
Web de análisis financiero en sector público
Financial ratings in public services web



Miquel Vera Ramis
Aldair Fredy Maldonado Honores

Director - *Luis Garmendia Salvador*
Codirector - *Esteban Sastre Torregrosa*

2021/2022

TRABAJO DE FIN DE GRADO

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

As a solution to the current difficulties when trying to access the Spanish's government public financial data, analisisfinancierosectorpublico.com was born offering an open website that presents the desired data in a clear, fast and intuitive way. This application includes a set of functionalities such as financial reports, query interface and API for developers; the development of which will be fully detailed in this paper.

Key words

Base de Datos

Database

Informe

Report

Entidades públicas

Public facilities

Consulta

Query

Comunidad Autónoma

Autonomous Community

Diputación

Council

Municipio

Municipality

Algoritmo

Algoritmo

Ingresos

Income

Gastos

Expenses

Table of contents

1. Introduction	6
1.1 Motivations	6
1.2 Goals / Description of the work to do	6
1.3 Work Plan	7
1.4 Tools and Technologies	8
1.5 Related Subjects	8
2. Financial information and concepts	10
2.1 Autonomous Communities	10
2.2 Councils	14
2.3 Municipality	17
3. Pre-development decisions	18
3.1 Data interpretation and initial decisions	18
3.2 Technologies and tools selection	18
4. Development	21
4.1 Database design	21
4.2 DAO model (Decentralised Autonomous Organisation)	37
4.3 Algorithm for automatic insertion and update of data in the database	38
4.4 High fidelity prototype	42
4.5 Reports	43
4.6 PDF Generation	44
4.7 Income and Expense Excel Generation	45
4.8 Queries	48
4.9 API	51
4.10 Usage of Hosting	53
4.11 Individual work	53
5. Evaluation	57
5.1 Evaluation with final users	57
6. Instructions for use	58
6.1 Access to the site	58
6.1.1 Main page	58
6.1.2 Reports page	59
6.1.3 Queries page	61
6.1.4 Contact page	62
6.1.5 Profile page	63
6.2 Access to the repository	66
6.3 Usage of the code locally	66

7. Conclusions	71
7.1 Conclusions	71
7.2 Future of the project	72
8. Licence	73
8.1 Licence	73
9. Bibliography	74
9.1 Bibliography	74

1. Introduction

1.1 Motivations

Normally, the majority of the public financial data of Spanish public facilities is available on the Internet, but it is usually distributed and cannot be found easily, since they are stored in different public webpages, such as Ministerio de Hacienda, Ministerio de Fomento, Ministerio de Empleo (SEPE), INE and Banco de España. Moreover, they are normally confusing as there is not a standardised way to obtain the data, and each public facility presents it in their own style, since they combine different macroeconomics variables. Therefore, it might be relatively difficult for a conventional user to find the information of an autonomous community, council or municipality in a clear and easy stepped way.

1.2 Goals / Description of the work to do

For the reasons proposed in the previous section, it was decided to develop a public and open website that would collect, analyse and display public financial data from all the public facilities in Spain. This would serve as a solution providing transparency while making the data more accessible and easier to consult and obtain in a standardised user-friendly way.

The original application idea was provided specially by the project co-director Esteban Sastre Torregrosa, who has been working with Spanish public data and found difficulties to obtain and analyse it in the past.

As an initial solution to the problem, NosterEconomia collected all the public data from every public facility during several years in a set of Excel spreadsheets, which resulted in a more efficient analysis once it was properly collected and stored. Said set of spreadsheets contain data from all the way back to 2015 until 2021 for the moment. Since the memory is written in 2022, the data from the current year is not yet fully available.

In addition to the data provided by these facilities, Esteban created a set of financial ratios relating different economic values extracted from the raw data. These ratios also supply an easy way of comparison between same-type facilities. Using these ratios, a scoring value was created that ranks each facility in a scale from A to E, allowing the user an easy understanding of the facility economics' situation at a glance.

After all this data collection was executed by Esteban and his team, it was proposed the idea of developing the previously said website which would work as an easy-to-use interface for any potential user who would like to visualise economic data from the Spanish government and its derivatives in a fully automated way.

As a result of several meetings with the director and co-director, it was understood the motivation of the project, goals and objectives. Therefore, a list of requirements and specifications was drafted such that it could be suitable for the project. Said list is as follows:

- **MySQL Database** that will help to computerise all the past and future data in an efficient way
- **Automated database data insertion and update algorithm.**
- **Automated fixed report-based data visualisation system.**
- **Automated PDF generation of the data reports.**
- **Automated Excel spreadsheet generation** with a selection of data from the reports.
- **User-friendly form-based queries interface.**
- **API**
- **Use of a paid hosting service** with the following domain: *analisisfinancierosectorpublico.com*

1.3 Work Plan

Before starting the development of the web application, a work plan had to be designed, taking into account different aspects such as: the time intervals for the project until deadline, an estimation of time that the team members would have to dedicate in order to complete each functionality and the quantity of hours needed for the development.

