# PARTE 1: Data Collection in the Social Sciences

“O volume de informações disponíveis para a pesquisa tem crescido rapidamente nas últimas décadas” (BALTAR; BALTAR, 2013, p. 11)

“Com o crescente incremento na capacidade de armazenamento de informação, aliada ao aumento na velocidade do processamento, contando ainda com a ampliação nos canais de comunicação entre os dispositivos digitais, cada vez mais diferentes atividades humanas passaram a ter um registro digital em tempo real”(BALTAR; BALTAR, 2013, p. 11–12)

“With the emergence of increasingly more powerful computers, along with most creative data processing software, all scientific disciplines gained access to historically unprecedented and unfathomably detailed information on micro and macro-level human interactions.” (ÜNVER, 2019, p. 159–160)

Dois tipos de dados principais: Digital Trace Data e dados públicos via Política de Dados Abertos

[ Digital Trace Data ]

[ Política de Dados Abertos ]

“Todas as entidades federais, estaduais e municipais, desde 2011, têm a obrigatoriedade de se adequarem a Lei Federal n.12527 que estabelece as normas para o acesso à informação no Brasil. A publicação da ‘Lei do Acesso à Informação’ é um elemento importante para fomentar a participação democrática a partir de uma maior simetria na informação entre os diversos atores sociais, sobretudo em relação com o poder público” (BALTAR; BALTAR, 2013, p. 14)

“O conjunto de dados disponíveis por órgãos públicos têm aumentado desde 2011. Uma iniciativa do Governo Federal, chamada “Dados Abertos”, reúne com facilidade e de maneira indexada informações de diversas pesquisas elaboradas pelo IBGE, TEM, Ministério do Planejamento...informações que dez anos atrás encontravam dispersas, não catalogadas e acessíveis a poucos funcionários públicos, agora estão cada vez mais abertas ao público” (BALTAR; BALTAR, 2013, p. 14)

“Somam-se a este conjunto, os acervos documentais que estão sendo digitalizados pela Biblioteca Nacional, o acervo audiovisual e documental do Centro de Pesquisa e Documentação Histórica da FGV, entre outros” (BALTAR; BALTAR, 2013, p. 14)

“No âmbito internacional, há também um conjunto cada vez maior de fontes de dados sistematizadas e de acesso público, como o UNDATA das Nações Unidas, o World Bank Data...o grande acervo da OIT, entre outros.” (BALTAR; BALTAR, 2013, p. 14)

“São inúmeros exemplos de acervos de dados sistematizados, organizados e disponíveis para pesquisa social” (BALTAR; BALTAR, 2013, p. 15)

“...as Ciências Sociais têm um papel importante para apropriação qualificada e sistematizada dos dados disponibilizados para o entendimento de questões sociais de relevo” (BALTAR; BALTAR, 2013, p. 14)

“É necessário um esforço coordenado e colaborativo para que se organize e analise esse volume de dados, aos quais se podem acrescentar uma grande quantidade de registros de estudos de casos, observações, surveys e coletas de documentos oriundas de pesquisadores individuais ou de grupos de pesquisas temáticos” (BALTAR; BALTAR, 2013, p. 15)

“Although the quantity and granularity of digital data produced every day is impressive, a key question remains how to process such data in a meaningful way and how to build social theory using it” (ÜNVER, 2019, p. 159)

“O desafio para as Ciências Sociais está em ter infraestrutura para coletar e armazenar esse volume de dados, ter recursos de tecnologia da informação para processar estes dados, dispor de métodos analíticos que possam oferecer respostas nítidas aos problemas da complexidade social e propor políticas e indicadores de monitoramento para intervir sobre mudanças sociais” (BALTAR; BALTAR, 2013, p. 18).

“O acesso e o tratamento analítico dessas informações tem impulsionado a renovação e ampliação dos métodos nas Ciências Sociais” (BALTAR; BALTAR, 2013, p. 12).

“No brasil e em grande parte da América Latina ainda nos encontramos com alguma defasagem em relação a estes debates interdisciplinares que têm movimentado as Ciências Sociais em outros países. Também temos estado aquém da capacitação necessária para o uso de ferramentas de informática e manipulação de bases de dados em pesquisa social, sejam em termos de pesquisa quantitativa, qualitativa, documental, ou qualquer outro tipo”. (BALTAR; BALTAR, 2013, p. 13)

“O objetivo da Ciência Social Computacional seria o de aliar o volume de dados (que chegará a quantia de Zettabytes sob a forma de informação quantitativa, qualitativa ou de qualquer natureza) com métodos analíticos (análise quantitativa, análise qualitativa, modelagem e simulação) para desvendar conexões, ou seja, a complexidade das relações sociais” (BALTAR; BALTAR, 2013, p. 17–18)

