

1. NARRATIVA COMPUTACIONAL COM JUPYTER NOTEBOOK COMO APOIO À PESQUISA NAS CIÊNCIAS SOCIAIS APLICADAS

*Milton Shintaku
Rebeca dos Santos de Moura
Lucas Rodrigues Costa*

1.1 INTRODUÇÃO

O termo “narrativa” tem sido muito empregado no meio político, sobretudo no início da segunda década do século XXI, para denominar a elaboração de histórias sobre determinado evento. Nesses casos, o termo torna-se o mais apropriado para tal fim. Entretanto, em sua etimologia, narrativa vem do latim *narrare*, que significa tornar algo conhecido, que, por sua vez, tem origem em *gnarus*, isto é, o que tem conhecimento. *Gnarus* deu origem a várias palavras do português, como ignorante, aquele que não tem conhecimento etc.

Assim, a narrativa, em sua forma mais pura, tem relação com a transmissão do conhecimento. Na linguística, por exemplo, Paiva (2008) descreve a narrativa como algo contado e recontado a partir de algo real ou fictício, uma série de eventos, relato de acontecimentos, entre outras possibilidades. Para essa área do conhecimento, o estudo das narrativas é importante por possibilitar o entendimento da história e da literatura escrita.

Na visão educativa de Kearney (2012), entende-se que narrativas são instrumentos de compreensão da condição humana, com uma complexidade filosófica que pode ser discutida em cinco visões diferentes, a saber: enredo, recriação, alívio, sabedoria e ética. Nesse caminho, o autor encerra a discussão com a questão ética da narrativa, na qual o narrador é o sujeito (agente) que sofre das interferências da própria história, de forma que ela nunca será neutra.

Squire (2014), por sua vez, discute a definição de narrativas, primeiro com foco na concentração, movimentos, sucessões de signos narrativos, descrição e construção de sentidos. Com isso, constata que a narrativa adota questões sociais, culturais e históricas, atribuindo um contexto particular. Por isso, ela pode assumir diversos formatos e tipos, com raízes semióticas.

Como metodologia de pesquisa, os estudos narrativos têm sido amplamente utilizados. Por isso, Clandinin e Connelly (2011) defendem a pesquisa com narrativas, na medida em que representam experiências pessoais e sociais, com continuidade que possibilita verificar presente, passado e futuro. Nesse sentido, apresenta um cenário complexo, em que a experiência estudada pode ser fruto de outras, o que exige aprofundamento de pesquisa.

Nota-se, no entanto, que narrativas são instrumentos relacionados às ciências humanas e sociais, muito utilizadas como metodologia de pesquisa. Como consequência, ao termo “narrativa” podem ser atribuídos qualificadores, de forma a apresentar *facetas* refinadas, como em textos históricos, literários, visuais, digitais etc. Com isso, possibilita-se a visão das narrativas de acordo com certo viés.

Dentre as novas terminologias destacadas neste capítulo, ressalta-se a narrativa computacional, para a qual se apresenta uma nova tendência nas ciências da computação com o uso das narrativas. Com isso, aliam-se os pontos discutidos nas ciências sociais e humanas, com as ciências técnicas e a computação. Salienta-se a computação como disciplina transversal e instrumento de estudos. Em um cenário em que as tecnologias digitais estão presentes em todas as áreas do conhecimento, as narrativas computacionais podem ser úteis nas ciências sociais aplicadas, servindo como ferramenta para apoio à pesquisa.

Pela construção do termo, narrativa computacional compreende a atividade narrativa restrita a questões voltadas para a ciência da computação. Assim, utilizam-se as técnicas da narrativa para descrever algoritmos, uma vez que, mesmo sendo uma representação da lógica, torna-se algo pessoal, como uma história contada pelo programador e influenciada por experiências pessoais, formação, atuação profissional, entre outras.

Com o desenvolvimento das Tecnologias da Informação e Comunicação (TIC), a oferta de algoritmos computacionais tem sido cada vez mais utilizada em todas as áreas do conhecimento. Assim, torna-se essencial que se compreendam os algoritmos, uma vez que eles tornam a lógica acessível a todos. Logo, deve-se narrar a lógica do algoritmo como se contasse uma história, apresentando o conhecimento contido no programa de computador.

1.2 SOBRE AS NARRATIVAS COMPUTACIONAIS

Desde o início da computação, um dos pontos principais é a construção de programas que materializam um algoritmo, ou seja, uma lógica que representa uma sequência organizada de elementos, com complexidade cada vez maior, voltada à resolução de uma atividade. Entretanto, os programas tendem a ser a representação da forma de pensar do programador, a lógica pessoal amparada por todas as experiências do profissional. Nesse sentido, durante muito tempo havia a máxima na computação, em que somente o autor de um programa conseguia fazer sua manutenção.

Em parte, como solução na computação organizacional, foi criada a linguagem de programação *Common Business Oriented Language (Cobol)*, que atuava quase como um inglês estruturado. Criado pelo Departamento de Defesa Norte-Americano para ser utilizada em computadores de grande porte (*mainframe*), possui uma estrutura rígida e burocrática, mas que possibilita ser construída, executada e mantida por equipe de colaboradores, devido às características apresentadas pela linguagem.