As a group, it had been decided that the most optimal option would be opting for an agile approach on project management like SCRUM. Mainly because the directors and us value constant communication and weekly meetings that help to make sure the project is being correctly developed. This methodology has also been useful to tweak the time estimations actively if it was needed, in addition to learning how project management works in a real application.

Using the previously drafted list of requirements, a roadmap of functionalities was designed in order to be developed from start to finish, ordered by importance and expectations of the co-director, Esteban. Even so, although a functionality had ended its first approximation, it still could have small adjustments when an error was found or a new approach was considered. By this way, following the SCRUM methodology, the work plan were divided into eight milestones, each one with a set of tasks to develop:

- Milestone/Sprint 1 (02/02/2022) *
 - Show the data of the reports by screen in the autonomous communities, councils and municipalities without format and parameters yet. For instance, charts, colours, etc.
- Milestone/Sprint 2 (09/02/2022) *
 - Show the data with format.
 - Include bar, linear or doughnut charts, in addition to tables to show the information.
- Milestone/Sprint 3 (16/02/2022) *
 - Create the filters in the reports interface (based in years, range of years, etc.).
- Milestone/Sprint 4 (23/02/2022) *
 - Implement PDF functionality into the reports interfaces with the data of the facility.
- Milestone/Sprint 5 (02/03/2022)*
 - Create the queries in the website (queries that cannot be downloaded in a PDF document, that is, that can only be visualised in the web).
- Milestone/Sprint 6 (09/03/2022)
 - Design interactive map
- Milestone/Sprint 7 (23/03/2022)
 - Create a regression model.
 - Implement a regression line in order to make data predictions.
- Milestone/Sprint 8 (06/04/2022, 2-3 weeks)
 - Develop the API.

The asterisks represent the milestones that are mandatory to develop in order to have a deliverable web application. The rest of milestones were optional that could be developed only if there was enough time.

However, during the project development, small changes were made to the original plan. Milestone 3 was discarded since it has a similar functionality as Milestone 5, and Milestone 8 had to be brought forward with respect to Milestone 6 and 7 since its development would be simpler than initially assumed. In addition, Milestones 5 had to be devoted more time since it was seen that it would be more long and complex, since it involved modifying the query depending on the user request. Therefore, at the end of the project, Milestone 6 and 7 could not be done and left for the future. Then, the final work plan would be the following one:

- Milestone/Sprint 1 (02/02/2022) *
 - Show the data of the reports by screen in the autonomous communities, councils and municipalities without format and parameters yet. For instance, charts, colours, etc.
- Milestone/Sprint 2 (09/02/2022) *
 - Show the data with format.
 - Include bar, linear or doughnut charts, in addition to tables to show the information.
- Milestone/Sprint 3 (16/02/2022) *
 - Implement PDF functionality into the reports interfaces with the data of the facility.
- Milestone/Sprint 4 (23/02/2022)*
 - Create the queries in the website (queries that cannot be downloaded in a PDF document, that is, that can only be visualised in the web).
- Milestone/Sprint 5 (16/03/2022)
 - Develop the API.
- Milestone/Sprint 6 (23/03/2022)
 - Design interactive map
- Milestone/Sprint 7 (13/04/2022)
 - Create a regression model.
 - Implement a regression line in order to make data predictions.

To begin with, the provided financial data was interpreted and tried to insert in an initial database, implement iterative models of user interfaces and test the resulting web application. All of this was done while gatherings between the project director and co-director were organised.

1.4 Tools and Technologies

To develop this project, it was used a set of tools and technologies, which are the following ones:

- Programming languages: PHP, Javascript
- Markup language: HTML
- Visual Studio Code: to write the code of the application.
- IONOS server
- Google Drive: tool used to store all the documentation and datasets from Esteban, Luis, Aldair or Miquel.
- Microsoft Excel: the format of the datasets:
- Balsamiq Wireframes: to make initially the prototyping.
- Figma: to make the prototype
- XAMPP: to prove the application locally.
- Postman: to help us test HTTP Requests on the API
- Microsoft Teams: to do the meetings.
- Google Meet: an alternative to do the meetings.
- Github repository (<https://github.com/MiquelVera/AnalisisFinancieroSectorPublico>) and Github Desktop: to manage the project development between the team members.