“Como as Ciências Sociais estão se preparando para os desafios conceituais e metodológicos da era Zettabyte ?” (BALTAR; BALTAR, 2013, p. 12)

“Computational International Relations (ComInt), introduced as a specific inquiry of research in this paper, derives from the Computational Social Science (ComSoc) revolution of the last decade” (ÜNVER, 2019, p. 157)

“Computational social science as a metabridge between extreme ends of hard sciences and social sciences. (…). It is rather a meeting point between diverse disciplines that seek to strengthen their analytical approach through the use of a wide array of computer and data science tools” (ÜNVER, 2019, p. 157)

“Current definitions of ComSoc therefore distinguish between computer-based social science, which is using computer programs to process quantitative social data and ComSoc, which processes enormous chunks of - often real-time - Internet data” (ÜNVER, 2019, p. 159)

“Related to, but separate from ComSoc, ComInt deals exclusively with core IR topics of power, conflict/peace, state behavior, international norms/institutions and the world system/order” (ÜNVER, 2019, p. 160)

“As ComInt starts dealing with non-state actors (NGOs, MNCs, media, religious groups, Diasporas, militants etc.) it steers further into the domain of sociology and shares common ground with digital, or tech sociologists.” (ÜNVER, 2019, p. 160)

“This domain requires even further novel methods, as tracing the transient shifts and trends of non-state actors require a way to bring ethnography close to the field of computational methods that both include, but also expand upon the existing approaches of digital and/or Internet ethnography” (ÜNVER, 2019, p. 160)

“Defined in simple terms ComInt, relies on the mining and processing of vast quantities of digital social footprint to study, model and explain world events.” (ÜNVER, 2019, p. 160)

“ComInt predominantly (but not exclusively) uses large chunks of digital footprint and focuses on social online activities that generate enormous quantities of social data” (ÜNVER, 2019, p. 160)

“This is one of the reasons why ComInt or ComSoc didn’t exist a decade ago, and also a reason why merely using numerical analysis software like R, Python and MatLab to model existing quantitative data, isn’t really ComSoc or ComInt” (ÜNVER, 2019, p. 160)

“The origin, size and type of data that is collected make the main difference, as well as the main focus of study; the Internet and digital interdependencies.” (ÜNVER, 2019, p. 160)

# PARTE 2: Data Collection Techniques

[ Web Scrapping ]

- Páginas Estáticas

**HTML**: “For browsing the Web, there is a hidden standard behind the scenes that structures how information is displayed—the Hypertext Markup Language or HTML” (MUNZERT *et al.*, 2015, p. 10)

“HTML is a language for presenting content on the Web that was first proposed by Tim Berners-Lee (1989). The standard has continuously evolved since the initial introduction, the most recent incarnation is HTML5 that is being developed by the World Wide Web Consor- tium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).” (MUNZERT *et al.*, 2015, p. 18)

“An HTML file is basically nothing but plain text—it can be opened and edited with any text editor. What makes HTML so powerful is its marked up structure. HTML” (MUNZERT *et al.*, 2015, p. 18)

“The markup definitions rely on predefined character sequences—the tags—that enclose parts of the text. Markup tells browsers how the document is structured and the function of its various parts.” (MUNZERT *et al.*, 2015, p. 18)

“there is a difference between the way data are presented in a browser on the one side and how they are stored within the HTML code on the other. (…). a basic understanding of HTML and the way it stores information is indispensable.” (MUNZERT *et al.*, 2015, p. 10)

“What you see in your browser is therefore not the HTML document itself but an interpretation of it.(…)” (MUNZERT *et al.*, 2015, p. 18)

“Plain text is turned into an HTML document by tags that can be interpreted by a browser. They can be thought of as named braces that enclose content and define its structural function. (…). The combination of start tag, content, and end tag is called element” (MUNZERT *et al.*, 2015, p. 20)

“Start tags and end tags are also known as opening and closing tags. Tags are always enclosed by < and > to distinguish them from the content. Start and end tags carry the same name, but the end tag is preceded by a slash /.”(MUNZERT *et al.*, 2015, p. 20)

“Although it is recommended that each element has a start and an end tag, this is not common practice for all types of elements. For example, the <br> tag indicates a line break and is not closed by a </br> counterpart” (MUNZERT *et al.*, 2015, p. 20)

<a>: “The anchor tag <a> is what turns HTML from just a markup language into a hypertext markup language by enabling HTML documents to link to other documents. Much of the site-to-site navigation in browsers works via anchor elements” (MUNZERT *et al.*, 2015, p. 24)

“We often find ourselves in situations where we want to extract information not from asingle page but from a whole series of pages. If we are lucky, the pages are listed on an index page. More frequently, however, we have to collect links from one page that points to the next page, which points to the next page, and so on. In both cases the information we are looking for—the location of another page—is stored in an <a> element” (MUNZERT *et al.*, 2015, p. 24)