Entretanto, o *Cobol* apresenta certas limitações, mesmo com todos os avanços apresentados pela sua evolução temporal, desde a sua criação até as versões mais modernas, como a lançada em 2002, pela *International Standardization Organization (ISO)*. Com estrutura e funcionalidades voltadas para programação, principalmente em lotes, o *Cobol* apresenta certas restrições quanto ao uso em pesquisa. Assim, sua função se volta mais à informatização de atividades administrativas e contábeis.

Mesmo com todas as limitações, o *Cobol* apresentava um embrião do que seria considerado como narrativas computacionais, no qual a lógica

do programa era explicada. Nesse caso, o próprio código e estrutura do programa ofertava explicações que facilitavam a compreensão de seu funcionamento. Outro ponto de destaque era a existência dos comentários no programa, muito utilizados para apresentar sua lógica.

Numa visão mais atual, narrativa computacional tem relação com a Inteligência Artificial (IA), tanto que Riedl (2016) defende o uso dessas narrativas com um viés da construção de formas práticas de criar contextos socioculturais voltados para máquinas. Da mesma forma, possibilita que outros programadores possam entender a lógica utilizada nos programas.

Ontañón e Zhu (2011) descrevem as narrativas computacionais como histórias criadas para serem contadas para computadores, a fim de poderem processá-las. Com isso, adiciona novos elementos no longo processo entre humanos e máquinas, desde a criação de padrões que mesclavam questões sintáticas com semânticas, a fim de que humanos e máquinas pudessem compreender os objetos digitais voltados para o processamento.

1.2.1 PESQUISAS COM A NARRATIVA COMPUTACIONAL

Vall-Vargas, Zhu e Ontañón (2017) categorizam os estudos da Narrativa Computacional em duas áreas: geração e análise. Ambos os casos são estudos de desenvolvimento de algoritmos para que computadores gerem ou analisem narrativas, a fim de tratar melhor os dados ou informações, numa tentativa de aproximar o processamento das pessoas.

No que tange à análise e extração de informação de narrativas, há alguns desafios, tendo em vista a amplitude de possibilidades que as narrativas oferecem, tratando de extração de informação de textos, entendimento de contexto etc. Nesse sentido, muitos casos requerem algoritmos especializados em processamento de linguagem natural a fim de estabelecer modelos que atendam a determinados tipos de narrativas, levando-se em conta a imensa variedade de tipos textuais.

No que diz respeito a dados, no entanto, em alguns pontos torna-se mais simples. Assim, inicialmente as bases de dados eram quase pessoais, atendendo apenas a uma aplicação. Posteriormente, com os administradores de bases de dados, surgiram os dicionários de dados, metadados

que definiam os elementos de dados. A proposta atual é que as narrativas computacionais ofereçam o contexto dos dados, possibilitando seu compartilhamento não apenas em sentido institucional, mas de forma global, principalmente na chamada ciência de dados.

Para tanto, o uso de narrativas computacionais com dados torna-se mais eficaz com o apoio dos chamados cadernos de notas computacionais, ou, no original em inglês, *computational notebooks* (Rule; Tabard; Holland, 2018). Segundo esses autores, os *notebooks* são apropriados para o compartilhamento de narrativas, as quais são compostas não apenas pelas informações, mas pelos dados, códigos, entre outros elementos, atendendo a humanos e máquinas.

Nesse sentido, entendem-se narrativas como a forma pela qual um pesquisador registra seus estudos utilizando um notebook, depositando informações, formas de processamento, resultados em gráficos, tabelas etc. A narrativa computacional se converte, então, na capacidade de publicação organizada de vários elementos da pesquisa, dos dados e da forma como eles se processaram, a fim de possibilitar sua validação, reúso, compartilhamento de código, entre outros fatores.

1.2.2 JUPYTER NOTEBOOK E A NARRATIVA COMPUTACIONAL

O *Jupyter Notebook* é uma ferramenta poderosa e versátil para programação interativa e análise de dados, que permite a criação de documentos dinâmicos, os quais combinam texto explicativo, código executável, visualizações e outras mídias, como imagens e vídeos, tudo em um único ambiente integrado (*Jupyter Notebook, 2023a, on-line*). O *Jupyter Notebook* suporta uma ampla variedade de linguagens de programação, incluindo *Python, R, Julia, Scala* e muitas outras, por meio de *kernels* específicos para cada linguagem (*Jupyter Notebook, 2023b, on-line*). Os *kernels* são processos computacionais que fornecem suporte para a execução de código em uma linguagem de programação específica do ambiente *Jupyter Notebook*. Dessa forma, é possível usar o mesmo ambiente para trabalhar com diferentes linguagens e tecnologias.

Além de oferecer a execução de códigos de forma interativa, facilitando a realização de testes e experimentos rapidamente, com o *Jupyter Notebook*

ainda é possível criar relatórios e documentos ricos em conteúdo, que podem ser compartilhados com outras pessoas. É possível, também, criar recursos avançados para visualização de dados, permitindo a criação de gráficos e visualizações complexas e interativas, facilitando a análise, a comunicação e a manipulação dos documentos por meio dos resultados obtidos.

