

Quarterly Newsletter

CHAIR

iDANAE

INTELLIGENCE · DATA · ANALYSIS · STRATEGY

1Q22



Open Source ecosystems



POLITÉCNICA

UNIVERSIDAD
POLITÉCNICA
DE MADRID

MS^o *Management
Solutions*
Making things happen

Introduction



The evolution from an information society to a knowledge society is closely related to the development of information systems and new technologies. Such systems are based on information, which has become an instrument of human development to identify, produce, process, transform, disseminate and use knowledge¹. The development of business tools to carry out these processes has undergone a major transformation. In fact, already at the beginning of the 21st century, the European Commission considers the concept of the digital ecosystem as an evolution of collaborative environments in some reports².

Therefore, one of the key elements for the extraction and processing of information is the development of efficient software, which is available in a technological environment suitable for its use. This has led to a natural evolution from traditional information systems to the so-called Open Source ecosystems.

Indeed, a growing adoption of Open Source by enterprises has been observed in recent years: in Q4 2021, the Open Source collaboration platform GitHub reported more than 16 million new users over the previous year³, with more than 73 million active developers, a growth of 30% over 2020. In addition, GitHub Enterprise (the enterprise-oriented version of GitHub) is

used by 84 of the top 100 companies with highest gross revenue in the US.

In the case of Europe, the Open Source ecosystem is increasingly becoming part of the business environment⁴. According to a Red Hat report⁵, 75% of enterprises use Open Source software in key areas of their infrastructure, such as security (52%), cloud tools (51%), databases (49%) and analytics and big data (47%).

Open Source also brings a dynamic perspective to the software development process, allowing faster progress to be made in code generation. This perspective is achieved through the community, where collaborative development and publication of projects allows for different forks in the evolution of a program, resulting in a competitive development environment.

The aim of this newsletter is to illustrate the main concepts of the Open Source domain, and to show the state of the art of technological ecosystems whose software components have Open Source and free software licences.

¹UNESCO (2005).

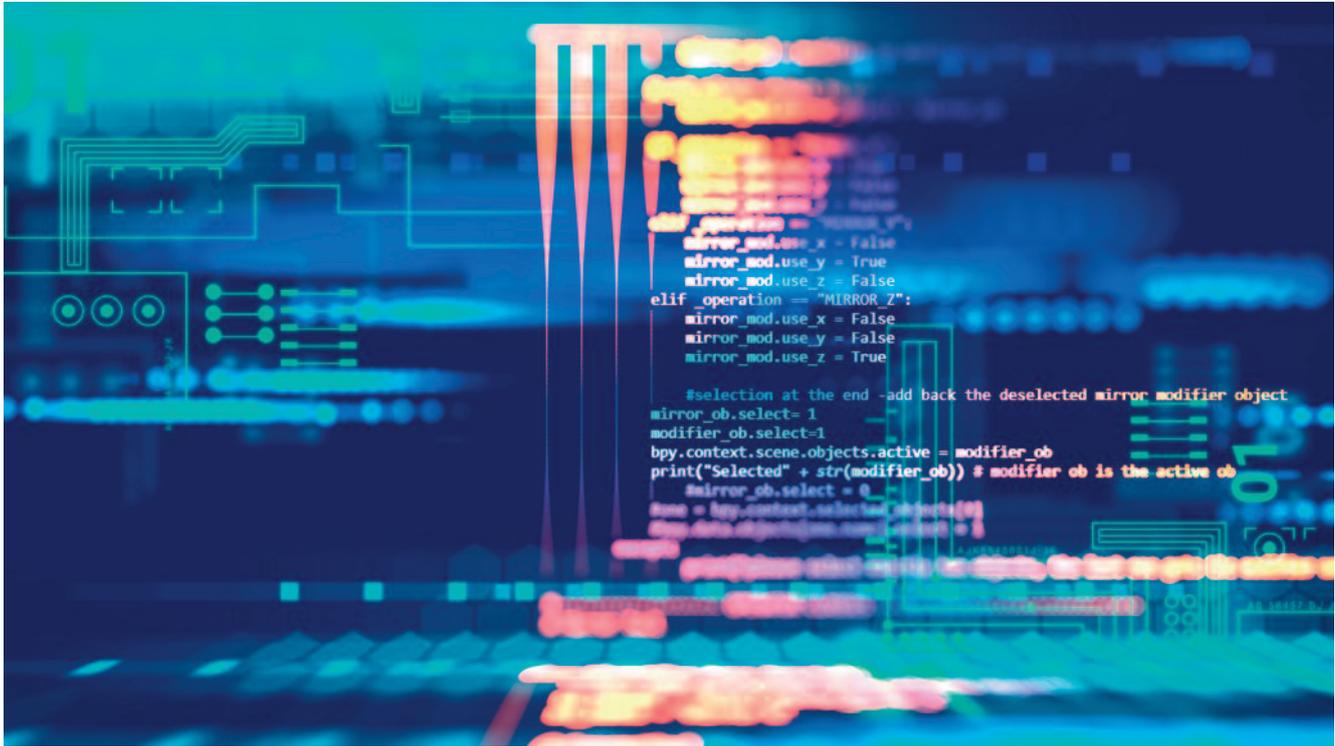
²García, A. and García, F. (2018).

³Octoverse (2021).

⁴NautaCapital (2021).

⁵Haff, G. (2021).

Concept and history



The term Open Source originally refers to code designed to be freely accessible to all types of users. Thus, anyone is able to view, transform and distribute the code without restrictions.

Open Source software development is decentralised and collaborative: any working group can contribute to the development of a piece of code, so that peer review and community production become important factors. This makes the software generation process more cost-effective and flexible than some proprietary solutions⁶.