<p>: “The <p> tag labels its content as being a paragraph and ensures that line breaks are inserted <p> before and after its content” (MUNZERT *et al.*, 2015, p. 27)

<h1>, <h2>, <h3>…: “In order to define different levels of headlines—level 1 to level 6—HTML provides a series <h1>, <h2>,… of tags <h1>, <h2>,…down to <h6>” (MUNZERT *et al.*, 2015, p. 27)

<ul>, <ol>,and <dl>: “Several tags exist to list content. They are used depending on whether they wrap around an ordered list (<ol>), an unordered list (ul), or a description list (<dl>). The former two tags make use of nested <li> elements to define list items, while the latter needs two further elements: <dt> for keyword and <dd> for its description” (MUNZERT *et al.*, 2015, p. 27)

<div> and <span>: “While <div> and <span> themselves do not change the appearance of the content they enclose, these tags are used to group parts of the document—the former is used to define groups across lines, tags, and paragraphs, while the latter is used for in-line grouping.” (MUNZERT *et al.*, 2015, p. 27)

<table>, <tr>, <td>,and <th>: “The next group of elements enables HTML to display tables. (…). To begin a table we make use of <table>.We start new lines with <tr>. Within <tr>, we can either use <td> for defining cells or <th> for header cells.” (MUNZERT *et al.*, 2015, p. 32)

Tree Structure: “A good way to describe the multiple layers of an HTML document is the tree analogy. (…). Elements need to be strictly nested within each other in a well-formed and valid HTML file. A pair of start and end tags has to be completely enclosed by another pair of tags.” (MUNZERT *et al.*, 2015, p. 21)

[ Attributes.. ]

CSS: “Grouping parts of an HTML document is handy when combined with Cascading Style Sheets (CSS), a language for describing the layout of HTML and other markup documents like XML, SVG, and XHTML” (MUNZERT *et al.*, 2015, p. 28)

“Style definitions are commonly stored in separate CSS files. (…). Later in the document they are passed to an element using an additional class [ and id ] attribute” (MUNZERT *et al.*, 2015, p. 28)

“The purpose of CSS is to separate content from layout to improve the document’s accessibility. Defining styles outside of an HTML and assigning them via the class attribute enables the web designer to reuse styles across elements and documents. This enables developers to change a style in one single place—within the CSS file—with effects on all elements and documents using this style” (MUNZERT *et al.*, 2015, p. 28)

“So why should we care about style? First of all, one should always care about style.But second, as CSS is so handy for developers, <div>, <span>, and class tags are used frequently. They thus provide structure to the HTML document that we can make use of to identify where our desired information is stored.” (MUNZERT *et al.*, 2015, p. 28)

[ Ok, then we should parse ]

“While performing web scraping, we usually get in touch with HTML in two steps: First,we inspect content on the Web and examine whether it is attractive for further analyses” (MUNZERT *et al.*, 2015, p. 33)

“Second, we import HTML files into R and extract information from them” (MUNZERT *et al.*, 2015, p. 33)

Third, “we identify the relevant information that is stored in an XML/HTML document and is therefore accessible with XPath queries” (MUNZERT *et al.*, 2015, p. 274)

“To identify which parts of the source code correspond to which elements in the browser window and vice versa, we can use an element inspector, which is implemented in most browsers.” *(MUNZERT et al., 2015, p. 18)*

“Next, we extract the actual information in step ④ by developing one or more XPath queries” (MUNZERT *et al.*, 2015, p. 274)

“In order to take full advantage of the Web as a nearly endless data source, we have to perform a series of filtering and extraction steps once the relevant web documents have been identified and downloaded”. (MUNZERT *et al.*, 2015, p. 11).

“This task consists of specifying the data we are interested in and locating it in a specific document and then tailoring a query to the document that extracts the desired information” (MUNZERT *et al.*, 2015, p. 12).

**XPath:** “The first tool at our disposal is the XPath query language. It is used to select specific pieces of information from marked up documents such as HTML, XML or any variant of it, for example SVG or RSS” (MUNZERT *et al.*, 2015, p. 11).

- Páginas Dinâmicas

“HTML itself is not a programming language. HTML is a markup language that describes content and defines its presentation. Once an HTML file is loaded in the browser, it remains stable and does not change by events or user interaction” (MUNZERT *et al.*, 2015, p. 30).

“you find that the HTML/HTTP infrastructure implies a rather static display of content in a page layout, which is retrieved through sequential, iterative requests initiated by the user. The inherent inflexibility ofHTML/HTTP is most apparent in its inability to create more dynamic displays of information, such as we are used to from standard desktop applications” (MUNZERT *et al.*, 2015, p. 149).