1.5 Related Subjects

The mechanisms and used knowledge to develop the website were learned in the following subjects:

- **Bases de Datos (BD)**, in order to create an entity-relation model of the database, in addition to the acquired knowledge about SQL language to store, query and manage the database.
- **Ampliación de Bases de Datos (ABD)**, in order to learn about the PHP programming language and how to connect to a SQL database through PHP.
- **Aplicaciones Web (AW)**, for the implementation of the DAOs (Data Access Objects) and for PHP and Javascript programming language learning, in addition to learning how to include styles in a website.
- **Ingeniería del Software (IS)**, since in the group it is implemented an SCRUM from the agile methodology that consists of establishing weekly meetings between the director and co-director and the team members in order to set goals that will be developed throughout the week. In the first semester, all the meetings were organised every Friday in the morning, but due to the change in the calendar, the meetings in the second semester had to be done every Wednesday in the afternoon.
- **Fundamentos de Algoritmos (FAL)**, in order to maintain a constant or linear complexity of the created PHP functions as far as possible. However, there are some functions that were requested to have a quadratic complexity, such as the import scripts.
- **Redes y Seguridad I y II (RySI y II)**, in order to do security tests to detect and solve vulnerabilities in the website, such as possible SQL injections.
- **Ética, Legislación y Profesión (ELP)**. There are some libraries in the website code that are imported from the Internet, such as the library to generate the graphics or the code to read Excel files in PHP. With the acquired knowledge in this subject, it was able to know about the licences of the code, and therefore, select only the external code that is open source in order to implement some functionalities of the website.
- **Diseño de Sistemas Interactivos (DSI)**, in order to create a usable website, with a simple and intuitive interface and design that can be used by any user, in addition to receiving feedback from potential final users.

2. Financial information and concepts

The collected information was obtained from Excel files provided by Esteban.

At the beginning, the files were uploaded to a shared Google Drive folder in which only the director, co-director and the team members have access to. After that, the files were downloaded and stored in the local project folder in order to be used by the import scripts.

Nevertheless, when the import option was implemented to the website, the files were directly uploaded from the website by both the team members or Esteban. This import option is implemented in the *Perfil* section, which is limited to only the authorised users that have an account provided by the administrator.

Once the file is uploaded, the information from the cells from all the pages are read and stored in the corresponding tables in the database.

The Excel files could be of autonomous communities, councils or municipalities. They are composed of five pages:

- The first page contains general information about the facility.
- The second page contains financial information about the facility.
- The third page contains data about the scoring and trend of the facility.
- The fourth page defines all the fields that appear in the three previous pages.
- The fifth page explains in a more structural manner each financial term that is in the Excel file.

Said pages of these Excel files contain a set of columns of different types of information for each row of facility. These columns are described in the following tables:

2.1 Autonomous Communities

General information

<i>Data</i>	<i>Description</i>
CÓDIGO_CCAA	Identificative 2 digit code of the Autonomous Community
NOMBRE_CCAA	Name of the Autonomous Community
POBLACION_AÑO	Population on said year
NOMBREPRESIDENTE	Autonomous Community President's first name
APELLIDO1PRESIDENTE	Autonomous Community President's first last name
APELLIDO2PRESIDENTE	Autonomous Community President's second last name
VIGENCIA	Legislature start date
PARTIDO	Political party
CIF	Tax identification code
TIPOVÍA	Address of the entity
NÚMEROVÍA	
CÓDIGOPOSTAL	

TELÉFONO	Office phone number
FAX	Office fax number
WEB	Official website
MAIL	Official mail
REFPIB	Date reference for the “PIB” data
PIB	Gross Domestic Product
REFPIBC	Date reference for the “PIBC” data
PIBC	Per cápita Gross Domestic Product
REFRESULTADO	Date reference for the “RESULTADO” data
RESULTADO	Economic result
REFDEUDAVIVA	Date reference for the “DEUDAVIVA” data
DEUDAVIVA	Outstanding debt

Financial information

Contains the data organised in different type of blocks of columns, following a pattern:

- **First set:**
 - *Reference: date reference for the data*
 - *Data on date reference*
 - *Data on date reference - 1 (last year)*
 - *Data on date reference - 2 (year before last)*
- **Second set (income item):**
 - *Income item forecast*
 - *Income item initial forecast modification*
 - *Income item definitive forecast*
 - *Income item recognized rights*
 - *Income item current year collection*
 - *Incom item closed fiscal years collection*
- **Third set (expense item):**
 - *Expense item initial credits*
 - *Expense item credits modification*
 - *Expense item total credits*
 - *Expense item recognized obligations*
 - *Expense item current year payments*
 - *Expense item closed fiscal years payments*

Both second and third dates repeat themselves containing data of the current, last and previous to last year. As an example, if the Excel sheet contains data of 2021, these sets will also include the information related to 2020 and 2019 for both income and expense, which result in a total of 749 columns only for the Excel sheet related to financial information.

