Capítulo 1

Data Ecosystem

A Data Analyst's ecosystem includes the following:

- Infraestructure
- Software
- Tools
- Frameworks
- Processes used to:
  - Gather,
  - Clean,
  - Mine,
  - and Visualize

  data.

1.1. Overview of the Ecosystem

Data can be categorized into the following types:

1. **Structured**: Data that follows rigid format and can be organized into rows and columns.

   This data is typically seen in databases and spreedsheets.

2. **Semi-structured**: Mixed of data that has consisten characteristics and data that does not conform to a rigid structure.

   For example, emails (strucured data such as subject, email, etc. . . and unstrucutred as the message).

3. **Unstrucutred**: Data that is complex and mostly qualitative information that cannot be structured into rows and columns.

   For example, photos, videos, files and social media contant.

Observación 1.1.1 (**Importance of Data Types**)
The type of data determines the type of data repositories where data can be collected, stored in and, tools used to query or process data.

1.1.1. Data Formats

Data can come in a variety of file formats, such as:

1. Relational.
2. Non-relational databases.
3. APIs.
4. Web Services
5. Data streams.
6. Social Platforms.
7. Sensor Devices.

Definición 1.1.1 (**Data Repositories**)
A **Data Repository** is a container of data. These include:

1. Databases.
2. Data Warehouses.
3. Data Marts.
4. Data Lakes.
5. Big Data Stores.

A Data Repository collects types of data, file formats and are sources of data. A data repository is dependent upon this characteristics mentioned earlier, to collect, store and mine data.

Ejemplo 1.1.1
A Big Data we will need data warehouses, where we need to store and process large volume and high velocity data. Also, frameworks that allow perform complex analytics in big data.

1.1.2. Languages

The languages available in the Data Analyst Ecosystem are:

- **Query Languages**: Such as SQL for quering data and manipulating data.
- **Programming Languages**: Such as Python for data applications.
- **Shell and Scripting Languages**: For repetitional and operational tasks.

1.1.3. Automated Tools

Automated tools, frameworks, and processes for all stages of the analytics process are part of the Data Analysis Ecosystem. These tools allow to:

- **Gather, extract, transfrom and load data**.
- **Data wrangling and cleaning**.
- **Data analysis and mining**.
- **Data visualization**.

1.2. Data

Definición 1.2.1 (**Data**)
Data is unorganized information that is processed to be meaningful. This can be conformed of:

- Facts, observations, perceptions.
- Numbers, characters, symbols.
- Images.

Data can be interpreted to have a meaning.

1.2.1. Categorization of Data

We can organize data in the following three types:

1. **Structured**.
2. **Semi-structured**.
3. **Unstrucutred**.

1.2.2. Structured Data

The characteristics of Structured Data are the following:

- Well-define structure.
- Can be stored in well-defined schemas.
- Can be represented in a tabular maner using rows and columns.

Uses facts and numbers which can be:

- Collected.
- Exported.
- Stored.
- Organized.

in databases.

Ejemplo 1.2.1 (**Sources of Structured Data**)
Structued data can come from:

1. SQL Databases.
2. Online Transaction Processing.
3. Spreadsheets.
4. Online forms.
5. Sensors GPS and RFID text.
6. Network and webserver logs.

1.2.3. Semistructured Data

Characteristics of semi-structured data are the following:

- Has some organizational properties but lacks a fixed or rigid schema.
- Cannot be stored in rows and columns.
- Contains tangs and elements or metadate used to group and organize it in a hierarchy.

Ejemplo 1.2.2 (**Sources of Semi-structured Data**)
Include:

- Emails.
- XML and other markup languages.
- Binary executables.
- TCP/IP packets.
- Zipped files.
- Integration of data from different sources.

Observación 1.2.1 (**Use of XML and JSON**)
`XML` and `JSON` allow users to define tags, associate atributes and store data in a hierarchy form and are used widely to store and exchange semi-structured data.

1.2.4. Unstructured Data

Characteristics of unstructured data are the following:

- Does not have an easily identificable structure.
- Cannot be organized ina mainstream relational database in the form of rows and columns.
- Does not follow any particular format, sequence, semantics and rules.

Unstrucutred data can deal with a heterogeneity of sources and has applications in business.

Ejemplo 1.2.3 (**Sources of Unstructured Data**)
Sources of unstructured data include the following:

- Web pages
- Social media feeds
- Images in a varied file formats
- Video and audio files
- Documents and PDF files
- PowerPoint presentations
- Media logs

- Surveys

Can be stored in files and documents, such as:

- **Files and Docs** for manual analysis.
- **NoSQL Databases** for the use of analysis tools.