Open Source software makes its source code legally available to end users through a specific licence⁷. Software is normally considered Open Source if it meets the following conditions:

- ▶ The source code is available at no additional cost, which means that users can view the software code and make any changes they wish.
- ▶ The source code can be reused in new software, so anyone can use the source code to develop their own program and distribute it.

Being Open Source does not mean that executable software is distributed free of charge. However, it does mean that its source code is available at no cost. There are two concepts related to the Open Source paradigm: free software and Open Source software. The former refers to an ethical concept according to which users of free software should always have the right to run, analyse, modify and distribute the software in question. Furthermore, any distribution of any modification of free software must also be free software. The second concept refers to a pragmatic aspect and bases its existence on the ineffectiveness of proprietary code to solve certain problems. Among the considerations of Open Source software is not the freedom of the user to use or modify that code, but that the source code is available to the user. Thus, any modification of Open Source software does not necessarily have to be Open Source or guarantee that its source code is available to users.

⁶Andrew M. St. Laurent (2008).

⁷See the following section for more details on the existing types of licences.



The origin of the Open Source paradigm can be traced back to the 1950s and 1960s, when researchers who carried out early developments of Internet-related technologies and telecommunications network protocols relied on a collaborative and open research environment. The Advanced Research Projects Agency Network (ARPANET), which would eventually become the basis for today's Internet, encouraged the collaborative review and feedback process that enabled the breakthrough that occurred at that time.

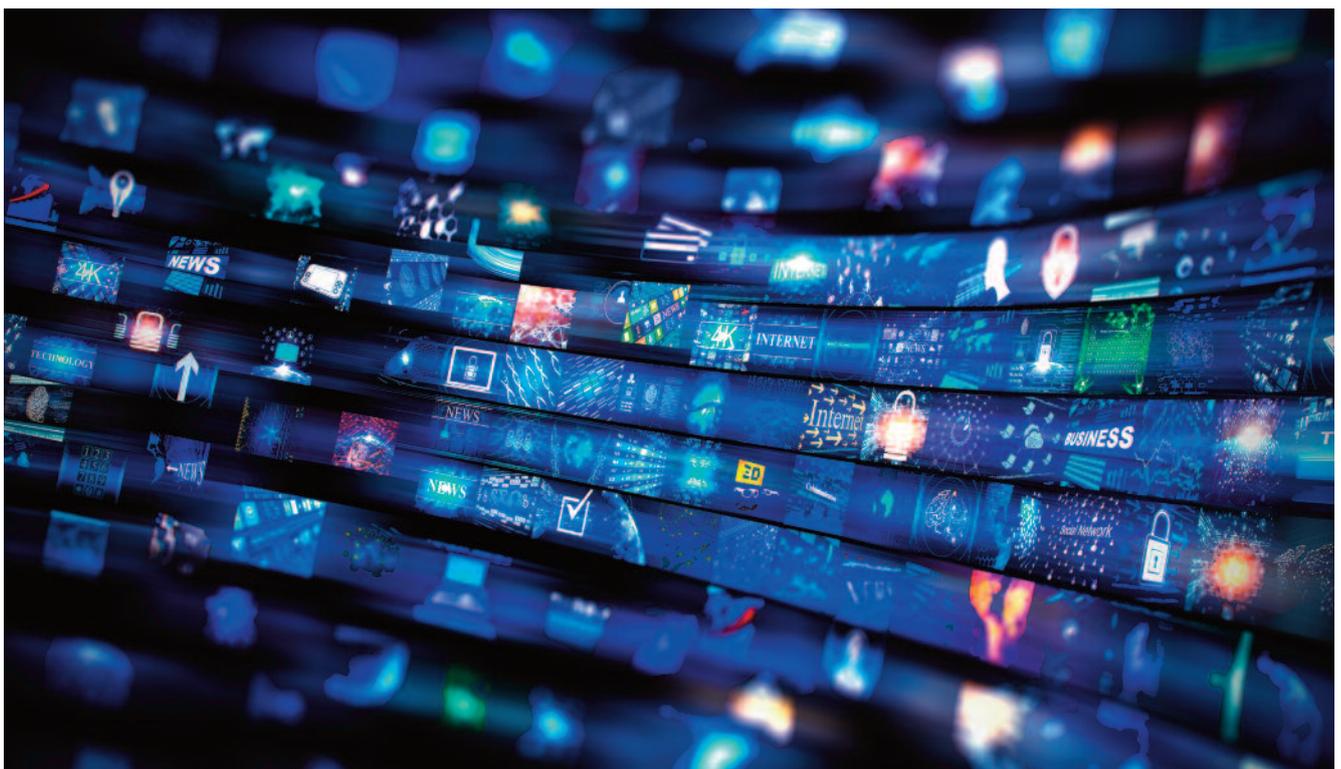
User groups shared source code, which could be used as a basis for new developments, and conversations were held through forums where standards for collaboration and open communications were developed. By the early 1990s, when the Internet originated, all these values and best practices that were part of this open culture were already well established in software production.

The advantages of this approach made Open Source development a way of working that transcended software production: many business models in different productive sectors adopted its values and its decentralised production model to find new ways of solving problems.