Uma das principais vantagens do *Jupyter Notebook* é sua flexibilidade e adaptabilidade, uma vez que pode ser usado tanto localmente, em uma máquina individual, quanto em servidores remotos, como na nuvem, permitindo a colaboração em projetos de equipe. Com isso, podem ser criados conjuntos de desenvolvedores em colaboração, ou mesmo grupos de pesquisadores atuando com conjunto de dados, compartilhando códigos e algoritmos.

No entanto, é importante ressaltar que o *Jupyter Notebook* depende da linguagem de programação *Python* para funcionar corretamente. Isso significa que é necessário ter um ambiente *Python* configurado e instalado em seu sistema antes de começar a trabalhar com os notebooks. A infraestrutura necessária para o bom funcionamento da plataforma requer certo conhecimento de *Python*.

Uma das vantagens de se utilizar o *Python* em conjunto com o *Jupyter Notebook* é a possibilidade de aproveitar funções nativas da linguagem, bem como instalar quantos pacotes adicionais forem necessários para atender às demandas de cada projeto. Dessa forma, é possível contar com recursos avançados de processamento de dados e análise estatística.

Dentre os pacotes e bibliotecas mais populares e amplamente utilizados para tratamento e análise de dados, podem ser citados:

Pandas é uma biblioteca que oferece estruturas de dados e ferramentas de análise de alto nível, permitindo a manipulação de grandes conjuntos de dados de forma rápida e eficiente (Panda, 2023).

Matplotlib é uma biblioteca para criação de gráficos e visualizações de dados em *Python*, que permite a criação de diversos tipos de gráficos, incluindo linhas, barras, dispersão e histogramas (Matplotlib, 2023).

Numpy é uma biblioteca que permite a realização de cálculos numéricos complexos e operações matemáticas avançadas, como álgebra linear e transformadas de *Fourier* (NumPy, 2023).

Há outras bibliotecas e pacotes com a oferta de outras facilidades voltadas ao tratamento de dados, muitos dos quais com certas especificidades. Assim, essas bibliotecas apresentam o básico necessário para atuação em dados de pesquisas em ciência social aplicada. Muitas delas, aliás, estão voltadas a operações mais simples, sem grandes quantidades de cálculos estatísticos, muitas das quais dizem respeito à geração de gráficos que serão discutidos nos textos.

1.2.2.1 EXEMPLO DE USO DE JUPYTER NOTE E NARRATIVAS EM MANIPULAÇÃO DE DADOS

A coleta de dados, em alguns casos, pode ser efetuada em fontes que oferecem, entre outras vantagens, a exportação em formatos abertos, como o *Comma Separated Value (CSV)*. Toma-se como exemplo a Biblioteca Digital Brasileira de Teses e Dissertações (BDTD), que agrega teses e dissertações defendidas em 133 instituições brasileiras, com mais de 800 mil documentos em texto integral. Assim, torna-se uma fonte indispensável para pesquisas.

Numa busca simples pelo termo *Python*, feita no dia 24 de abril de 2023, foram encontrados 487 documentos, como mostra a Figura 1. A BDTD exporta os dados, identificando os campos em inglês com a codificação *Unicode Transformation Format 8 (UTF-8)*. Logo, pode ser que sejam necessários ajustes para leitores padrão *ISO*.

Figura 1 - Dados extraídos da BDTD no formato CSV abertos em leitor de planilha.

id	title	authors	subjectsPOR	institutions	types	accesslevel	publicationDates	urls	formats	languages
2	UFG_ea73bfca9Aritmética com Py	primary[Cavalcaei LÃgica]	Programaã	UFG	masterThesis	openAccess	2018	http://i	masterThes	por
3	UERJ_43e00404 Retroanálise com o	primary[Santiago]	Engenharia civil	En UERJ	masterThesis	openAccess	2019	http://i	masterThes	por
4	UTFPR-12_b60t: O ensino de matem	primary[Presente Matemática - Estu	UTFPR		masterThesis	openAccess	2019	http://i	masterThes	por
5	UNICAMP-30_9 Desenvolvimento di	primary[Gonãsa Programaã	UFG	em UNICAMP	masterThesis	openAccess	2007	https://	masterThes	por
6	UTFPR-1_6649c: O ensino de matem	primary[Presente Matemática - Estu	UTFPR		masterThesis	openAccess	2019	http://i	masterThes	por
7	URGS_794ef535 A Python library for	primary[Sarate, IFã-sica teãrica]	Te	UFRGS	masterThesis	openAccess	2022	http://i	masterThes	eng
8	USP_66ebecd3: Automaãdo de pr	primary[Paulo Cãzar Peixoto de Se	USP		masterThesis	openAccess	2019	https://	masterThes	por
9	UFMT_5e139fb: Estudo de ãrbitas p	primary[Almeida Leis de Kepler]	Grav	UFMT	masterThesis	openAccess	2016	http://i	masterThes	por
10	UFPB-2_f927e2: Monty Python e por	primary[Ramos, Audiovisual]	Cultur	UFPB	masterThesis	openAccess	2017	https://	masterThes	por
11	UFPE_e0e946c: Integraãdo pytho	primary[REIS, Eli Engenharia Civil]	Eli	UFPE	masterThesis	openAccess	2018	https://	masterThes	por
12	UNICAMP-30_6 Implementaãdo c	primary[Gonãsa Compressãdo de in	UNICAMP		masterThesis	openAccess	2003	https://	masterThes	por
13	UNICAMP-30_7 Ambiente de suport	primary[Silva, AI Processamento de	UNICAMP		masterThesis	openAccess	2003	https://	masterThes	por
14	FGV_3ba41619: Visualizaãdo de c	primary[Oliveira Coleãães Digital]	FGV		masterThesis	openAccess	2021	https://	masterThes	por
15	SCAR_c4740dc2 Efeitos da alimentaã	primary[Ciprianc Fisiologia]	Funãã	UFSCAR	masterThesis	openAccess	2013	https://	masterThes	por