“Because HTML/HTTP is technically unable to provide any of the above features, aseries of additional web technologies have found their way into the toolkit of modern web developers over the last 15 years. A prominent role in this transformation is assumed by a group of technologies that are subsumed under the term AJAX, short for “Asynchronous JavaScript and XML.” (MUNZERT *et al.*, 2015, p. 149).

“The JavaScript programming language has a prominent role in the group of AJAX technolo- gies. Developed by Brendan Eich at Netscape in 1995, JavaScript is a complete, high-level programming language (Crockford 2008). What sets JavaScript apart from other languages is its seamless integration with other web technologies (e.g., HTML, CSS, DOM) as well as its support by all modern browsers, which contain powerful engines to interpret JavaScript code.” (MUNZERT *et al.*, 2015, p. 150).

“Although AJAX-enriched websites provide tremendous advantages from a user perspective, they create difficulties for our efforts to automatically gather web data.” (MUNZERT *et al.*, 2015, p. 150).

“This is so because AJAX-enriched webpages constitute a significant departure from the static HTML/HTTP site model in which a HTTP-requested webpage is displayed equally for all users and all infor- mation that is displayed on screen is delivered upfront”

“This presents a serious obstacle to analysts who care to collect web data since a simple HTTP GET request may not suffice if information is loaded iteratively only after the site has been requested.”

**Selenium:** “As a powerful alternative to initiating web requests from the R console, we present the Selenium framework as a hands-on approach to getting a grip on web data” (MUNZERT *et al.*, 2015, p. 12).

“Selenium/Webdriver is an open-source suite ofsoftware with the primary purpose ofproviding a coherent, cross-platform framework for testing applications that run natively in the browser.” (MUNZERT *et al.*, 2015, p. 253).

“In the development of web applications, testing is a necessary step to establish expected func- tionality ofthe application, minimize potential security and accessibility issues, and guarantee reliability under increased user traffic” (MUNZERT *et al.*, 2015, p. 253).

“Before the creation of Selenium this kind of testing had been carried out manually—a tedious and error-prone undertaking. Selenium solves this problem by providing drivers to control browser behavior such as clicks, scrolls, swipes, and text inputs.” (MUNZERT *et al.*, 2015, p. 253).

“This enables programmatic approaches to the problem by using a scripting language to characterize sequences of user behaviors and report if the application fails.” (MUNZERT *et al.*, 2015, p. 253).

“Selenium’s capability to drive interactions with the webpage through the browser is of Installing the more general use besides testing purposes. Since it allows to remote-control the browser, can work with and request information directly from the live DOM tree, that is, how the visual package display is presented in the browser window.” (MUNZERT *et al.*, 2015, p. 253).

“Selenium allows us to direct commands to a browser window, such as mouse clicks or keyboard inputs, via R. By working directly in the browser, Selenium is capable of circumventing some of the problems discussed with AJAX-enriched webpages” (MUNZERT *et al.*, 2015, p. 12).

API

“Uma [interface de programação de aplicações (API)](https://www.redhat.com/pt-br/topics/api) é um conjunto de definições e protocolos para criar e integrar softwares de aplicações.”

“Às vezes, as APIs são referidas como um contrato entre um provedor e um usuário de informações, estabelecendo o conteúdo exigido pelo consumidor (a chamada) e o conteúdo exigido pelo produtor (a resposta).”

“Em outras palavras, ao interagir com um computador ou sistema para recuperar informações ou executar uma função, a API ajudará a comunicar o que você quer ao sistema para que ele entenda e realize o que foi solicitado.”

“Pense nas APIs como um mediador entre os usuários ou clientes e os recursos que eles querem obter. As APIs também servem para que organizações compartilhem recursos e informações e, ao mesmo tempo, mantenham a segurança e o controle, pois permitem determinar quem tem acesso e o que pode ser acessado. “

“At a high level, APIs are the way computer programs “talk” to each other so that they can request and deliver information. “

“This is done by allowing a software application to call what's known as an **endpoint**: an address that corresponds with a specific type of information we provide (endpoints are generally unique like phone numbers).”

“When we talk about APIs in this book, we refer to web services or (web) APIs, that is interfaces with web applications. We treat the terms “API” and “web service” synonymously, APIs although the term API encompasses a much larger body of software.” (MUNZERT *et al.*, 2015, p. 259)

“The reason why APIs are of importance for web data collection tasks is that more and more web applications offer APIs which allow retrieving and processing data. With the rise ofWeb 2.0, where web APIs provided the basis for many applications, application providers recognized that data on the Web are interesting for many web developers.” (MUNZERT *et al.*, 2015, p. 259)

“The general logic of retrieving data from web APIs is simple The API provider sets up a service that grants access to data from the application or to the application itself. The API user accesses the API to gather data or communicate with the application” (MUNZERT *et al.*, 2015, p. 259).

“APIs provide data in various formats” (MUNZERT *et al.*, 2015, p. 259).