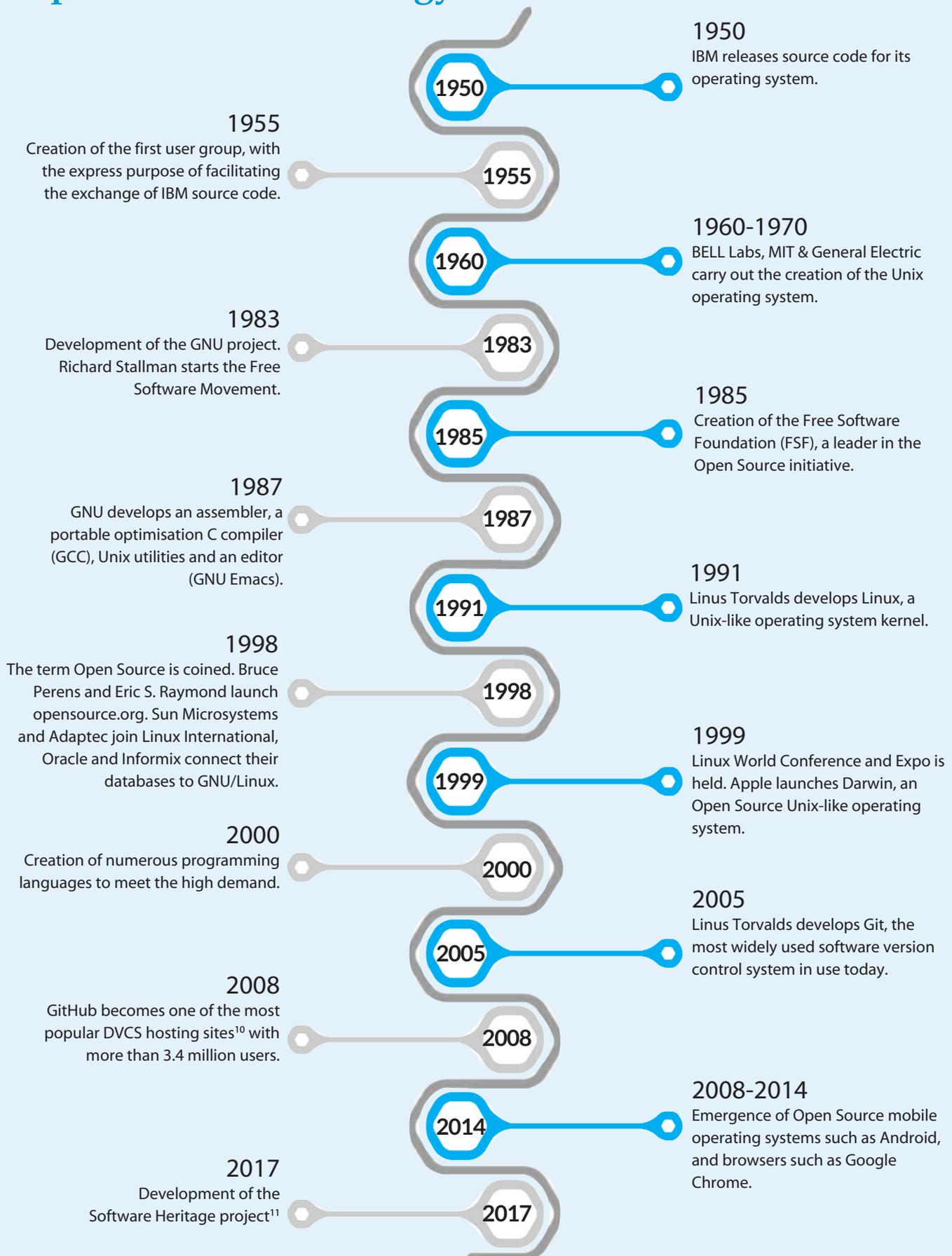
Today, the Open Source movement has become one of the fundamental pillars of digital transformation, both in small and large companies. Tesla, OpenAI, Facebook and Google are some of the companies leading the way in Open Source innovation. Technology giants such as Apple and Microsoft already offer software compatible with Linux⁸.

One of the largest Open Source projects is the Software Heritage Initiative, whose mission is to bring together in one place all the world's source code so that anyone can easily explore it⁹. It was carried out at Inria (Institut National de Recherche en Informatique et en automatique) under the protection of UNESCO, with the intention of preserving the knowledge of source code and making it widely available through a large digital library. In 2017, UNESCO declared this software to be part of the cultural heritage of humanity, to be preserved in the same way as music or literature. Public institutions such as the French Ministry of Innovation and several universities, but also banks such as Société Générale and companies such as Microsoft, Google, Intel and Huawei are contributing to its funding.

⁸Jim Whitehurst.
⁹Software Heritage.



Open Source Chronology



¹⁰Distribution Version Control Systems: allows many software developers to work on a common project without the need to share a common network.

¹¹<https://www.softwareheritage.org/>

Open Source ecosystems



Definition and Functioning

A software ecosystem is a workspace in which a series of tools coexist that, accompanied by good practices, allow a development team to carry out a project involving software under a working methodology¹². With the development of new applications and the use of open software, many companies have managed to create an Open Source ecosystem, a community with a multitude of developers who advocate Open Source and who develop their projects in a dedicated way while sharing their progress with other contributors outside the company. One example is the ecosystem created by Intel "01.org"¹³, where Intel developers share the projects they maintain and develop.

However, not all companies are in favour of sharing their code. Depending on the purpose of the code, a distinction can be made between Open Source and closed source software:

- ▶ Open Source is code whose source code is publicly available, and which also allows it to be used, modified and redistributed in a completely free manner.

- ▶ Closed source code, on the other hand, does not allow access to its source code, which makes it impossible to analyse its functionalities through the code. In this case, the guarantee regarding the processing of personal data or the security of the software, among others, is delegated to the owner of the code (whether a natural or legal person). Likewise, its modification and subsequent distribution is prevented.

The distribution of software, whether open or closed, is done through the use of different software licences, which constitute a contract between the distributor of the code and the end users of the software. These licences establish the criteria under which the users can make the software available and under what conditions they can redistribute or modify the code, if the licence allows it.

¹²J.V. Joshua, D.O. Alao, S.O. Okolie, O. Awodele (2013).