Fonte: Captura de tela (2023).

Usando a *biblioteca Pandas* do *Python*, é possível ler e manipular dados em diversos formatos, incluindo arquivos *CSV* e *Excel*. Usando o arquivo exportado da BDTD, o exemplo a seguir mostra como ler um arquivo *CSV* e criar um *dataframe*, isto é, uma estrutura de dados semelhante a uma tabela em um banco de dados. Com ela, os dados podem ser manipulados de forma mais fácil.

Quadro 1 - Comandos em *Python* para ler arquivos *CSV* e criar um *dataframe*.

```
# Importação das bibliotecas necessárias
import pandas as pd

# Lê o arquivo CSV
df = pd.read_csv('search_results.csv', sep=';')

# Mostrar as colunas do dataframe
print(df.columns)

# Selecionar algumas colunas do dataframe
df_selected = df[['title', 'authors', 'institutions',
                  'types', 'publicationDates', 'languages']]

# Mostrar as primeiras linhas de conteúdo
print(df_selected.head())
```

Fonte: Elaborado pelos autores (2023).

Nesse exemplo, o comando `pd.read_csv("search_results.csv", sep=";")` comporta um arquivo CSV chamado `search_results.csv` e o armazena em um `DataFrame` do `Pandas` chamado `df`. O parâmetro `sep=";"` especifica que o separador utilizado no arquivo CSV é o ponto e vírgula. Tal comando permite ler e estruturar os dados do arquivo CSV em uma estrutura tabular, facilitando a manipulação e análise dos dados por meio das funcionalidades do `Pandas`. O comando `df.columns` apresenta as colunas existentes no `dataframe`, com o seguinte resultado:

Quadro 2 - Código Python de apresentação das colunas existentes no dataframe.

```
['id', 'title', 'authors', 'subjectsPOR', 'institutions',
'types', 'accesslevel', 'publicationDates', 'urls',
'formats', 'languages']
```

Fonte: Elaborado pelos autores (2023).

A partir das colunas do `dataframe`, podemos realizar a seleção vertical ou `slice`, por meio do comando `df[['title', 'authors', 'institutions', 'types', 'publicationDates', 'languages']]`, que seleciona as colunas designadas. Em seguida, as primeiras linhas de conteúdo do `dataframe` são apresentadas com o comando `print(df_selected.head())`, cujo resultado é o seguinte:

Quadro 3 - Código Python de apresentação dos resultados obtidos pelo comando chamado df.

```
authors institutions types
pubDates languages
0 Cavalcante, Rogério da Silva UFG masterThesis 2018
  por
1 Santiago, Carlos Alexandre UERJ masterThesis 2019
  por
2 Pesente, Guilherme Moraes UTFPR masterThesis
2019 por
3 Gonçalves Neto, Jahyr UNICAMP masterThesis 2007
  por
4 Pesente, Guilherme Moraes UTFPR masterThesis
2019 por
```

Fonte: Elaborado pelos autores (2023).

Para extrair apenas a lista de autores do conjunto de dados, podemos usar uma simples seleção de colunas `print(df['authors'])`. Entretanto, os dados dessa coluna apresentam informações de *primary* (autor primário) e o *link* do *Lattes* do autor entre parênteses, como vemos no quadro a seguir:

Quadro 4 - Código Python de apresentação dos resultados obtidos pelo comando `print(df['authors'])`.

```
0          primary[Cavalcante, Rogério da Silva(NA)]
1          primary[Santiago, Carlos Alexandre de Almeida(...
2          primary[Presente, Guilherme Moraes(http://latte...
3          primary[Gonçalves Neto, Jahyr, 1980-(NA)]
4          primary[Presente, Guilherme Moraes(http://latte...
...
482          primary[Hertzog, Lucas(NA)]
483          primary[Botelho, Gilberto Garcia(http://lattes...
484          primary[Marinho, Jos?? Lino do Nascimento(http...
485          primary[Xavier, Pedro Armentano Mudado(http://...
486          primary[Ferreira, Leandro Martins(http://latte...
Name: authors, Length: 487, dtype: object
```

Fonte: Elaborado pelos autores (2023).