“JSON has probably become the most popular data exchange format of modern web APIs, but XML is still frequently used, and any other formats such as HTML, images, CSVs, and binary data files are possible” (MUNZERT *et al.*, 2015, p. 259–260).

**JSON**: “Another standard data storage and exchange format that is frequently encountered on the Web is the JavaScript Object Notation or JSON. Like XML, JSON is used by many web applications to provide data for web developers.” (MUNZERT *et al.*, 2015, p. 10).

“In contrast to HTML or XML documents, JSON documents are more lightweight and easier to parse. To extract data from JSON, we do not draw upon a specific query language, but rely on high-level R functionality, which does a good job in decoding JSON data” (MUNZERT *et al.*, 2015, p. 12).

**XML**: “The Extensible Markup Language or XML is one of the most popular formats for exchanging data over the Web. It is related to HTML in that both are markup languages. However, while HTML is used to shape the display of information, the main purpose of XML is to store data.” (MUNZERT *et al.*, 2015, p. 10)

“Thus, HTML documents are interpreted and transformed into pretty-looking output by browsers, whereas XML is “just” data wrapped in user-defined tags. The user-defined tags make XML much more flexible for storing data than HTML.” (MUNZERT *et al.*, 2015, p. 10).

“There is little doubt that gathering data from APIs/web services is the gold standard for web data collection. Scraping data from HTML websites is often a difficult endeavor. We first have to identify in which slots of the HTML tree the relevant data are stored and how to get rid of everything else that is not needed” (MUNZERT *et al.*, 2015, p. 276).

“APIs provide exactly the information we need, without any redundant information. They standardize the process of data dissemination, but also retain control for the provider over who accesses what data.” (MUNZERT *et al.*, 2015, p. 276).

“Web services allow providing standardized formats that most programming languages can deal with” (MUNZERT *et al.*, 2015, p. 276).

“The advantages of web services over the other techniques stem from the fact that tapping APIs is in fact not web scraping. Many of the disadvantages of screen scraping, malformed HTML, other robustness to legal issues, do not apply to data collection with web services. As a result, we can draw upon clean data structures and have higher trust in the collection outcomes.” (MUNZERT *et al.*, 2015, p. 277).

“APIs are implemented for developers and thus must be understandable to humans. Therefore, an extensive documentation of features, functions, and parameters is often part of an API. It gives programmers an overview of the content and form of information an API provides, and what information it expects, for example, via queries” (MUNZERT *et al.*, 2015, p. 260).

“Commercial APIs can be very restrictive or offer no data at all if you do not pay a monthly fee, so you should find out early what you get for which payment. And do not invest time for nothing—not all web services are well-maintained. Before you start to program wrappers around an existing API, check whether the API is regularly updated.” (MUNZERT *et al.*, 2015, p. 276).

“As the focus of web services is on the delivery of data, not layout, our code is generally

more robust. Web services usually satisfy a certain demand and we are often not the only ones interested in the data. If many people create interfaces to the API from various programming environments, we can benefit from this “wisdom of the crowds” and adding robustness to our code” (MUNZERT *et al.*, 2015, p. 276).

“Although there are not many general disadvantages of using APIs for automated data

“It is important to note that in order to tap web services with R, we often do not have to have any deeper knowledge of these techniques—either because others have already programmed a handy interface to these APIs or because our knowledge about HTTP, XML, and JSON suffices to understand the documentation of an API and to retrieve the information we are looking for” (MUNZERT *et al.*, 2015, p. 260).

“Wrappers are pieces of software which wrap around existing software—in our case to be able to use R functions to call an API and to make the data we retrieve from an API accessible for further work in R” (MUNZERT *et al.*, 2015, p. 276).

“It may be necessary to write wrapper software for convenient data exchange with the web service. Wrappers are functions that handle details of API access and data transformation, for example, from JSON to R objects” (MUNZERT *et al.*, 2015, p. 259).

“We could also construct convenient wrapper functions for the API calls. Interfaces for RPackages exist for some web services which offer convenience functions to pass R objects to the API and get back R objects. Such functions are not too difficult to create once you are familiar with an API’s logic and the data technology that is returned.” (MUNZERT *et al.*, 2015, p. 263).

“There are several API standards or styles, the more popular ones being REST and SOAP.” (MUNZERT *et al.*, 2015, p. 260).

“The modus operandi of APIs varies— we shortly discuss the popular standards REST and SOAP further below” (MUNZERT *et al.*, 2015, p. 259).

[ Rest ]

“API REST, também chamada de API RESTful, é uma interface de programação de aplicações que segue conformidade com as restrições da arquitetura REST. A sigla REST significa Representational State Transfer”

“REST stands for Representational State Transfer (Fielding 2000). The core idea behind REST is that resources are referenced (e.g., via URLs) and representations of these resources are exchanged. Representations are actual documents like an HTML, XML, or JSON file.” (MUNZERT *et al.*, 2015, p. 260).