¹³<https://01.org/community>

Broadly speaking, two main types of licence can be distinguished¹⁴:

Open Source licences:

- ▶ Permissive: allow modification and distribution of the original code without any limitations.
- ▶ Robust: they limit the type of licence that can be granted to derivative software, which is based on modification of the original code. Within this group, two types can be found:
 - Strong: they oblige to keep the same licence as the original code in all subsequent modifications made.
 - Weak: they oblige to keep the same licence as the original code on code modifications, but allow freedom in the case of derivative works.

Closed source licences: these licences set limitations on the use, modification and distribution of the code at the owner's discretion. They usually limit modification and distribution, and set limits on the number of copies that can be used, and on the purposes for which they can be used.

The most commonly used licences include the following:

- ▶ GNU GPL¹⁵ (Strong Robust Open Licence): this licence allows to use and modify the software freely. It also allows to distribute new versions of the software as long as they are done under the same licence. Examples of Open Source software using this licence are Bash and GIMP.
- ▶ MIT¹⁶ (permissive open licence): this licence allows to use and modify the software freely. It also allows to distribute new versions of the software under any other licence, including closed source licences.
- ▶ Apache License¹⁷ (permissive open licence): this licence allows the software to be used and modified freely as long as the copyright and disclaimer are preserved, but does not require making the source code available in distributed developed versions. Examples of Open Source software using this licence are Android and Swift.

Ecosystem typologies

The main concepts and examples of some of the actors in the technological framework that make up software ecosystems in different areas of software development are presented below.

Operating systems

Among the Open Source ecosystems, Linux¹⁸, the kernel of an operating system developed by the Finnish programmer Linus Torvalds, stands out. Today, Linux is generally understood as a set of Open Source operating systems that have become one of

the most popular on the market. They are most commonly used on web servers, on supercomputers and even at CERN¹⁹. The development of these operating systems is one of the most prominent examples of free software: all their source code can be freely used, modified and redistributed by any person, company or institution, under the terms of the GNU General Public License.

The main advantage that Linux has over other operating systems is that it was released under a strong robust Open Source licence, publishing its source code, and allowing its modification and distribution, but always maintaining the original licence. On the Linux kernel there are commercially supported distributions such as Fedora (RedHat)²⁰, openSUSE²¹ (SUSE) and Ubuntu²² (Canonical Ltd) and also distributions maintained by the community itself such as Debian²³.

In addition, Linux offers a high degree of modularity (i.e. different modules or parts of the operating system, such as the graphical interface or the file system, are built on top of the kernel, without the kernel depending on a single module in particular, making it customisable). Each of the modules is separated independently, so that if one fails, it does not affect the others or the kernel, which makes these operating systems fault-tolerant.

Integrated Development Environment (IDE) tools

An integrated development environment (IDE) is a computer application that facilitates the developer's task of software development through various functionalities. Generally, an IDE has a source code editor, a compiler for the developed software and a debugger for error handling. IDEs allow developers to start programming new applications quickly, as they do not need to manually set up and integrate various tools as part of the configuration process²⁴.

There are several different commercial and technical business cases for IDEs, resulting in a wide variety of proprietary and Open Source IDE options on the market. In general, there are a number of important characteristics that differentiate IDEs, such as the number of languages they support, the operating systems

¹⁴<https://opensource.org/licenses/category>

¹⁵GNU General Public License (2007).

¹⁶<https://opensource.org/licenses/MIT>

¹⁷Apache License, version 2.0 (2004).

¹⁸<https://www.redhat.com/es/topics/linux>

¹⁹<https://home.cern/science/computing/open-source-open-science>

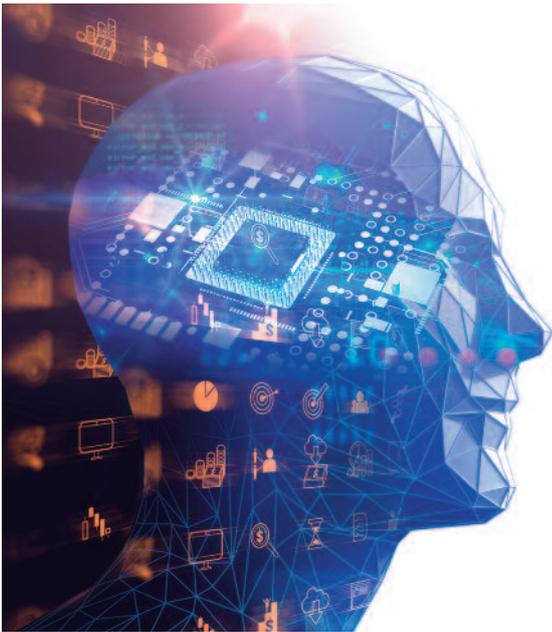
²⁰<https://getfedora.org/es/>

²¹<https://www.opensuse.org/>

²²<https://ubuntu.com/>

²³<https://www.debian.org/index.es.html>

²⁴The SDI concept (2019).



they support, the plugins²⁵ and extensions they contain, or the impact on system performance. Some of the most popular IDEs on the market are Eclipse, PyCharm, IntelliJ or XCode.

FrontEnd web development

As for FrontEnd development (the mechanism for web development that works on the user interaction interface), there are many Open Source tools available on the market, mostly based on the JavaScript programming language. This is a powerful language that interacts in a simple way with HTML²⁶ and CSS²⁷, which allows the developer to add dynamic functions to a web page.

A prominent JavaScript framework is AngularJS. Maintained by Google, it is used to create and maintain single-page web applications. It aims to augment browser-based applications with Model View Controller (MVC) capability, in an effort to make easier development and testing²⁸. Other popular frameworks in the industry are React or Vue.

