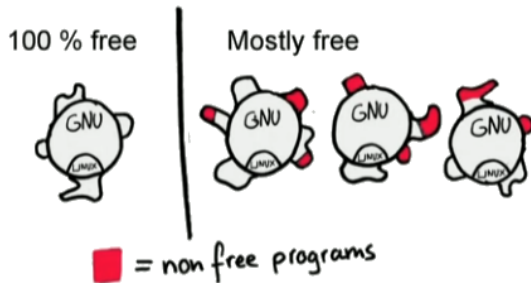
- Linux (1991) é um núcleo de sistema operacional (*kernel*), o termo correto para uma distribuição como Debian é GNU/Linux.
- Linux = “Unix-like” do Linus Torvalds.
- O GNU/Linux é utilizado pela maioria das corporações ativas na internet como solução para servidores.
- Android possui o Linux como *kernel*.



# Software Livre — Linux

## 1 Computação e Software Livre

- Linux e GNU possuem licença GPL, mas é constantemente contaminado e redistribuído com soluções proprietárias.





# Software Livre — Licenças

## 1 Computação e Software Livre

Existem quatro tipos de licenças básicas que se originaram a partir do projeto GNU.

- **GNU GPL:** Licença Pública Geral - Aplicações
- **GNU LGPL:** Licença Pública Geral Menor - Bibliotecas de Software
- **GNU AGPL:** Licença Pública Geral Affero - Servidores
- **GNU FDL:** Licença de Documentação Livre - Manuais e Documentos de Software





+



100% Livres



dynebolic  
PureOS



LUKS



searX



LXQt





# Software Livre - Críticas

## 1 Computação e Software Livre

- **Opinião:** Forte em aspectos tecnológicos, frágil em aspectos políticos.
- Em Torres (2013), evidencia-se uma **retórica neoliberal** submersa na estrutura da comunidade software livre.
- A *open source initiative* é uma dissidência da comunidade software livre para “amenizar” o caráter combativo e se tornar **mais palatável para o mundo corporativo**.
- Autoritarismo, egocentrismo e elitismo tecnológico ainda permeiam os canais da comunidade.



# Digitaliza

1 Computação e Software Livre

- Projeto de restaurações de dispositivos computacionais (Maciel (2021)).
- Priorização de Softwares Livres e código-fonte aberto.
- **Sistemas legados lidam bem com o software livre.**
- É interessante para o projeto também a **recuperação de smartphones**, entretanto a limitação fica a cargo do modelo e do tipo dos Androids Custom Roms.





## Table of Contents

### 2 Infoproletariado e Uberização

► Computação e Software Livre

► **Infoproletariado e Uberização**

► Algoritmos de Destruição em Massa





# Proletariado na era digital

## 2 Infoproletariado e Uberização

- Percepção da quebra de paradigma laboral: não existe mais o trabalhador sem um *smartphone*.
- O **infoproletariado**, como termo, surge da compreensão de um novo subconjunto de tipos de trabalho que mesclam a tecnologia do século XXI e as condições de trabalho abusivas do século XIX. (Antunes and Braga (2009))



# Proletariado na era digital

## 2 Infoproletariado e Uberização

- A indústria 4.0 acentua o uso das tecnologias da informação e comunicação (TIC).
- A expansão do trabalho precarizado pelas TICs significará a **ampliação dos processos produtivos** ainda mais automatizados e robotizados em toda a cadeia de valor, de modo que a **logística empresarial será toda controlada digitalmente**. (Antunes (2020))
- Ampliação do trabalho morto.



# Uberização

## 2 Infoproletariado e Uberização

- Mutaç o do *Zero Hour Contract*: o trabalho   condicionado ao chamado de uma plataforma.
- O trabalhador ganha estritamente pelo que fizer, n o recebe pelo **tempo que espera**.
- **Uberiza o**: as rela  es de trabalho s o crescentemente **individualizadas** e **invisibilizadas**, assumindo a apare ncia de “prest o de servi os” e **oblitando as rela  es de assalariamento** e de explora  o de trabalho. (Antunes (2020))



## Breque dos Apps

### 2 Infoproletariado e Uberização

- **Breque dos Apps:** Primeira grande mobilização grevista organizada por entregadores de aplicativos no Brasil. (01/07 e 25/07 de 2020)
- Clima de pandemia global e intensa precarização dos entregadores: pulou de 280 mil para 500 mil entregadores registrados.
- Entregadores Antifacistas, Paulo Galo e reivindicações:
  - Diminuição do valor por km rodado, diminuição da taxa mínima de entrega, fim dos bloqueios arbitrários, fim do sistema de pontuação, auxílio pandemia (EPI e auxílio saúde)