<i>Type</i>	<i>Data</i>	<i>Description</i>
First	INCREMENTO PIB	Gross Domestic Product Increase
	TASA DE PARO	Unemployment rate
	TRANSACCIONES INMOBILIARIAS	Real estate transactions
	NÚMERO DE EMPRESAS	Number of companies
	PORCENTAJE CCAA EN PIB	Percentage of public deficit
	DEUDA VIVA ENTRE INGRESOS CORRIENTES	Outstanding debt divided by current income
	RATIO SOSTENIBILIDAD FINANCIERA	Financial sustainability ratio
	RATIO DE RIGIDEZ	Stiffness ratio
	RATIO DE SOSTENIBILIDAD DEL ENDEUDAMIENTO	Debt sustainability ratio
	PERIODO MEDIO DE PAGO	Average payment period
	RATIO DE EFICIENCIA	Efficiency ratio
	RATIO DE EJECUCIÓN DE INGRESOS CORRIENTES	Execution ratio of current income
	RATIO DE EJECUCIÓN DE GASTOS CORRIENTES	Execution ratio of current expenses
	RATIO DE DEUDA COMERCIAL PENDIENTE DE PAGO	Commercial debt pending payment ratio
	PAGOS RECONOCIDOS SOBRE OBLIGACIONES	Recognized payments on obligations
RATIO EFICACIA RECAUDATORIA	Collection efficiency ratio	
Second (income items)	IMPUESTOS DIRECTOS	Direct taxes
	IMPUESTOS SOBRE SUCESIONES Y DONACIONES	Taxes on inheritances and donations
	IMPUESTO RENTA PERSONAS FÍSICAS	Personal income tax
	IMPUESTOS INDIRECTOS	Indirect taxes
	ITP Y AJD	Property Transfer Taxes and Documented Legal Acts
	IMPUESTOS SOBRE EL VALOR AÑADIDO (IVA)	VAT (Value Added Tax)
	IMPUESTOS ESPECIALES	Special taxes

	TASAS, PRECIOS PÚBLICOS Y OTROS INGRESOS	Rates, public prices and other incomes
	TRANSFERENCIAS CORRIENTES	Current transfers
	FONDO DE SUFICIENCIA GLOBAL	Global sufficiency fund
	FONDO DE GARANTÍA	Guarantee fund
	FONDOS DE CONVERGENCIA	Convergence fund
	INGRESOS PATRIMONIALES	Estate income
	OPERACIONES CORRIENTES	Current operations
	ENAJENACIÓN INVERSIONES REALES	Disposal of real investments
	TRANSFERENCIAS CAPITAL	Capital transfers
	FONDO DE COMPENSACIÓN INTERTERRITORIAL	Inter-territorial compensation fund
	OPERACIONES DE CAPITAL	Capital operations
	OPERACIONES NO FINANCIERAS	Non-financial operations
	ACTIVOS FINANCIEROS	Financial assets
	PASIVOS FINANCIEROS	Financial liabilities
	TOTAL INGRESOS	Total income
Third (expense items)	GASTOS PERSONAL	Personnel expenses
	GASTOS CORRIENTES EN BIENES Y SERVICIOS	Current expenses on goods and services
	GASTOS FINANCIEROS	Financial expenses
	TRANSFERENCIAS CORRIENTES	Current transfers
	FONDO DE CONTINGENCIA	Contingency fund
	INVERSIONES REALES	Real investments
	TRANSFERENCIAS CAPITAL	Capital transfers
	OPERACIONES DE CAPITAL	Capital operations
	OPERACIONES NO FINANCIERAS	Non-financial operations
	ACTIVOS FINANCIEROS	Financial assets
	PASIVOS FINANCIEROS	Financial liabilities
		TOTAL GASTOS

Scoring

<i>Data</i>	<i>Description</i>
CIF	Tax identification code
CODIGO_CCAA	Identificative 2 digit code of the Autonomous Community
NOMBRE_CCAA	Name of the Autonomous Community
RATING_N-1	A to E Scoring on previous year
TENDENCIA_N-1	Tend on previous year
RATING_N	A to E Scoring of current year
TENDENCIA_N	Tend on current year

2.2 Councils

General information

<i>Data</i>	<i>Description</i>
CÓDIGO_DIP	Identificative 2 digit code of the Council
DIPUTACION	Name of the Council
CIF	Tax identification code
TIPOVÍA	Address of the entity
NÚMEROVÍA	
CÓDIGOPOSTAL	
TELÉFONO	Office phone number
FAX	Office fax number
WEB	Official website
MAIL	Official mail
INGRESOS	Income
FONDLIQUIDOS	Liquid funds
DERPENDCOBRO	Entitlements to be collected
DEUDACOM	Commercial debts
DEUDAFIN	Financial debt

LIQUAJUST	Total adjusted liquidity
INGRESOSCORR	Current incomes
GASTOSCORR	Current expenses

Financial information (idem for Municipality)

Contains the data organised in different type of blocks of columns, following a pattern:

- **First set (income item):**
 - *Income item prevision*
 - *Income item recognized entitlements*
 - *Income item current year's revenues*
- **Second set (expense item):**
 - *Expense item total credits*
 - *Expense item recognized payment obligations*
 - *Expense item current year's payments*