Podemos utilizar os métodos `str.replace()` e `str.strip()` do *Pandas* alinhado ao uso de expressões regulares para tratar o dado e remover as informações desnecessárias, como mostrado a seguir:

Quadro 5 - Comandos em Python `str.replace()` e `str.strip()` para uso de expressões regulares.

```
# Remover os prefixos e sufixos da coluna 'authors'
df['autores'] = df['authors'].str.replace(r'^primary\
[|\]$',' ', regex=True)

# Remover os conteúdos entre parênteses da coluna
'autores'
df['autores'] = df['autores'].str.replace(r'\([^)]*\)',
'', regex=True).str.strip()
```

Fonte: Elaborado pelos autores (2023).

Assim, com o resultado do processamento, tem-se a lista de autores após tratamento:

Quadro 6 - Código Python de apresentação dos resultados obtidos pelo comando `str.replace ()` e `str.strip ()`.

```

0          Cavalcante, Rogério da Silva
1    Santiago, Carlos Alexandre de Almeida
2          Presente, Guilherme Moraes
3    Gonçalves Neto, Jahyr, 1980-
4          Presente, Guilherme Moraes
          ...
482          Hertzog, Lucas
483          Botelho, Gilberto Garcia
484    Marinho, Jos?? Lino do Nascimento
485          Xavier, Pedro Armentano Mudado
486          Ferreira, Leandro Martins
Name: autores, Length: 487, dtype: object

```

Fonte: Elaborado pelos autores (2023).

Uma função interessante do *Pandas* é a `value_counts()`, que calcula valores únicos em determinada coluna do *dataframe*. Com essa função, podemos extrair informações importantes dos tipos de arquivos do conjunto de dados, conforme o exemplo a seguir:

Quadro 7 - Comandos em Python `value_counts ()` para calcular valores únicos.

```

# Contar valores únicos em uma coluna:
contagem_tipos = df['types'].value_counts()

# Imprime o dataframe resultante
print(contagem_tipos)

```

Fonte: Elaborado pelos autores (2023).

O resultado desse comando é o seguinte:

Quadro 8 - Código Python de apresentação dos resultados obtidos pelo comando `value_counts ()`.

```

masterThesis    382
doctoralThesis  105

```

Fonte: Elaborado pelos autores (2023).

Esse exemplo pode ser aprimorado para mostrar o resultado da porcentagem em relação ao total usando o parâmetro `normalize=True` no método `value_counts()` e multiplicando por 100 para converter em porcentagem, como mostrado a seguir:

Quadro 9 - Comandos em Python `normalize=True` para visualização em porcentagem.

```
# Contar valores únicos em uma coluna, com normalização
contagem_tipos_porcentagem = df['types'].value_
counts(normalize=True) * 100

# Imprime o dataframe resultante
print(contagem_tipos_porcentagem)
```

Fonte: Elaborado pelos autores (2023).

O resultado desse comando é o seguinte:

Quadro 10 - Código Python de apresentação dos resultados obtidos pelo comando `normalize=True`.

masterThesis	78.439425
doctoralThesis	21.560575

Fonte: Elaborado pelos autores (2023).

O *Matplotlib* é uma biblioteca para visualização de dados em *Python*. Ela permite criar gráficos de linhas, barras, dispersão, histogramas, entre outros tipos de visualizações. O exemplo a seguir une o método `value_counts()` e o gráfico de pizza da biblioteca *Matplotlib*. Os comandos básicos para o gráfico são o `df['institutions'].value_counts()`, que conta a quantidade de itens de cada instituição, e o `plt.pie(...)`, que gera o gráfico de pizza. Outros comandos são necessários para estilizar as *labels* e a legenda, além de adicionar um título ao gráfico (`plt.title("Distribuição das Instituições")`).

Quadro 11 - Comandos em Python utilizando a biblioteca Matplotlib para visualizar gráficos pizza.

```

# Importação das bibliotecas necessárias
import pandas as pd
import matplotlib.pyplot as plt

# Contagem das instituições
contagem_instituicoes = df['institutions'].value_counts()

# Filtra os itens com porcentagem acima do limite mínimo
itens_labels = contagem_instituicoes[contagem_instituicoes
>= 19]

# Criação do gráfico de pizza
labels = [f'{label} ({contagem_instituicoes[label]})'
if label in itens_labels else '' for label in contagem_
instituicoes.index]
plt.pie(contagem_instituicoes, labels=labels,
autopct=lambda pct: f'{pct:.1f}%' if pct > 3.5 else '')

# Adiciona um título ao gráfico
plt.title("Distribuição das Instituições")

# Função de formatação condicional para os rótulos da
legenda
def format_legend(label):
    return f'{label} ({contagem_instituicoes[label]})'

# Criação da legenda com duas colunas e formatação
condicional nos rótulos
legend_labels = [format_legend(label) for label in
contagem_instituicoes.index]
plt.legend(labels=legend_labels, loc='best', ncol=3, bbox_
to_anchor=(1.1, 1.25))

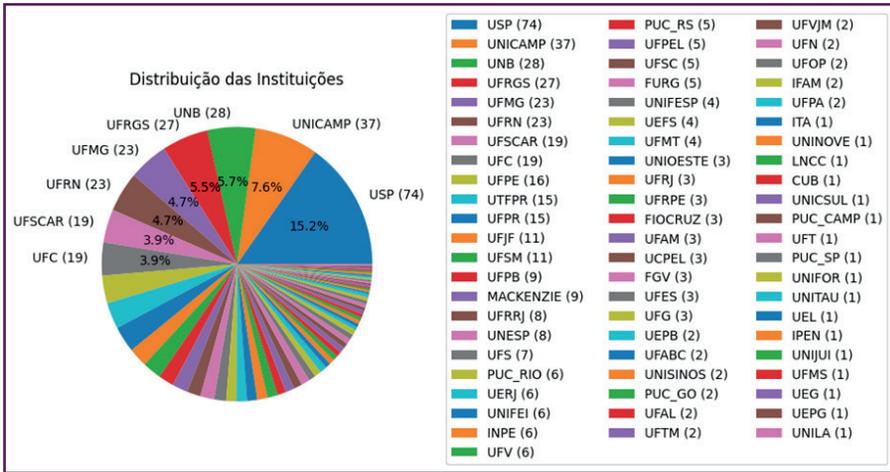
# Exibe o gráfico
plt.show()