1.3. File Structures

Al told earlier, data has to be saved and we have plenty of file formats in order to store data. Turns out important to understand the structure of file formats in order to choose between their benefits and limitations.

We will see the following file formats:

- **Delimited file formats or** `.CSV`**.**
- **Microsoft Excel Open XML spreedsheet or** `.XLSX`**.**
- **Extensible Markup Langauge, or** `.XML`**.**
- **Portable Document Format, or** `.PDF`**.**
- **JavaScript Object Notation, or** `.JSON`**.**

1.3.1. Delimited Text Files

Definición 1.3.1 (**Delimited Text Files and Delimiters**)
Delimited Text Files are *files used to store data as text in which each line and row has a value separated by a delimiter.*

A **delimiter** is a *sequence of one or more characters for specifying the boundary between independent entities or values.*

Most common delimeters are coma, tab, colon, vertical bar and space.

Ejemplo 1.3.1 (**Commonly Used Delimited Text Files**)
Two delimited text files are Comma-separated values and Tab-separated values (`.CSV` and `.TSV`, respectively) are the most commonly used.

Ejemplo 1.3.2
A `.CSV` looks like this:

```
1 id,nombre,edad,puesto,salario
2 1,Cristo Alvarado,24,Data Analyst,45000
3 2,Ana Lopez,29,Software Engineer,62000
4 3,Marco Gomez,31,Project Manager,70000
5 4,Laura Ruiz,26,QA Tester,40000
6 5,Carlos Perez,35,DevOps Engineer,75000
```

Código 1.1: `.CSV` File Content Example.

And, a `.TSV` like this:

```
1 id  nombre  edad  puesto  salario
2 1 Cristo Alvarado 24  Data Analyst  45000
3 2 Ana Lopez 29  Software Engineer 62000
```

```
3 Marco Gomez 31  Project Manager 70000
4 Laura Ruiz  26  QA Tester 40000
5 Carlos Perez  35  DevOps Engineer 75000
```

Código 1.2: .TSV File Content Example.

The first row are the names of the variables of each column.

Observación 1.3.1
Delimiters represent one of variuous means to specify bonudaries in a data stream.

1.3.2. Microsoft Excel Open XML Spreadsheet or .XLSX

Is a Microsoft Excel Open XML file format that falls under the spreadsheet file format. It is an XML-based file format created by Microsoft.

Is a secure file format since it cannot contain malicious malware.

1.3.3. Extensible Markup Language or .XML

Definición 1.3.2 (**Extensible Markup Language (XML)**)
Extensible Markup Language is a markup language with a set of rules for encoding data.

- This format is readable by humans and machines.
- Self-descriptive language.
- Similar to .HTML in some respects.
- Does not use predefined tags like .HTML does.
- Platform independent.
- Programming langauge dependent.
- Makes it simpler to share data between systems.

Ejemplo 1.3.3
An example of a .XML file is the follwing:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<employees>
  <employee>
    <id>1</id>
    <nombre>Cristo Alvarado</nombre>
    <edad>24</edad>
    <puesto>Data Analyst</puesto>
    <salario>45000</salario>
  </employee>
  <employee>
    <id>2</id>
    <nombre>Ana Lopez</nombre>
    <edad>29</edad>
    <puesto>Software Engineer</puesto>
    <salario>62000</salario>
  </employee>
```

```xml
  <employee>
    <id>3</id>
    <nombre>Marco Gomez</nombre>
    <edad>31</edad>
    <puesto>Project Manager</puesto>
    <salario>70000</salario>
  </employee>
  <employee>
    <id>4</id>
    <nombre>Laura Ruiz</nombre>
    <edad>26</edad>
    <puesto>QA Tester</puesto>
    <salario>40000</salario>
  </employee>
  <employee>
    <id>5</id>
    <nombre>Carlos Perez</nombre>
    <edad>35</edad>
    <puesto>DevOps Engineer</puesto>
    <salario>75000</salario>
  </employee>
</employees>
```

Código 1.3: .XML File Content Example.

1.3.4. Portable Document File (PDF)

Definición 1.3.3 (**Portable Document File (PDF)**)
A **Portable Document File** .PDF is a file format developed by adobe to present documents independent of application software, hardware, and operating systems.

- Can be viewed the same way on any device.
- Is frequently used in legal and financial documents.
- Can also be used to fill data for forms.

1.3.5. JavaScript Object Notation (JSON)

Definición 1.3.4 (**JavaScript Object Notation (JSON)**)
A **JavaScript Object Notation** .JSON is a text-based open standard designed to transmit data over the web.

- Langauge-independent data format.
- Can be read by any programming language.
- Easy to use.
- Comptaible with a wide range of browsers.
- Best tools for sharing data.

Ejemplo 1.3.4
An example of a .JSON file is the follwing:

```
{
  "employees": [
    {
      "id": 1,
      "nombre": "Cristo Alvarado",
      "edad": 24,
      "puesto": "Data Analyst",
      "salario": 45000
    },
    {
      "id": 2,
      "nombre": "Ana Lopez",
      "edad": 29,
      "puesto": "Software Engineer",
      "salario": 62000
    },
    {
      "id": 3,
      "nombre": "Marco Gomez",
      "edad": 31,
      "puesto": "Project Manager",
      "salario": 70000
    },
    {
      "id": 4,
      "nombre": "Laura Ruiz",
      "edad": 26,
      "puesto": "QA Tester",
      "salario": 40000
    },
    {
      "id": 5,
      "nombre": "Carlos Perez",
      "edad": 35,
      "puesto": "DevOps Engineer",
      "salario": 75000
    }
  ]
}
```

Código 1.4: .JSON File Content Example.

1.4. Sources of Data

Common sources of data are:

- Relational databases.
- Flat files and XML Datasets.

- APIs and Web Services.
- Web Scraping.
- Data Streams and Feeds.

1.4.1. Relational Databases

Typically, an organizational unit posess a store of data ot his:

- Business activities.
- Customer transactions.
- Human resource activities.
- Workflows.

These systems use Relational Databases, such as SQL Server, Oracle, MySQL and IBM DB2. In this relational databases they store structured data.

These relational databases could be used for analysis.

1.4.2. Flat File and XML Datasets

External to the organization, there are other datasets.

Ejemplo 1.4.1
Goverment can give datasets which are either public or private. For example, demographic or economic dataset are released on a regular time period.

Also, this type of data could be a point of sale, financial or weather.

Idea 1.4.1
This data could be used in companies to define a strategy, predict demand, and make distribution decisions, among other things.

This data is often available in form of:

- Flat files.
- Spreadsheet files.
- XML documents.

Definición 1.4.1 (**Flat Files**)
Flat Files store data in plain text format. Each line, or row is one record. Each value is separated by a delimiter (comma, semicolon, tabs, etc...).

Flat Files only have one table to organize all of his data. A common example of a Flat File are `.CSV` and `.TSV` files.

Definición 1.4.2 (**Spreadsheet Files**)
Spreadsheet files are a special type of flat files, which can organizse data in a tabular format, can contain multiple worksheets.

Common examples are `.XSL` or `.XLSX` spreadsheet formats.
Also, Google sheets, apple numbers and libreoffice calc.

Definición 1.4.3 (**XML Files**)
XML Files contain data values that are identified or marked up using tags (as we saw earlier). Can support more complex data structures.

The uses we have for XML files are online surveys, bank statements, and other unstructured datasets.

1.4.3. APIs and Web Services

Definición 1.4.4 (**Application Program Interface API**)
An **Application Programming Interface (API)**, is a *set of rules and protocols that allows different software applications to communicate and interact with each other*. It acts as *an intermediary, enabling one application to request data or functionality from another without needing to understand the internal workings of the other application.*

Definición 1.4.5 (**Web Service**)
A **Web Service** is a *software that enables machine-to-machine communication over the internet using standardized protocols like HTTP, allowing different applications to exchange data and function together regardless of their underlying programming languages or platforms.*

So, basically we can obtain data using APIs and Web Services.

Observación 1.4.1 (**Calling an API or Web Service**)
Typically, APIs and Web Services listen for upcoming requests, which can be in form of Web requests or Network requests. After a request has been made, they can return data in form of a JSON, XML, Media Files, etc...

Ejemplo 1.4.2 (**Examples of APIs**)
Some popular APIs are the following:

- Twitter and Facebook APIs.
- Stock Market APIs.
- Data Lookup and Validation APIs.

1.4.4. Web Scraping

Definición 1.4.6 (**Web Scraping**)
Web Scraping is *the automated process of using bots to extract data from websites and save it into a structured format, like a database or spreadsheet.*

Instead of copying and pasting by hand, software is used to parse a website's HTML code to pull specific information, which can then be used for analysis, research, or other applications.

Observación 1.4.2 (**Use of Web Scraping**)
Web Scraping is useful to:

- Extract data from unstructured sources.

- Also known as screen scraping, web harvesting, and data straction.
- Download specific data based on defined parameters.
- Can extract text, contact information, images, videos, product items, and more...

Ejemplo 1.4.3
Popular uses of Web Scraping are:

- Providing price comparasions by collecting product details from retailer, manufacturers, and eCommerce websites.
- Generating sales leads though public data sources.
- Extracting data from posts and authors on varios forums and communities.
- Collecting training and testing datasets for machine learning models.

Observación 1.4.3 (**Popular Web Scraping Tools**)
Some popular web scraping tools are:

- BeautifulSoup.
- Scrapy.
- Pandas.
- Selenium.

1.4.5. Data Streams