<i>Type</i>	<i>Data</i>	<i>Description</i>
First (income items)	IMPUESTOS DIRECTOS	Direct taxes
	IMPUESTOS INDIRECTOS	Indirect taxes
	TASAS, PRECIOS PÚBLICOS Y OTROS INGRESOS	Fees, public prices and other revenues
	TRANSFERENCIAS CORRIENTES	Current transfers
	INGRESOS PATRIMONIALES	Patrimonial revenues
	ENAJENACIÓN DE INVERSIONES REALES	Disposal of real investments
	TRANSFERENCIAS DE CAPITAL	Capital transfers
	ACTIVOS FINANCIEROS	Financial assets
	PASIVOS FINANCIEROS	Financial liabilities
Second (expense items)	GASTOS DE PERSONAL	Personnel expenses
	GASTOS CORRIENTES EN BIENES Y SERVICIOS	Current expenditure on goods and services
	GASTOS FINANCIEROS	Financial expenses
	TRANSFERENCIAS CORRIENTES	Current transfers
	FONDO DE CONTINGENCIA	Contingency fund
	INVERSIONES REALES	Real investments

	TRANSFERENCIA DE CAPITAL	Capital transfers
	ACTIVOS FINANCIEROS	Financial assets
	PASIVOS FINANCIEROS	Financial liabilities

Scoring (idem for Municipality)

<i>Data</i>	<i>Description</i>
CIF	Identification code of the Council
RATING 1 - <i>NIVEL DE ENDEUDAMIENTO</i>	Level of indebtedness
RATING 2 - <i>RATIO DE SOSTENIBILIDAD FINANCIERA O AHORRO NETO</i>	Financial sustainability ratio or net savings
RATING 3 - <i>APALANCAMIENTO OPERATIVO</i>	Operating leverage
RATING 4 - <i>RATIO DE SOSTENIBILIDAD DEL ENDEUDAMIENTO</i>	Debt sustainability ratio
RATING 5 - <i>REMANENTE DE TESORERÍA PARA GASTOS GENERALES</i>	Cash surplus for general expenses
RATING 6 - <i>RATIO DE LIQUIDEZ INMEDIATA O DISPONIBILIDAD</i>	Immediate liquidity or availability ratio
RATING 7 - <i>RATIO DE SOLVENCIA A CORTO PLAZO</i>	Short-term solvency ratio
RATING 8 - <i>RATIO DE EFICIENCIA</i>	Efficiency ratio
RATING 9 - <i>RATIO DE EJECUCIÓN DE INGRESOS CORRIENTES</i>	Current revenue performance ratio
RATING 10 - <i>RATIO DE EJECUCIÓN DE GASTOS CORRIENTES</i>	Current expense execution ratio
RATING 11 - <i>PERIODO MEDIO DE PAGO A ACREEDORES</i>	Average payment period to creditors
RATING 12 - <i>PAGOS SOBRE OBLIGACIONES RECONOCIDAS</i>	Payments on recognized obligations
RATING 13 - <i>EFICACIA RECAUDATORIA</i>	Collection efficiency
RATING_N-1	A to E Scoring on previous year
TENDENCIA_N-1	Tend on previous year
RATING_N	A to E Scoring of current year
TENDENCIA_N	Tend on current year

2.3 Municipality

General information

<i>Data</i>	<i>Description</i>
CÓDIGO_MUN	Identificative code of the Municipality
CIF_MUNICIPIO	Tax identification code
MUNICIPIO	Name of the Municipality
COD_PROVINCIA	Identificative code of the Province it belongs to
PROVINCIA	Name of the Province it belongs to
AUTONOMIA	Name of the Autonomous Community it belongs to
POBLACION	Population on said year
NOMBREALCALDE	Municipality's President's first name
APELLIDO1ALCALDE	Municipality's President's first last name
APELLIDO2ALCALDE	Municipality's President's second last name
VIGENCIA	Legislature start date
PARTIDO	Political party
TIPOVÍA	Address of the entity
NÚMEROVÍA	
CÓDIGOPOSTAL	
TELÉFONO	Office phone number
FAX	Office fax number
WEB	Official website
MAIL	Official mail
INGRESOS	Income
FONDLIQUIDOS	Liquid funds
DERPENDCOBRO	Entitlements to be collected
DEUDACOM	Commercial debts
DEUDAFIN	Financial debt
LIQUAJUST	Total adjusted liquidity
INGRESOSCORR	Current incomes
GASTOSCORR	Current expenses

3. Pre-development decisions

3.1 Data interpretation and initial decisions

Once the given data structure was properly analysed, it was decided that all of the information would be stored on a brand-new specialised database created specifically for the project, for the following reasons:

- Maximise the access to the data
- Avoid data repetition
- Reduce number of columns to reduce search time on (normalization of the database)
- PHP-friendly queries environment (for both insert and get data)

As it's explained further in this document, an automatic insertion and update algorithm has been implemented in order to allow an easy way to put new information on the website without the trouble of changing the Excel file structure which our director is used to work with.