BackEnd Development

Among the variety of languages for BackEnd development (the part of web development that ensures that all the logic of a web page works) currently available, Python stands out, an interpreted and multi-paradigm language with which any type of programme can be created, and which includes libraries and functions that guarantee a wide range of versatility. It also has multiple API to facilitate the use of other languages, such as Spark and its Pyspark API, and various tools such as Mongo and its Pymongo API. Another example is Java, an object-oriented programming language, with which code can be developed once and then run on any type of device. These two languages are, according to the TIOBE index²⁹, the first and third most

widely used programming languages in the world. This widespread use, due in part to its easy accessibility as Open Source software, is extremely useful for software development, as many of the problems that need to be addressed are already solved by other developers.

Databases

A database is a persistent collection of data, which is used by systems and applications³⁰. Databases are indispensable for any web application, and can be differentiated between relational and non-relational. Relational databases are based on the organisation of information in small chunks, which are related to each other by a series of identifiers. On the other hand, non-relational databases do not have an identifier that serves as a relationship between one set of data and another.

Examples of popular Open Source relational database management systems on the market are MySQL, PostgreSQL or MariaDB. These are database management systems that make use of multiple tables for storing and organising information. Non-relational databases include MongoDB and Cassandra, databases that do not store data in records, but in the form of key-value.

Code versioning control

Git³¹ is version control software designed by Linus Torvalds, with the efficiency, reliability and compatibility of application version maintenance in mind when applications have a large number of source code files.

Since the code is available to anyone who wishes to consult it, any user or developer can suggest improvements to it and resolve bugs at any time. In this sense, this is a key tool in Open Source ecosystems, as code repositories allow the coordination of the different versions developed by each of the multiple contributors to a project. This is essential when there are multiple developers working in a decentralised way.

²⁵Software that encapsulates and hides all the technical aspects associated with a particular application under a simple interface that allows a non-expert user to perform these tasks.

²⁶HyperText Markup Language (HyperText Markup Language) is the most basic component of the Web. It defines the meaning and structure of web content.

²⁷Cascading Style Sheets (CSS) is the style language used to describe the presentation of HTML documents.

²⁸<https://angularjs.org/>

²⁹TIOBE Index (2022).

³⁰C.J. Date (2004).

³¹<https://git-scm.com/>

This gives rise to the concept of a fork, which is the development of a software project based on an existing source code, giving rise to a branching of a parent project into several independent child projects, which may have different goals and different developers. For example, Android, Debian and Ubuntu are software forks derived from GNU Linux. This allows for faster progress in code generation, and a competitive dynamic, where the most efficient branches can be leveraged in achieving the final programme.

There are different services for hosting Git repositories, the most important of which is GitHub³², a platform that allows users from all over the world to collaborate, comment and develop code together, which is why it has become one of the main pillars of the Open Source ecosystems. Given the importance of distributing code for subsequent modification in the Open Source environment to include improvements or new functionalities, GitHub allows any registered user to download code, modify it, and propose incorporating their changes into the original code.

This has led to GitHub's rapid growth since its launch in 2008, from 46,000 repositories to 10 million in 2013³³, to the point where it was bought by Microsoft for \$7,500 million³⁴. Microsoft's evolution has led it to become one of the companies with the most developers contributing to Open Source on github³⁵.

Another of the most popular version control platforms is GitLab, which despite its similarity to GitHub has some notable differences. In principle, both platforms can be installed on a proprietary server; in the case of GitHub, the paid Enterprise version is required, while GitLab allows to host the programme on the server with the free Community Edition. The server stability of the hosted variant of GitLab is slightly worse than that of GitHub, so it can be very advantageous to install it a proprietary server. In addition, GitLab offers free continuous integration, which GitHub lacks³⁶.

DevOps tools

DevOps tools are understood as a work philosophy aimed at enabling a more agile and automated scaling of projects. They are used in all phases of software development and are essential for efficient development. One of the most widely used DevOps tools today is Docker: a platform that serves as a container for packages that includes libraries, files and configurations, speeding up their implementation and making continuous integration possible. Another example of a DevOps platform is Kubernetes, which automates container operations on Linux.

³²<https://github.com/>

³³B. Doll (2013).

³⁴T. Warren (2018).

³⁵En la última sección se detallan ventajas y riesgos asociados al Open Source para las empresas.

³⁶GitLab vs. GitHub (2020).

Example of development of a Machine Learning model based on Open Source

The lifecycle of a Machine Learning model can be structured in different phases that require specific capabilities, and where Open Source tools are currently very relevant in many projects. This life cycle can be conceptualised in three main blocks: Data, Development and Industrialisation³⁷.

- ▶ The Data block is the first phase of any Machine Learning project lifecycle. In this phase, the characteristics of the information storage to be used in the project are defined, as well as the way in which the data is ingested. For this purpose, there is a wide variety of Open Source ecosystems that facilitate this task. For example, we can highlight Pentaho, a Business Intelligence (BI) software that provides data integration, OLAP services, reports, information panels, data extraction, as well as data extraction, transformation and loading (ETL) capabilities.
- ▶ The Development block comprises multiple phases that follow one after the other for the generation of model creation pipelines. The first phase is the collection and preparation of data. This can be done using DVC, an Open Source version control system for Machine Learning projects, designed to make Machine Learning models shareable and reproducible. DVC can handle large files, datasets, models and metrics, as well as code. For the subsequent feature engineering and model training phases, there are a multitude of specialised Open Source libraries that speed up and optimise these tasks, such as those available in the Python programming language (for example, Scikit-learn, which has classic Machine Learning algorithms for classification, regression, clustering or dimensionality reduction tasks, among others; NumPy, SciPy or matplotlib). There are libraries specialised in Deep Learning, such as Tensorflow or Pytorch, which allow the architecture of each neural network to be built in blocks. The last phase of this block is the evaluation of the developed model, where MLflow, an Open Source platform, allows the lifecycle of Machine Learning to be managed, including experimentation, reproducibility, implementation and a central model registry.
- ▶ Finally, the Industrialisation block is developed: to put a model into production, it is possible to opt for the use of a high-level framework that allows the rapid development of web platforms in a secure and maintainable way. Among the many Open Source options available on the market, Django and FastAPI stand out. After putting the model into production, it is necessary to establish how to use the model correctly according to the needs of the project. Similarly, it is necessary to use a queue management tool, such as RabbitMQ or Apache Kafka. Finally, in order to monitor the model in production, updated data is used to analyse possible degradation of the model. For this purpose, one of the most widely used Open Source platforms is MLFlow, mentioned in the development phase.