Definición 1.4.7 (**Data Stream**)
A **data stream** *is a continuous, ordered sequence of data generated by a source in real-time. Unlike data that is stored and processed in batches, a data stream is processed as it arrives, allowing for immediate analysis and action.*

Examples include sensor data from IoT devices, website clickstreams, or financial transactions. This data can come from:

- Stock market tickers for financial trading.
- Retrail transaction streams for predicting demand and supply chain management.
- Surveillance and video feeds for threat detection.
- Social media feeds for sentiment analysis.
- Sensor data feeds for mnoitoring industrial or farming machinery.
- Web clicks feeds for monitoring web performance and improving design.
- Real-time flight events for rebooking and rescheduling.

Observación 1.4.4
Some popular technologies used to preocess data streams include:

- Apache Kafka.
- Apache Spark.
- Apache Storm.

Observación 1.4.5
RSS (or Really Simple Syndication) feeds are another pupular data sources. These are used for capturing updated date for online forum and news sites, where the data is refreshed on an ongoing basis.

Using a feed, we can use RSS and convert this data to use it.

1.5. Languages for Data Profesionals

We will see some of the most used languages used by data analysts. These can be:

- **Query languages**. These languages are designed for accessing and manipulating data in a database (SQL).
- **Programming languages**. These languages are used for developing applications and controlling application behavior (Python, R, Java).
- **Shell scrpiting**. Ideal for repetitive and time consuming operational tasks (Unix/Linux Shell, PowerShell).

1.5.1. Query Langauges

Definición 1.5.1 (**Structured Query Language (SQL)**)
Structured Query Language (SQL) is a *querying language designed for accessing and manipulating informacion from, mostly, though not exclusively, relational databases.*

With SQL we can perform operations such as:

1. Insert, update and delete records in a database.
2. Create new databases, tables and views.
3. Write stored procedures.

Observación 1.5.1
Advantages:

- SQL is portable and platform independent.
- Can be used for querying data in a wide variety of databases and data repositories.
- Has simple sintax similar to english.
- Allows developers to write programs with fewer lines of code using keywords.
- Can retrieve large amounts of data quickly and efficiently.

- Runs on an interpreter system.

1.5.2. Python

Python is widely used open-source general-purpuse, high-level programming language.

- Allows programmers to express their concepts in fewer lines of code.
- Ideal for beginners.
- Great for performing high-computational tasks in large volumes of data.
- A lot of In built-functions.
- Multiple programming paradigms.

Idea 1.5.1
Some libraries used in Python are:

- Pandas for cleaning and analysis.
- Numpy and Scipy for statistical analysis.
- BeautifulSoup and Scrapy for web scraping.
- Matplotlib and Seaborn to visually represent data in the form of bar graphs, histogram and pie-charts.

1.5.3. R

R is an open-source programming language and enviroment for data-analysis, data visualization, machine learning, and statistics.

Used for:

- Developing statistical software.
- Performing data analytics.
- Creating compelling visualizations.

One advantage is that it can be paired with many programming languages. It's highly extensible.

Facilitates the handling of structured and unstructured data.

Observación 1.5.2 (**Advantages of R**)
This programming language offers libraries such as Ggplot2 and Plotly that offer aesthetic graphical plots to its users.

Allows data an dscripts to be embedded in reports.

Allows the creation of interactive web apps.

1.5.4. Unix/Linux Shell

Definición 1.5.2 (**Unix/Linux Shell**)
A **Unix/Linux Shell** is a *computer program written for the UNIX shell, It is a series of UNIX commands written in a plain text file to accomplish a specific task.*

Writing a shell script is fast and easy.

Observación 1.5.3
Typical operations performed by shell scripts include:

- File manipulation.
- Program execution.
- System administration tasks such as disk backups and evaluating system logs.
- Installation scripts for complex programs.
- Executing routine backups.
- Running batches.

1.5.5. PowerShell

Definición 1.5.3 (**PowerShell**)
PowerShell is a *cross-platform automation tool and configuration framework by Microsoft that its optimized for working with structured data formats, such as JSON, CSV, XML and REST APIs, websites, and office applications.*

It consists of a command-line shell and a scripting language.

Its object based, to it can be used to filter, sort, measure, group, and compare objects as they pass through a data pipeline.

Also, it's a good tool for data mining, building GUIs, creating charts, dashbords, and interactive reports.

1.6. Data Repositories

A data repository is data that has been collected, organized and isolated to use in business operations, mined for reporting and data analysis.

It can be or several databases.

Types of data repositories include:

- Databases
- Data warehouses
- Big Data Stores