3.2 Technologies and tools selection

At the beginning, it was decided to parse the Excel information to XML, and afterwards to read the data by PHP scripts and store it in the database. However, this possibility was finally discarded. Instead, it was decided to read the information directly from the Excel file from PHP itself by using an external open-source library called *PHPSpreadSheet*. This library is implemented in the *includes* folder (Fig.01) and it is used by the scripts of the *imports* folder for the file imports and the *exports* folder for file exports.

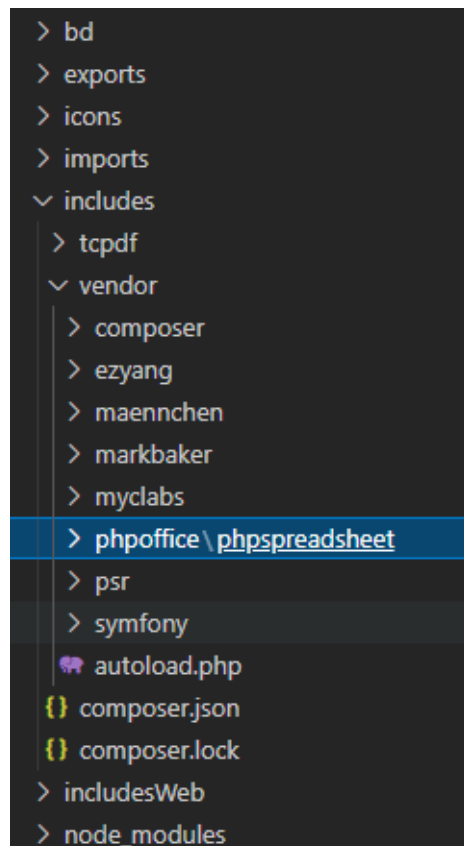


Fig.01: Location of PHPSpreadSheet in the project

On the other hand, other tools were used during project development for different purposes, such as:

- **Github**, which was used to submit the code and its periodic modifications. The repository is public.
- **XAMPP** as a local server in order to make different tests and check the implementations done before submitting the final versions before the meetings.
- **Visual Studio Code** as the main tool to program the PHP scripts, pages and styles of the website.
- Teams in order to make the meetings between the members of the project group and the directors.
- **Google Drive**, which was used by Esteban to provide documentation, datasets and comments about the web application. Besides, it was also used by the team members to write the memory and create the entity-relation model of the database.
- **Github Desktop** to commit all the modifications of the code from the local repository to the main Github repository, in addition to managing all the commits and pulls.
- **IONOS**, which is the server where the website is hosted. Once the modifications and corrections were implemented and committed to the Github repository, the project was compressed and uploaded to IONOS. Furthermore, it was also possible to create and manage an SQL database inside the server in order to store all the information.
- **Figma**, used to create a sketch of the website interface.
- **Postman** to help us test HTTP Requests on the API

4. Development

4.1 Database design

The design went through several phases, modifying the entities and relations between them, in addition to adding and removing fields, until getting the final version (Fig.02).