³⁷More details on this type of project have been described in the MLOps publication of the same Chair. See iDanae (2020).

The Open Source community

The Open Source community is a very broad concept that includes both users of free software and its developers. The basic principle underlying any Open Source community is to share all source code with both developers and end users. The way to articulate and ensure the availability of this source code, and in general the Open Source character of a project, is achieved through Open Source licences.

One of the fundamental aspects of a community in general, and an Open Source community in particular, is how the contributions of its members are coordinated. Open Source communities are generally decentralised and non-hierarchical organisations. Projects are built with contributions from the various developers, which are reviewed by other community contributors before being added to each version of the project.

Another important consideration about Open Source communities is their recent new role as a tool for connecting prospective employees and employers. Many large technology companies use these communities to find potential employees and vice versa. This is often done by posing problems related to the development of Open Source projects (possibly from among those they are doing internally). In this way, they have all the resources of the Open Source community at their disposal, and those developers who contribute to solving

them in a significant way can be potential candidates for job offers. In recent years, the Open Source community has become an essential resource for large technology companies. In fact, an increasing amount of funding and Open Source code is being provided by these companies.

In the university environment, the Open Source movement has gained traction thanks to the university community, with a large number of projects of this type being created. There are many repositories where Open Source projects currently being developed by different universities and research groups are grouped together, such as MIT³⁸ or Harvard University³⁹. This facilitates project dissemination and collaboration.

³⁸<http://mitopensource.sourceforge.net/>

³⁹<https://github.com/IQSS/open-source-at-harvard>



A brief history of the Open Source community

Throughout history, the Open Source community has changed as society introduced computers into its development.

In 1983, when the free software movement was taking its first steps, the community was mainly made up of academics and professionals from the world of programming, although these members of the Open Source community had already been applying the principles of the Open Source movement since the 1950s. However, practically no company involved in the production of software did so under the Open Source philosophy.

The development of the Open Source community began to grow exponentially during the 1990s in parallel with the use of computers in the business and personal world. During these years, not only did the number of members of the Open Source community grow, but also the types of members expanded:

- ▶ Large non-profit foundations dedicated exclusively to promoting the creation of free software were created or consolidated. Some of the most relevant are the Apache Software Foundation⁴⁰ (ASF, 1999) and the Free Software Foundation⁴¹ (1985). These foundations have different ways of operating, but they pursue the same objective: to contribute to and promote the development of Open Source code.

The ASF is dedicated to developing and supporting software projects under the name Apache. It defines itself as a decentralised community of developers working on various Open Source projects, in which consensus among developers is necessary to determine the future of each project. Although there are project leaders for each project, they are elected by a vote among the project developers themselves. The ASF currently has more than 41,000 code developers and almost 500,000 community contributors who, despite working in a decentralised way, are organised around codes of conduct and good practices that are essential and common to forming part of their community of developers and contributors. Some of the projects it has developed and supports are Apache HTTP Server (one of the most widely used HTTP web servers), Cassandra (distributed NoSQL database) and Spark (Open Source cluster computing framework).

On the other hand, the Free Software Foundation focuses its efforts on articulating the legal and organisational measures necessary to ensure the survival of the Open Source philosophy (as opposed to Open Source code development).

- ▶ New companies dedicated exclusively to the development and commercialisation of Open Source software emerged and found ways to monetise it. One example is Red Hat⁴² (1993), which develops Open Source software mainly for companies, and even buys private software which it then distributes as Open Source.

On the other hand, many Open Source projects have continued to be developed at the university level. Some examples are highlighted below:

- ▶ MIT has developed a tool to evaluate the performance of computer vision tools in the biomedical field. This tool has tens of thousands of images and statistical tools for the evaluation of applications for tumour detection using different medical imaging modalities. It proposes a standard for homogeneous results across different research groups, allowing results to be compared on a common basis. Both images and code are available in a GitHub repository, and users are encouraged to adapt and improve the code for use in other projects⁴³.
- ▶ The AI Clinitian project at Imperial College London uses reinforcement learning techniques to modulate the intravenous treatment received by patients in the ICU for severe sepsis⁴⁴. In this case, the code is available on GitLab, and the project has been published in the journal Nature.
- ▶ Universidad Carlos III de Madrid has developed the deepImageJ plugin for the image processing tool ImageJ. This allows users with no previous training in Artificial Intelligence (AI), such as healthcare personnel, to use pre-trained neural networks for different applications in microscopy, such as density map estimation, or automatic cell segmentation⁴⁵.

⁴⁰<https://www.apache.org/>

⁴¹<https://www.fsf.org/es>

⁴²<https://www.redhat.com/es>

⁴³Toolkit for real-world performance evaluation of medical grade deep learning models (2021).

⁴⁴Komorowski, M., Celi, L. A., Badawi, O., Gordon, A. C., & Faisal, A. A. (2018).