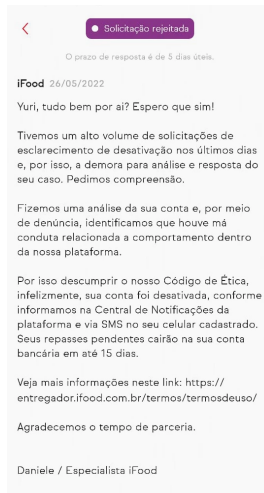


## Yuri Fontes

### 2 Infoproletariado e Uberização

# ‘TUDO BEM POR AÍ?’

Yuri morreu fazendo entrega para o iFood. 11 dias depois, sua conta foi desativada por ‘má conduta’





## Table of Contents

### 3 Algoritmos de Destruição em Massa

- ▶ Computação e Software Livre
- ▶ Infoproletariado e Uberização
- ▶ Algoritmos de Destruição em Massa



# Algoritmos de Destruição em Massa

## 3 Algoritmos de Destruição em Massa

- *Weapons of Math Destruction* (Algoritmos de Destruição em Massa) termo cunhado por Cathy O'Neil.
  - Três características: **difusão, mistério e destruição.**
- Modelos matemáticos podem ser construídos à base de **preconceitos, equívocos e vieses** humanos e integram sistemas de software que gerem nossas vidas. (O'Neil (2020))
- Há exemplos absurdos da prática arbitrária do uso da modelagem matemática em questões sociais importantes no EUA.



# Algoritmos de Destruição em Massa

## 3 Algoritmos de Destruição em Massa

- **Modelagem de valor agregado:** Erradicação dos professores de baixa performance, em 2009 Washington D.C., como forma de melhorar a qualidade de ensino. (caso Michelle Rhee e a ferramenta IMPACT de avaliação de professores)
  - *New York Post* conseguiu os scores de teste dos professores e publicou como um ato de envergonhamento dos professores de baixa performance.
  - Código do sistema de avaliação era proprietário.





# Algoritmos de Destruição em Massa

## 3 Algoritmos de Destruição em Massa

- **Algoritmo de Policiamento Preditivo:** técnica que utiliza aprendizado de máquina para calcular previsões de atividades criminosas e assim, preventivamente, alocar policiamento na cidade. A base dados consiste de horários, locais, fatores ambientais e natureza dos crimes passados.
  - Enviesado contra a população pobre e preta.





# Algoritmos de Destruição em Massa

## 3 Algoritmos de Destruição em Massa

- **Risco de reincidência criminal:** técnica que faz uso de um questionário aplicado a um réu e um modelo matemático baseado em scores. O modelo é alimentado com registros criminais.
- O questionário possui **perguntas enviesadas** como: saúde mental, violência no bairro que mora, parentesco com pessoas criminosas.
- O resultado da aplicação do algoritmo ajuda o sistema jurídico estadunidense a decidir o **tempo de reclusão**.



# Algoritmos de Destruição em Massa

## 3 Algoritmos de Destruição em Massa

“Os algoritmos **não tornam as coisas justas** se forem aplicados de forma cega e displicente. Eles **repetem nossas práticas passadas**, nossos padrões. Eles automatizam o *status quo*. Isso seria ótimo se tivéssemos um mundo perfeito, mas não temos. Achamos que os algoritmos são objetivos, verdadeiros e científicos, isso é um truque de marketing. **Essa é uma luta política**. Precisamos exigir **prestação de contas dos ‘senhores dos algoritmos’**.” - Cathy O’Neil.



## Referências

### Referências

Antunes, R. (2020). *Uberização, trabalho digital e Indústria 4.0*. Boitempo Editorial.

Antunes, R. and Braga, R. (2009). *Infoproletários: degradação real do trabalho virtual*. Boitempo Editorial.

Collective, D. R. T. (2009). An anarchist's Guide to free software. Acessado em: 03/10/2023.

GNU (2023). Proprietary drm.

<https://www.gnu.org/proprietary/proprietary-drm.html.en>. Acessado em: 03/10/2023.

Johnson, L. (1998). A view from the 1960s: how the software industry began. *IEEE Annals of the History of Computing*, 20(1):36–42.



## Referências

### Referências

- Maciel, V. F. (2021). Mob lança o projeto digitaliza em patos de minas.  
<https://patosaesquerda.com.br/o-mob-patos-de-minas-lanca-o-projeto-digitaliza/>.  
Acessado em: 05/10/2023.
- O'Neil, C. (2020). *Algoritmo de Destruição em Massa: como o big data aumenta a desigualdade e ameaça a democracia*. Editora Rua do Sabão.
- Torres, A. L. (2013). A tecnologia do software livre: uma história do projeto técnico e político do gnu. Master's thesis, Faculdade de Filosofia, Letras e Ciências Humanas da Universidade de São Paulo. Departamento de História, São Paulo.