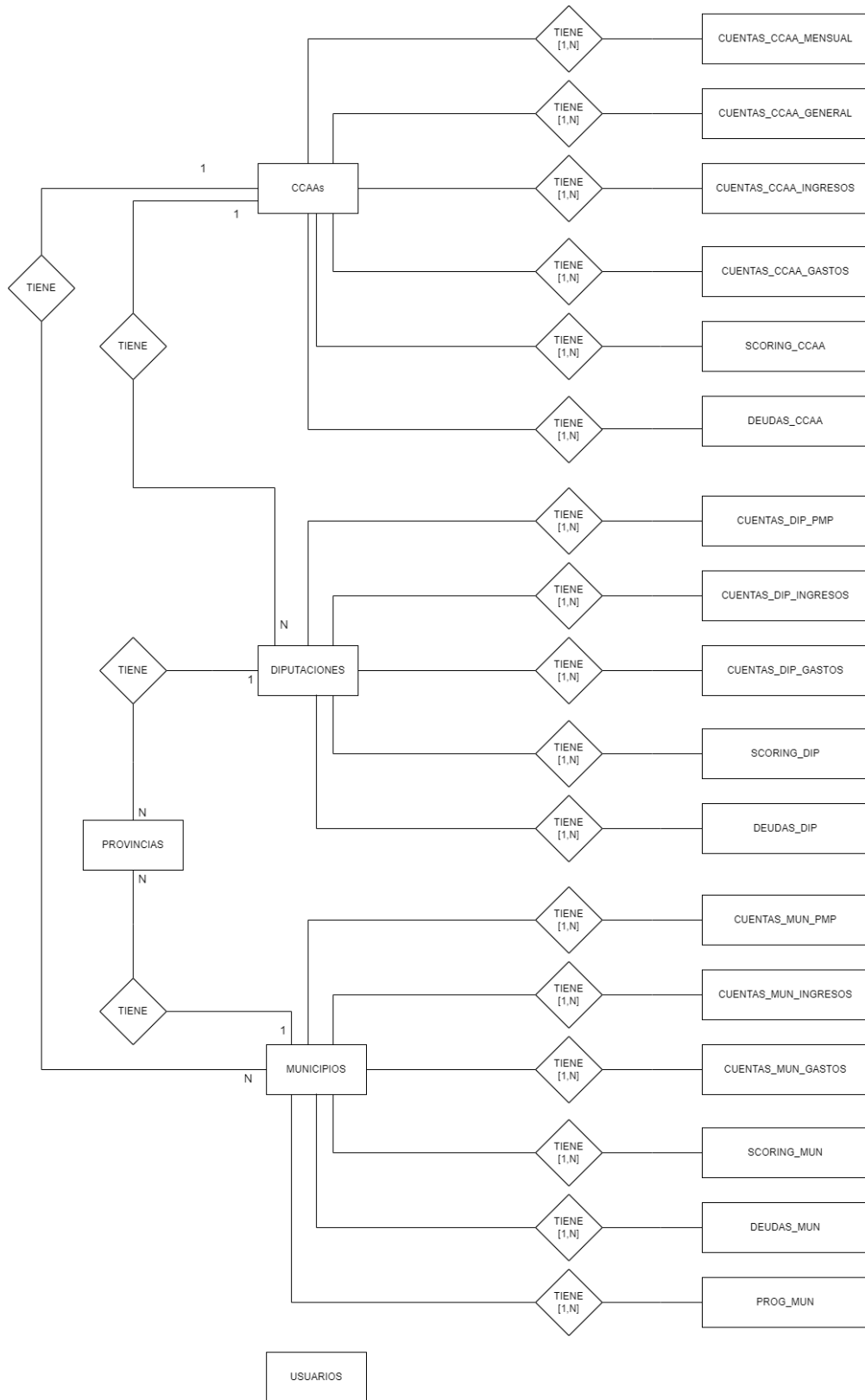


Fig.02: Entity-relation model

There are three main tables, which are CCAAs, DIPUTACIONES and MUNICIPIOS. Each one establishes a [1:N] relation with the corresponding tables.

- CCAAs with:
 - CUENTAS_CCAA_MENSUAL
 - CUENTAS_CCAA_GENERAL
 - DEUDAS_CCAA
 - SCORING_CCAA
 - CUENTAS_CCAA_INGRESOS
 - CUENTAS_CCAA_GASTOS
 - DIPUTACIONES (since one autonomous community can have more than one council, but a council can belong to only one autonomous community)
 - MUNICIPIOS (for the same reason as the councils).
- DIPUTACIONES with:
 - CUENTAS_DIP_PMP
 - DEUDAS_DIP
 - SCORING_DIP
 - CUENTAS_DIP_INGRESOS
 - CUENTAS_DIP_GASTOS
 - PROVINCIAS (since one council can have more than one council, but a province can belong to only one council).
- MUNICIPIOS with:
 - CUENTAS_MUN_PMP
 - DEUDAS_MUN
 - SCORING_MUN
 - CUENTAS_MUN_INGRESOS
 - CUENTAS_MUN_GASTOS
 - PROVINCIAS (since one municipality can have more than one council, but a province can belong to only one council).

Apart from that, there is an USUARIOS table where the information of the authorised users are stored. Each entity will contain the following fields:

PROVINCIAS (2)

CÓDIGO
NOMBRE

CCAAAs (16)

CÓDIGO
NOMBRE

NOMBRE DEL PRESIDENTE
PRIMER APELLIDO DEL PRESIDENTE
SEGUNDO APELLIDO DEL PRESIDENTE
VIGENCIA
PARTIDO
CIF
TIPO VIA
NOMBRE VIA
NUMERO DE VIA
COD_POSTAL
TELÉFONO
FAX
WEB
MAIL

DEUDAS_CCAA (6)

CÓDIGO
AÑO
PIB
PIBC
RESULTADO

SCORING_CCAA (5)

CÓDIGO
AÑO
POBLACIÓN

SCORING
TENDENCIA

CUENTAS_CCAA_GENERAL (12)

CÓDIGO
AÑO
INCREMENTO PIB
NÚMERO DE EMPRESAS EN LA FECHA DE REFERENCIA
PORCENTAJE DE LA CCAA EN EL PIB
RATIO DE SOSTENIBILIDAD FINANCIERA
RATIO DE EFICACIA
RATIO DE RIGIDEZ
RATIO DE SOSTENIBILIDAD DEL ENDEUDAMIENTO
RATIO DE EJECUCIÓN DE INGRESOS CORRIENTES
RATIO DE EJECUCIÓN DE GASTOS CORRIENTES
PAGOS RECONOCIDOS SOBRE OBLIGACIONES
RATIO DE EFICACIA RECAUDATORIA

CUENTAS_CCAA_GENERAL_MENSUAL (9)

CÓDIGO
AÑO
MES
TASA DE PARO (TRIMESTRAL)
PERIODO MEDIO DE PAGO (PMP) (TRIMESTRAL)
RATIO DE DEUDA COMERCIAL PENDIENTE DE PAGO (TRIMESTRAL)
DEUDAVIVA (MENSUAL)