⁴⁵<https://deepimagej.github.io/deepimagej/index.html>

Benefits and associated risks

The use of Open Source software offers many advantages, but it is important to understand and manage the potential risks associated with it:

1. One of the advantages of using Open Source software is its possible use as infrastructure (operating systems, web servers, databases, etc.). In this case, licensing costs can be saved.
2. On the other hand, a company can increase the added value it brings to its products by capturing novel or relevant approaches or developments for its solutions that have been developed by third parties outside the company.
3. Finally, by contributing to Open Source development, a company can build a good reputation, making it more attractive to developers and to other companies seeking partnerships.

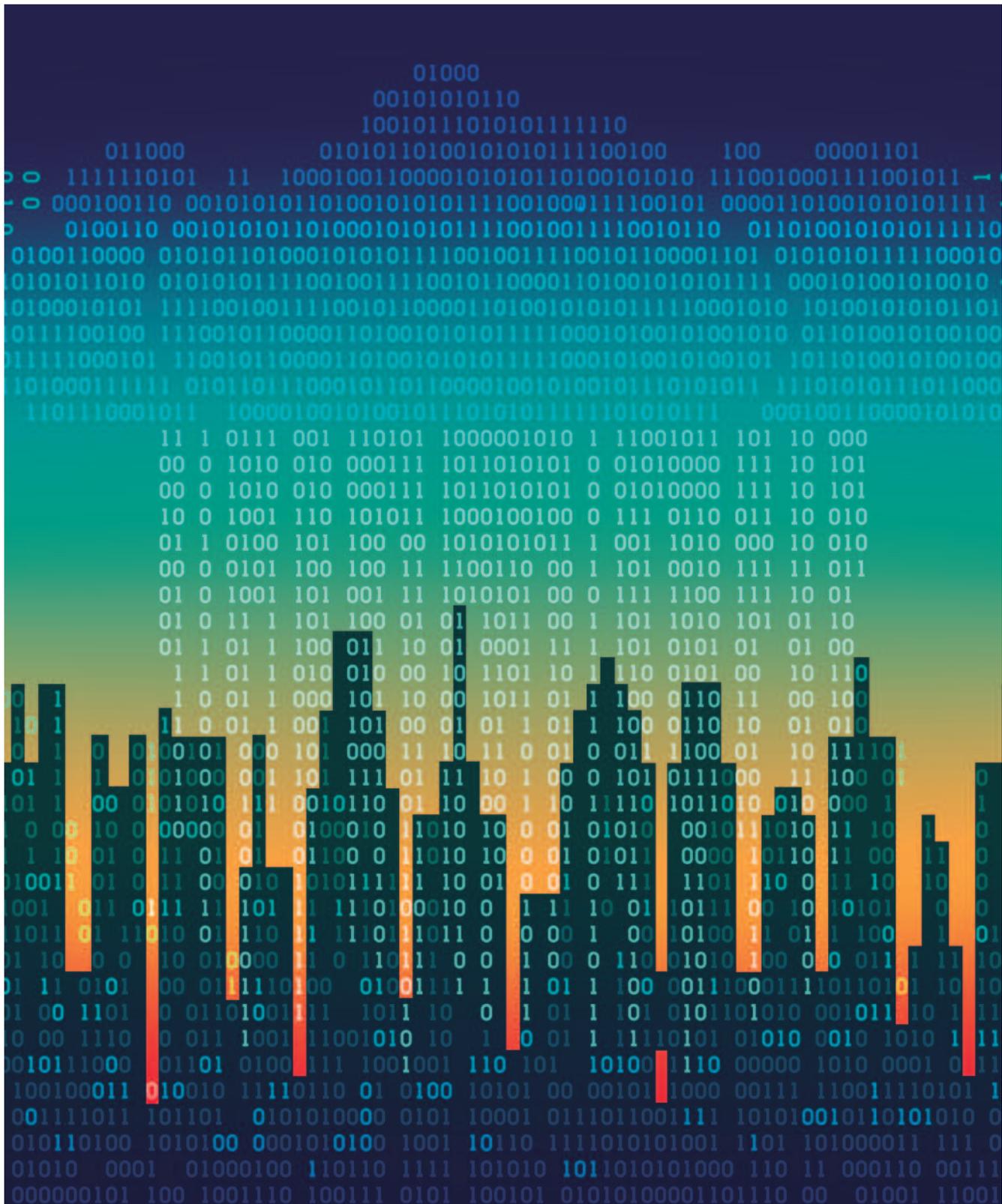
Although it has traditionally been considered difficult to make a return in Open Source (since many projects were initially created altruistically), different business models have been developed and have emerged over time, which has led many companies to evolve this model (and in many cases several of them coexist):

- ▶ One way to commercialise Open Source software is through service offerings (e.g. through technical support or consultancy services, including packaging or installation of software in the business environment for use in the core business). This way of commercialising Open Source software is the first to prove to be a viable form of business, with successful examples such as Red Hat or Suse.
- ▶ Another modality centred on the direct commercialisation of the software consists of the multi-licensing model, in which, on the one hand, the code is provided openly under a GNU copyleft licence for use in other Open Source projects; on the other hand, the possibility is given of acquiring a commercial licence or subscription that grants property rights and allows proprietary versions to be redistributed without distributing the code under a GNU licence. Some well-known examples of this modality are offered by large companies in the database field such as MySQL, with the free Community Edition and its paid Enterprise version, similar to MongoDB. This business model tries to create business opportunities by segmenting the market through the utility derived from the different licences that grant the software a certain use.

- ▶ Similarly, a new way of commercialising Open Source software is Open core: offering, maintaining and contributing to a free Open Source version of the code, usually more closely linked to the developer community, while offering extensions or new functionalities for a fee that complement the Open Source core. These services can include programming tools, cloud storage, or infrastructure management and maintenance (for example, companies such as Cloudera, GitLab or Aiven represent cases associated with these services).