```

Fonte: Elaborado pelos autores (2023).

Por fim, o gráfico é exibido na tela utilizando o comando `plt.show()`.

Figura 2 - Apresentação do gráfico de pizza obtido pelo comando Python.



Fonte: Elaborado pelos autores (2023).

No próximo exemplo, usamos o gráfico de barra da biblioteca Matplotlib para mostrar a quantidade de itens publicados por ano. Primeiro, calculamos os itens por ano utilizando o método `value_counts()` e ordenamos esse objeto com o método `sort_index()` para garantir que os anos estejam em ordem crescente. Por fim, criamos o gráfico de barras, usando a função `plt.bar()`, passando os anos filtrados como valores do eixo x e as quantidades de itens correspondentes como valores do eixo y. Adicionamos rótulos aos eixos x e y, e um título ao gráfico.

Quadro 12 - Comandos em Python utilizando a biblioteca Matplotlib para visualizar gráficos em barras.

```
# Importação das bibliotecas necessárias
import pandas as pd
import matplotlib.pyplot as plt

# Contagem dos itens por ano
contagem_por_ano = df['publicationDates'].value_counts().
sort_index()

# Criar o gráfico de barras
plt.bar(contagem_por_ano.index, contagem_por_ano.values)

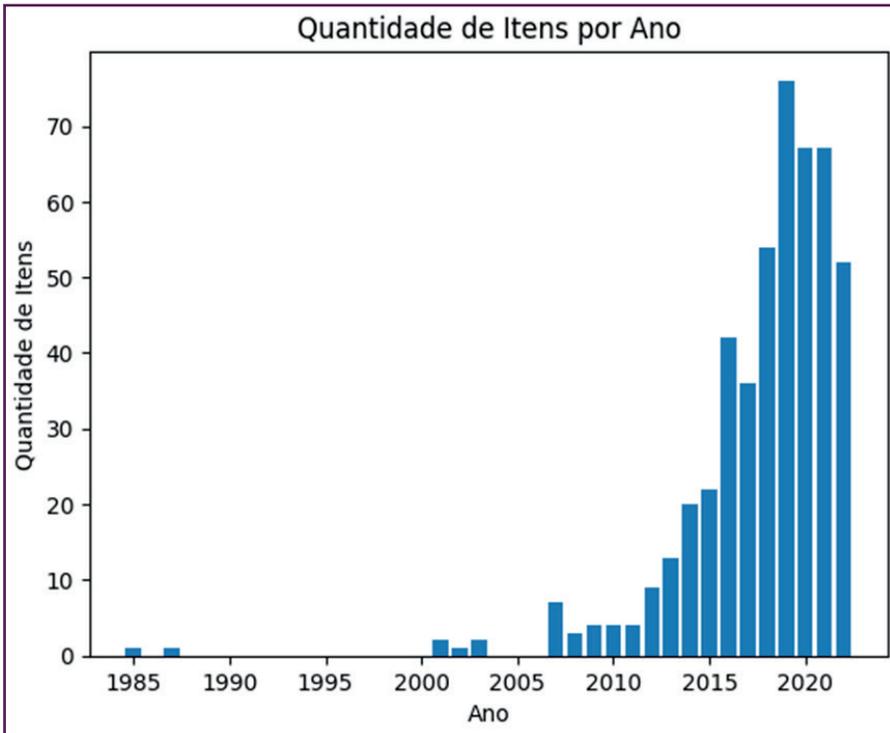
# Adicionar rótulos aos eixos x e y, e um título ao gráfico
plt.xlabel('Ano')
plt.ylabel('Quantidade de Itens')
plt.title('Quantidade de Itens por Ano')

# Exibir o gráfico
plt.show()
```

Fonte: Elaborado pelos autores (2023).

Por fim, o gráfico é exibido na tela utilizando o comando `plt.show()`.

Figura 3 - Apresentação do gráfico em barras obtido pelo comando *Python*.



Fonte: Elaborado pelos autores (2023).