“Quando uma solicitação é feita por meio de uma API RESTful, essa API transfere uma representação do estado do recurso ao solicitante. Essa informação, ou representação, é fornecida utilizando um dos vários formatos possíveis via HTTP: Javascript Object Notation (JSON), HTML, XLT ou texto simples. O formato JSON é o mais usado porque, apesar de seu nome, é independente de qualquer linguagem e pode ser lido por máquinas e humanos.”

“A RESTful API is an API that confirms to the REST constraints. The constraints include the existence of a base URL to which query parameters can be added, a certain representation (JSON, XML,...), and the use of standard HTTP methods.” (MUNZERT *et al.*, 2015, p. 260).

“Para que uma API seja considerada do tipo RESTful, ela precisa está em conformidade com os seguintes critérios”: “i) Ter uma arquitetura cliente/servidor formada por clientes, servidores e recursos, com solicitações gerenciadas por meio de HTTP.” “ii) Realizar comunicação cliente/servidor [stateless](https://www.redhat.com/pt-br/topics/cloud-native-apps/stateful-vs-stateless). Isso significa que cada solicitação é separada e não conectada com outras, e que nenhuma informação do cliente é armazenada entre elas” “iii) Armazenar dados em cache para otimizar as interações entre cliente e servidor” “iv) Ter uma interface uniforme entre os componentes para que as informações sejam transferidas em um formato padronizado” “v) Ter um sistema em camadas que organiza os tipos de servidores (responsáveis pela segurança, pelo carregamento de carga e assim por diante) envolvidos na recuperação das informações solicitadas em hierarquias que o cliente não pode ver.” “vi) Possibilitar código sob demanda (opcional): a capacidade de enviar do servidor para o cliente um código executável quando solicitado, a fim de ampliar as funções do cliente. “

“The development of REST is closely linked to the design of HTTP, as the standard HTTP methods GET, POST, PUT, and DELETE are used for the transfer of representations” (MUNZERT *et al.*, 2015, p. 260).

---------------------------------------- // ------------------------------------------------

**HTTP**: “To retrieve data from the Web, we have to enable our machine to communicate with servers and web services. The lingua franca of communication on the Web is the Hypertext Transfer Protocol (HTTP). It is the most common standard for communication between web clients and servers” (MUNZERT *et al.*, 2015, p. 11)

“HTTP dates back to the late 1980s when it was invented by Tim Berners-Lee, Roy Fielding and others at the CERN near Geneva, Switzerland” (MUNZERT *et al.*, 2015, p. 101)

“Virtually every HTML page we open, every image we view in the browser, every video we watch is delivered by HTTP. Despite our continuous usage of the protocol we are mostly unaware of it as HTTP exchanges are typically performed by our machines.” (MUNZERT *et al.*, 2015, p. 11)

“We hardly ever come into direct contact with HTTP. Constructing and sending HTTP requests and processing servers’ HTTP responses are tasks that are automatically processed by our browsers and email clients” (MUNZERT *et al.*, 2015, p. 101)

“We will learn that for many of the basic web scraping applications we do not have to care much about the particulars of HTTP, as R can take over most of the necessary tasks just fine. In some instances, however, we have to dig deeper into the protocol and formulate advanced requests in order to obtain the information we are looking for” (MUNZERT *et al.*, 2015, p. 11)

“As we will see, for many of the basic web scraping tasks we still do not have to care much about the HTTP particulars in the background, as R handles this for us by default.” (MUNZERT *et al.*, 2015, p. 101).

“In some instances, however, we have to dig deeper into protocol file transfers and formulate precise requests in order to get the information we want”. (MUNZERT *et al.*, 2015, p. 101).

“To access content on the Web, we are used to typing URLs into our browser or to simply clicking on links to get from one place to another, to check our mails, to read news, or to download files. Behind this program layer that is designed for user interaction there are several more layers—techniques, standards, and protocols—that make the whole thing work.” (MUNZERT *et al.*, 2015, p. 102).

“Together they are called the Internet Protocol Suite (IPS). Two of the most prominent players of this Protocol Suite are TCP (Transmission Control Protocol) and IP (Internet Protocol). They represent the Internet layer (IP) and the transportation layer (TCP).” (MUNZERT *et al.*, 2015, p. 102).

“What is worth mentioning, however, is that TCP and IP take care of reliable data transfer between computers in the network.” (MUNZERT *et al.*, 2015, p. 102).

“On top ofthese transportation standards there are specialized message exchange protocols like HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol), Post Office Protocol (POP) for email retrieval, SMTP (Simple Mail Transfer Protocol) or IMAP (Internet Message Access Protocol) for email storage and retrieval.” (MUNZERT *et al.*, 2015, p. 102).