However, the Open Source ecosystem is not risk-free. According to Red Hat⁴⁶, the main obstacles to the adoption of Open Source developments that are most important for companies are aspects such as code security (given that by sharing the source code openly, the software may be more susceptible to attacks, despite the fact that the community constantly analyses and contributes to fixing possible security breaches), the level of support, compatibility between versions or the lack of skills within the organisation. Additionally, from a legal point of view, Open Source commercial licences are usually not as clear as those of commercial software, which can pose a legal risk in the commercialisation of products. In addition, GNU licences oblige to distribute the source code of proprietary versions that are based on an original Open Source version, something that companies may not be willing to do.

⁴⁶Red Hat (2021).



Conclusions



Open Source has made it possible to promote the development of software in an open and collaborative way, thus encouraging a global and decentralised development model. This has boosted the development process, providing it with a number of advantages, such as greater speed, greater capacity for innovation, lower costs, interoperability and the exchange of ideas and knowledge.

Increasingly, universities, companies and individuals are adopting the Open Source approach, strengthening the community, and implementing different ecosystems, extending their scope. This has generated a trend that will continue to

consolidate in the future. However, the adoption of Open Source development systems requires companies to make an additional investment, as they have to transform themselves to incorporate Open Source into production processes, ensuring the viability and security of the systems, and developing skills through specialised training.

Bibliography

Andrew M. St. Laurent (2008). *Understanding Open Source and Free Software Licensing*. O'Reilly Media, 2008.

APACHE LICENSE, VERSION 2.0 (2004). Retrieved from <https://www.apache.org/licenses/LICENSE-2.0>

B. Doll (2013). *10 Million Repositories*, 2013. Archived from the original on October 9, 2017. Retrieved from <https://github.blog/2013-12-23-10-million-repositories/>

C.J. Date (2004). *An introduction to database systems*. 8th edition. Addison Wesley, 2004.

The SDI concept (2019). Retrieved from <https://www.redhat.com/es/topics/middleware/what-is-ide>

García, A. and García, F. (2018). García-Holgado, A. and García-Peñalvo, F.J. (2018). *Gestión del conocimiento abierto mediante ecosistemas tecnológicos basados en soluciones open source*. En J. A. Merlo Vega (Ed.), *Ecosistemas del Acceso Abierto* (pp. 147-160). Salamanca: Ediciones Universidad de Salamanca.

GitLab vs. GitHub (2020). *Comparison of both version control systems*. Retrieved from <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/gitlab-vs-github/>

GNU General Public License (2007). Retrieved from <https://www.gnu.org/licenses/gpl-3.0.html>

Haff, G. (2021). *Gordon Haff: The state of enterprise Open Source*. Red Hat. 2021.

iDanae (2020). *iDANA Chair: MLOps, a key element in the digital ecosystem*. 3T2020.

TIOBE Index (2022). Retrieved from <https://www.tiobe.com/tiobe-index/>

Jim Whitehurst. Retrieved from <https://www.inc.com/author/jim-whitehurst>

J.V. Joshua, D.O. Alao, S.O. Okolie, O. Awodele (2013). *Software Ecosystem: Features, Benefits and Challenges*, (IJACSA) *International Journal of Advanced Computer Science and Applications*, Vol. 4, No. 8, 2013.

Komorowski, M., Celi, L. A., Badawi, O., Gordon, A. C., & Faisal, A. A. (2018). *AI Clinician: Reinforcement Learning in Intensive Care*, 2018. Retrieved from <https://www.imperial.ac.uk/artificial-intelligence/research/healthcare/ai-clinician/>

NautaCapital (2021). *Open Source is Eating Europe*. May 2021. Retrieved from <https://nautacapital.com/wp-content/uploads/2021/05/Open-Source-Is-Eating-Europe-Report-2021.pdf>

Octoverse (2021). *The 2021 State of the Octoverse*. Retrieved from <https://octoverse.github.com/#lets-look-back-at-the-code-and-communities-built-on-git-hub-this-year>

Red Hat (2021). *The State of Enterprise Open Source*, 2021. Retrieved from <https://www.redhat.com/es/resources/state-of-enterprise-open-source-report-2021?intcmp=7013a000003BkveAAC>

Open Source versus closed source software. Retrieved from <https://www.ibm.com/es-es/topics/open-source>

Toolkit for real-world performance evaluation of medical grade deep learning models (2021). Retrieved from <https://www.media.mit.edu/projects/toolkit-for-real-world-performance-evaluation-deep-learning-models/overview/>

T. Warren (2018). *Microsoft completes GitHub acquisition*, 2018. *The Verge*. Archived from the original on November 9, 2020. Retrieved from <https://www.theverge.com/2018/10/26/17954714/microsoft-github-deal-acquisition-complete>

UNESCO (2005). Bindé, J. *Towards Knowledge Societies: UNESCO World Report*. UNESCO, 2005.

Authors

Ernestina Menasalvas (UPM)

Manuel Ángel Guzmán (Management Solutions)

Jaime del Río (Management Solutions)



POLITÉCNICA

UNIVERSIDAD
POLITÉCNICA
DE MADRID

MSO Management
Solutions
Making things happen

The Universidad Politécnica de Madrid is a multi-sector and multi-disciplinary Public Law Entity, which carries out teaching, research and scientific and technological development activities.

Management Solutions is an international consulting firm, focused on business, finance, risk, organization, technology and process consulting, operating in more than 40 countries and with a team of over 2,800 professionals working for more than 1,200 clients worldwide.

www.upm.es

www.managementsolutions.com

For more information visit

blogs.upm.es/catedra-idanae/