O *NumPy* é uma biblioteca para computação numérica em *Python*. Ela é amplamente utilizada em ciência de dados, *machine learning* e outras áreas, nas quais é necessário lidar com cálculos matemáticos complexos. No exemplo a seguir, a biblioteca *NumPy* é utilizada para criar um *array* com valores aleatórios, por meio do método `np.random.rand(5)`. Em seguida, é calculada a média dos valores desse *array* utilizando o método `np.mean(a)`.

Quadro 13 - Comandos em *Python* utilizando a biblioteca *NumPy* para cálculos matemáticos.

```
# Importação das bibliotecas necessárias
import numpy as np

# Cria um array NumPy com valores aleatórios
a = np.random.rand(5)

# Calcula a média dos valores do array
media = np.mean(a)

# Imprime a média
print(media)
```

Fonte: Elaborado pelos autores (2023).

Por fim, a média é impressa na tela através do comando *print(media)*.

Quadro 14 - Apresentação dos resultados obtidos pelo comando *Python*.

```
0.33580568767017327
```

Fonte: Elaborado pelos autores (2023).

A visualização padrão do *Jupyter Notebook* é a seguinte:

Figura 4 - Visualização dos comandos Python na plataforma Jupyter Notebook

```

Narrativas Computacionais

Exemplo do uso da biblioteca pandas

Para utilizar o pandas, é necessário importar a biblioteca. Para isso, basta executar o seguinte código:

[ ] import pandas as pd

Para carregar o arquivo, ele deve estar disponível para acesso pelo Notebook

[ ] # Carregar o arquivo CSV em um DataFrame
df = pd.read_csv("search_results.csv", sep=";")

Mostrar as primeiras linhas do DataFrame utilizando o método head()

[ ] # Mostrar as primeiras linhas do DataFrame
print(df.head())

      id \
0      UFG_ea73bfca9fe38376aae3be46a2568b0d
1      UERJ_43e004040d52178c814f82ad4d955707
2      UTFPR-12_b60bcf0310320d0e5598b03d5743a61f
3      UNICAMP-30_98833fa1affb96f3147210dd3a509f55
4      UTFPR-1_6649cb129c518d83768831532924f477

      title \
0      Aritmética com Python
1      Retroanálise com o uso de rotina em Python apl...
2      O ensino de matemática por meio da linguagem d...
3      Desenvolvimento de uma plataforma multimídia u...
4      O ensino de matemática por meio da linguagem d...

      authors \
0      primary[Cavalcante, Rogério da Silva(NA)]
1      primary[Santiago, Carlos Alexandre de Almeida(...
2      primary[Presente, Guilherme Moraes(http://latte......
3      primary[Gonçalves Neto, Jahyr, 1980-(NA)]
4      primary[Presente, Guilherme Moraes(http://latte......

```

Fonte: Captura de tela (2023).

Esses exemplos são apenas o básico do que é possível fazer com as bibliotecas. O *Pandas*, *Numpy* e *Matplotlib* são amplamente utilizados na análise e visualização de dados, além de possuírem muitas funções e recursos para ajudar a tornar a análise mais eficiente e produtiva, especialmente quando acoplados ao *Jupyter Notebook*.

1.3 CONSIDERAÇÕES FINAIS

O processamento de dados em pesquisa requer, inevitavelmente, ferramentas informatizadas, principalmente pela quantidade. Mesmo nas ciências humanas e sociais, os estudos quantitativos ou mistos não estão totalmente descartados. Em muitos casos, os estudos qualitativos e quantitativos unidos oferecem a melhor das perspectivas, isto é, a precisão do quantitativo com a profundidade qualitativa. Assim, a presença dessas ferramentas se torna essencial para a execução do estudo.

Além disso, adotar narrativas computacionais na metodologia ajuda na apresentação da lógica utilizada no desenvolvimento dos programas de processamento, a fim de possibilitar seu reúso. Nas ciências, a reprodutibilidade da metodologia faz parte da base fundamental da criação do conhecimento, validando o estudo por meio de análise da reprodutibilidade, no sentido de serem alcançados resultados confiáveis.

REFERÊNCIA

CLANDININ, D. Jean; CONNELLY, F. Michael. **Pesquisa narrativa**: experiência e história em pesquisa qualitativa. Tradução do Grupo de Pesquisa Narrativa e Educação de Professores ILEEI/UFU. Uberlândia, MG: EDUFU, 2011. 250 p.

JUPYTER NOTEBOOK. Documentation. **The Jupyter Notebook**. 2023a. Disponível em: <https://jupyter-notebook.readthedocs.io/en/latest/notebook.html>. Acesso em: 20 set. 2023.

JUPYTER NOTEBOOK. **Project Jupyter**. 2023b. Disponível em: <https://jupyter.org>. Acesso em: 20 set. 2023.

KEARNEY, Richard. Narrativa. **Educação & Realidade**, Porto Alegre, v. 37, n. 2, 2012. Disponível em: <https://seer.ufrgs.br/educacaoerealidade/article/view/30354>. Acesso em: 19 set. 2023.

MARIANI, Fábio; MATTOS, Magda. Pesquisa narrativa: experiência e história em pesquisa qualitativa. **Revista de Educação Pública**, Cuiabá, v.