“All of these protocols define standard vocabulary and procedures for clients and servers to talk about specific tasks—retrieving or storing documents, files, messages, and so forth. They are subsumed under the label application layer” (MUNZERT *et al.*, 2015, p. 102).

“Other than the name suggests, HTTP is not only a standard for hypertext document retrieval. Although HTTP is quite simple, it is flexible enough to ask for nearly any kind of resource from a server and can also be used to send data to the server instead of retrieving it.” (MUNZERT *et al.*, 2015, p. 102).

“Simply put, when we access a website like www.r-datacollection.com/index.html, our browser serves as the HTTP client. The client first asks a DNS server (Domain Name System) which IP address is associated with the domain part of the URL we typed in” (MUNZERT *et al.*, 2015, p. 102–103).

“After the browser has received the IP address as response from the DNS server, it establishes a connection to the requested HTTP server via TCP/IP. Once the connection is established, client and server can exchange information—in our case by exchanging HTTP messages” (MUNZERT *et al.*, 2015, p. 103).

“The most basic HTTP conversation consists of one client request and one server response. For example, our browser asks for a specific HTML document, an image, or some other file, and the server responds by delivering the document or giving back an error code if something went wrong.” (MUNZERT *et al.*, 2015, p. 103).

“There are two important facts about HTTP that should be kept in mind. First, HTTP is not only a protocol to transport hypertext documents but is used for all kinds of resources. Second, HTTP is a stateless protocol. This means that without further effort each pair of request and response between client and server is handled by default as though the two were talking to each other for the first time no matter how often they previously exchanged information.” (MUNZERT *et al.*, 2015, p. 103).

***URL syntax***

“The location of websites and other web content are identified by Uniform Resource Locators (URLs). They are not part of HTTP but make communication via HTTP and ther protocols straightforward for users” (MUNZERT *et al.*, 2015, p. 104–105).

“The general URL format can be expressed as follows: ‘scheme://hostname:port/path?querystring#fragment’” (MUNZERT *et al.*, 2015, p. 105).

“Each URL starts with a scheme that defines the protocol that is used to communicate between client/application and server. In the example, the scheme is http, separated by a colon.” (MUNZERT *et al.*, 2015, p. 105).

“The hostname provides the name of the server where the resource of interest is stored.It is a unique identifier of a server. The hostname along with the port component tell the client at which door it has to knock in order to get access to the requested resource.” (MUNZERT *et al.*, 2015, p. 105).

“As we usually provide the human-friendly versions of URLs, the Domain Name System(DNS) translates hostnames into numerical IP addresses. Therefore, the DNS is frequently compared to a worldwide phone book that redirects users who provide hostnames to services or devices.” (MUNZERT *et al.*, 2015, p. 105).

“The path determines the location of the requested resource on the server. It works likepaths on any conventional file system where files are nested in folders that may again be nested in folders and so on. Path segments are separated by slashes (/).”(MUNZERT *et al.*, 2015, p. 105).

“In some cases, URLs provide supplementary information in the path that helps the server to process the request correctly. The additional information is delivered in query strings that hold one or more name=value pairs. The query string is separated from the rest of the URL by a question mark” (MUNZERT *et al.*, 2015, p. 105).

“Finally, fragments help point to a specific part of a document. This works well if the requested resource is HTML and the fragment identifier refers to a section, image, or similar. In the example above, the fragment #Bio requests a direct jump to the biography section of the document.” (MUNZERT *et al.*, 2015, p. 105).

***HTTP messages***

“HTTP messages, whether client requests or server response messages, consist of three parts: start line, headers, and body. While start lines differ for request and response, the messages’ header and body sections are structured identically” (MUNZERT *et al.*, 2015, p. 107).

“The start line is the first and indispensable line of each HTTP message. In requests, the start line defines the method used for the request, followed by the path to the resource requested, followed by the highest HTTP version the client can handle” (MUNZERT *et al.*, 2015, p. 107).

“When initiating HTTP client requests, we can choose among several request methods. The two most important HTTP methods are GET and POST. Both methods request a resource from the server, but differ in the usage of the body” (MUNZERT *et al.*, 2015, p. 108).

“Whereas GET does not send anything in the body of the request, POST uses the body to send data. In practice, simple requests for HTML documents and other files are usually executed with the GET method. ” (MUNZERT *et al.*, 2015, p. 108).

“The server response start line begins with a statement on the highest HTTP version the server can handle, followed by a status code, followed by a human-readable explanation of the status” (MUNZERT *et al.*, 2015, p. 107).

“When a server responds to a request, it will always send back a status code in the start line of the response. The most famous response that nearly everybody knows from browsing the Web is 404, stating that the server could not find the requested document. Status” (MUNZERT *et al.*, 2015, p. 108).

“Headers define the actions to take upon reception of a request or response. Headers can be general or belong to one specialized group: header fields for requests, header fields for responses, and header fields regarding the body of the message.” (MUNZERT *et al.*, 2015, p. 109).

“Accept is a requet header field that tells the server about the type of resources the clientis willing to accept as response” (MUNZERT *et al.*, 2015, p. 109).

“If no resource fits the restrictions made in Accept, the server should send a 406 status code. The specification of accepted content follows the MIME type scheme.” (MUNZERT *et al.*, 2015, p. 109).

“The body of an HTTP message contains the data. This might be plain text or binary data. Which type of data the body is composed of is specified in the content-type header, following the MIME type specification (Multipurpose Internet Mail Extensions)” (MUNZERT *et al.*, 2015, p. 107).

“MIME types tell the client or server which type of data it should expect. They follow a scheme of main-type/sub-type. Main types are, for example, application, audio, image, text, and video with subtypes like application/pdf, audio/mpg, audio/ogg, image/gif, image/jpeg, image/png, text/plain, text/html, text/xml, video/mp4, video/quicktime, and many more” (MUNZERT *et al.*, 2015, p. 107).

---------------------------------------- // ------------------------------------------------

“GET is the usual method when the goal is to retrieve data. To simplify matters, the difference between a GET request of a REST API and a GET request our browser puts to a server when asking for web content is that (a) parameters are often well-documented and (b) the response is simply the content, not any layout information” (MUNZERT *et al.*, 2015, p. 260).

“POST, PUT, and DELETE are methods that are implemented when the user needs to create, update, and delete content, respectively. This is useful for APIs that are connected to personal accounts, such as APIs from social media platforms like Facebook or Twitter. Finally,”

# PARTE 3: Practical Issues and Ethical Data Collection

“Whatever the level of difficulty for scraping information from the web, the circle of on the art of scraping remains almost always identical The followings tasks are part of most scraping exercises: 1. Information identification 2. Choice of strategy 3. Data retrieval 4. Information extraction 5. Data preparation 6. Data validation 7. Debugging and maintenance 8. Generalization” (MUNZERT, 2014, p. 221–222)

“Each scraping scenario is different, so some of the advantages or disadvantages of a method may not apply for your task. Besides, as always, there is more than one way to skin a cat.” (MUNZERT, 2014, p. 270)

“Scraping with HTTP and extracting information with regular expressions, information extraction via XPath queries, and data gathering using APIs. They should usually be preferred over each other in ascending order (i.e., scraping with regular expressions is least preferable and gathering data via an API is most preferable),” (MUNZERT, 2014, p. 270)

“If data on a site are not provided for download in ready-made files or via an API, scraping them off the screen is often the only alternative.” (MUNZERT, 2014, p. 270)

“The fact that the overwhelming majority of resources on the Web are still not accessed by web APIs motivates large parts of this book. This is no drawback of web services as a tool for data collection per se but merely reflects the fact that there are more data sources on the Web people like to work with than data providers who are willing to offer neat access to their databases.” (MUNZERT, 2014, p. 278)

“In the disclaimer of the book (see p. xix), we noted a caveat concerning the unauthorized use or reproduction of somebody else’s intellectual work.” (MUNZERT, 2014, p. 278)

“Scraping inherently involves copying, and therefore one of the most obvious claims against scrapers is copyright infringement.” Our advice is to work as transparently as possible and document the sources of your data at any time” (MUNZERT, 2014, p. 278)

“Not everything that can be scraped should be scraped, and there are more and less polite ways of doing it. The programs you write should behave nicely, provide you with the data you need, and be efficient—in this order.” (MUNZERT, 2014, p. 284)

“We suggest that if you want to gather data from a website or service, especially when the amount of data is considerable, try to stick to our etiquette manual for web scraping.” (MUNZERT, 2014, p. 284)

“As soon as you have identified potentially useful data on the Web, you should look for an “official” way to gather the data. If you are lucky, the publisher provides ready-made files of the data which are free to download or offers an API. If an API is available, there is usually no reason to follow any of the other scraping strategies. APIs enable the provider to keep control over who retrieves which data, how much of it, and how often.“ (MUNZERT, 2014, p. 284)

[ The issue of Digital Trace Data ]

**Referências**

BALTAR, R.; BALTAR, C. As Ciências Sociais na Era do Zettabyte. *Mediações - Revista de Ciências Sociais*, v. 18, n. 1, p. 11–19, 2013.

MUNZERT, S.  *XML and Web Technologies for Data Sciences with R* . [S.l: s.n.], 2014. v. 61.

MUNZERT, S. *et al.* *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*. West Sussex: John Wiley & Sons, 2015.

ÜNVER, H. A. Computational international relations what can programming, coding and internet research do for the discipline? *All Azimuth*, v. 8, n. 2, p. 157–182, 2019.