21, n. 47, p. 663-667, 10 jul. 2012. DOI 10.29286/rep.v21i47.1766. Disponível em: <https://periodicoscientificos.ufmt.br/ojs/index.php/educacaopublica/article/view/1766>. Acesso em: 19 set. 2023.

MATPLOTLIB. **Matplotlib**: Visualization with Python. 2023. Disponível em: <https://matplotlib.org/>. Acesso em: 20 set. 2023.

NUMPY. **NumPy**: the fundamental package for scientific computing with Python. 2023. Disponível em: <https://numpy.org/>. Acesso em: 20 set. 2023.

ONTAÑÓN, Santiago; ZHU, Jichen. On the role of domain knowledge in analogy-based story generation. *In*: TWENTY-SECOND INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, Barcelona, 16-22 July 2011. **Proceedings** [...]. Barcelona: AAAI Press, 2011. p. 1717-1722. Disponível em: <https://dl.acm.org/doi/10.5555/2283516.2283681>. Acesso em: 19 set. 2023.

PAIVA, Vera Lúcia Menezes de Oliveira e. A pesquisa narrativa: uma introdução. **Revista Brasileira de Linguística Aplicada**, Belo Horizonte, v. 8, n. 2, 2008. DOI 10.1590/18984-63982008000200001. Disponível em: <https://www.redalyc.org/articulo.oa?id=339829603001>. Acesso em: 19 mar. 2023.

PANDAS. **Pandas**: version 2.1.1. 2023. Disponível em: <https://pandas.pydata.org/>. Acesso em: 20 set. 2023.

RIEDL, Mark O. **Computational Narrative Intelligence**: A Human-Centered Goal for Artificial Intelligence. 21 Feb. 2016. Disponível em: <http://arxiv.org/abs/1602.06484>. Acesso em: 19 set. 2023.

RULE, Adam; TABARD, Aurélien; HOLLAN, James D. Exploration and Explanation in Computational Notebooks. *In*: CHI '18: CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 19 Apr. 2018. **Proceedings** [...]. Montreal: ACM, 2018. p. 1-12. DOI 10.1145/3173574.3173606. Disponível em: <https://dl.acm.org/doi/10.1145/3173574.3173606>. Acesso em: 20 set. 2023.

SQUIRE, Corinne. O que é narrativa? **Civitas**: Revista de Ciências Sociais, Porto Alegre, v. 14, n. 2, p. 272-284, jun. 2014. DOI 10.15448/1984-7289.2014.2.17148. Disponível em: <https://revistaseletronicas.pucrs.br/ojs/index.php/civitas/article/view/17148>. Acesso em: 19 set. 2023.

VALLS-VARGAS, Josep; ZHU, Jichen; ONTAÑÓN, Santiago. From computational narrative analysis to generation: a preliminary review. *In: FDG'17: INTERNATIONAL CONFERENCE ON THE FOUNDATIONS OF DIGITAL GAMES 2017*, Hyannis, 2017. **Proceedings** [...]. Hyannis, MA: ACM, 2017. p. 1-4. DOI 10.1145/3102071.3106362. Disponível em: <https://dl.acm.org/doi/10.1145/3102071.3106362>. Acesso em: 20 set. 2023.

DADOS DOS AUTORES:

Milton Shintaku



Milton Shintaku é Doutor em Ciência da Informação pela Universidade de Brasília. Coordenador de Tecnologia para Informação (Cotec) do Instituto Brasileiro de Informação em Ciência e Tecnologia (Ibict).

<https://orcid.org/0000-0002-6476-4953>

shintaku@ibict.br

Rebeca dos Santos de Moura



Rebeca dos Santos de Moura é Mestre em Engenharia de Sistemas Eletrônicos e de Automação e Bacharel em Engenharia da Computação pela Universidade de Brasília (UnB). Desenvolvedora e assistente de pesquisa no Instituto Brasileiro de Informação em Ciência e Tecnologia (Ibict).

<https://orcid.org/0000-0002-7685-8826>

becahp@gmail.com

Lucas Rodrigues Costa



Lucas Rodrigues Costa é Doutor em Ciência da Computação pela Universidade de Brasília (UnB), professor substituto da UnB, desenvolvedor e assistente de pesquisa no Instituto Brasileiro de Informação em Ciência e Tecnologia (Ibict).

<https://orcid.org/0000-0002-0973-4866>

lucasrodrigues@ibict.br

Como referenciar o capítulo 1:

SHINTAKU, Milton; MOURA, Rebeca dos Santos de; COSTA, Lucas Rodrigues. Narrativa computacional com Jupyter Notebook como apoio à pesquisa nas ciências sociais aplicadas. *In*: SHINTAKU, Milton; MACÊDO, Diego José; MARIN, Luciano Heitor Gallegos (org.). **Tecnologias utilizadas em pesquisas acadêmicas em Ciências Sociais Aplicadas**. Brasília, DF: Ibict, 2023. cap. 1. p. 15-37. ISBN 978-65-89167-94-5. DOI: <http://doi.org/10.22477/9786589167938cap1>.