RATIO DE DEUDA VIVA ENTRE INGRESOS CORRIENTES (TRIMESTRAL)

TRANSACCIONES INMOBILIARIAS (TRIMESTRAL)

CUENTAS_CCAA_INGRESOS (9)

CÓDIGO
AÑO
TIPO
PREV_INI
MOD_PREV_INI
PREV_DEF
DER_REC
RECAUDA_COR
RECAUDA_CER

CUENTAS_CCAA_GASTOS (9)

CÓDIGO
AÑO
TIPO
CRED_INI
MOD_CRED_INI
CRED_TOT
OBLG_REC
PAGOS_COR
PAGOS_CER

DIPUTACIONES (13)

CÓDIGO
NOMBRE
CIF
TIPO_VIA
NOMBRE_VIA
NUM_VIA
COD_POSTAL
PROVINCIA
AUTONOMÍA
TELÉFONO
FAX
WEB
MAIL

DEUDAS_DIP (10)

CÓDIGO
AÑO
INGRESOS
FONDLÍQUIDOS
DERPENDCOBRO
DEUCACOM
DEUDAFIN
LIQUAJUST
INGRESOSCORR
GASTOSCORR

SCORING_DIP (30)

CÓDIGO
AÑO
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R1_NAC
R2_NAC
R3_NAC
R4_NAC
R5_NAC
R6_NAC
R7_NAC
R8_NAC
R9_NAC
R10_NAC
R11_NAC
R12_NAC
R13_NAC

SCORING
TENDENCIA

CUENTAS_DIP_INGRESOS (6)

CÓDIGO
AÑO
TIPO
PRES
DERE
RECA

CUENTAS_DIP_GASTOS (6)

CÓDIGO
AÑO
TIPO
PRES
OBLG
PAGOS

CUENTAS_DIP_PMP (4)

CÓDIGO
AÑO
TRIMESTRE
PMP

MUNICIPIOS (18)

CÓDIGO
NOMBRE
CIF
PROVINCIA
AUTONOMÍA
NOMBREALCALDE
APELLIDO1ALCALDE
APELLIDO2ALCALDE
VIGENCIA
PARTIDO
TIPOVIA
NOMBREVIA
NUMVIA
CODPOSTAL
TELÉFONO
FAX
WEB
MAIL

DEUDAS_MUN (10)

CÓDIGO
AÑO
INGRESOS
FONDLÍQUIDOS
DERPENDCOBRO
DEUCACOM

DEUDAFIN
LIQUAJUST
INGRESOSCORR
GASTOSCORR

SCORING_MUN (31)

CÓDIGO
AÑO
POBLACIÓN
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R1_NAC
R2_NAC
R3_NAC
R4_NAC
R5_NAC
R6_NAC

R7_NAC
R8_NAC
R9_NAC
R10_NAC
R11_NAC
R12_NAC
R13_NAC
SCORING
TENDENCIA

CUENTAS_MUN_INGRESOS (6)

CÓDIGO
AÑO
TIPO
PRES
DERE
RECA

CUENTAS_MUN_GASTOS (6)

CÓDIGO
AÑO
TIPO
PRES
OBLG
PAGOS

CUENTAS_MUN_PMP (6)

CÓDIGO
AÑO
TRIMESTRE
PMP
PARO
TRANSAC_INMOBILIARIAS

PROG_MUN (33)

CÓDIGO
AÑO
Administración General de la Seguridad y Protección Civil (AGSPC)
Seguridad Y Orden Público (SOP)
Ordenación Del Tráfico y del Estacionamiento (OTE)
Movilidad Urbana (MU)
Protección Civil (PC)
Servicio de Prevención y Extinción de Incendios (SPEI)
Promoción y Gestión de Vivienda de Protección Pública (PGVPP)
Conservación y Rehabilitación de la Edificación (CRE)
Pavimentación de Vías Públicas (PVP)
Alcantarillado (A)
Recogida, Gestión y Tratamiento de Residuos (RGTR)
Recogida de Residuos (RR)
Gestión de Residuos Sólidos Urbanos (GRSU)
Tratamiento de Residuos (TR)
Limpieza Viaria (LV)
Cementerio y Servicios Funerarios (CSF)
Alumbrado Público (AP)

Parques y Jardines (PJ)
Pensiones (P)
Servicios Sociales y Promoción Social (SSPS)
Fomento del Empleo (FE)
Sanidad (S)
Educación (E)
Cultura (C)
Deporte (D)
Agricultura, Ganadería y Pesca (AGP)
Industria y Energía (IE)
Comercio (COM)
Transporte Público (TP)
Infraestructura del Transporte (IT)
Investigación, Desarrollo e Innovación (IDI)

USUARIOS(4)

CORREO
NOMBRE
CONTRASEÑA
ROL

For each entity, the following primary and foreign keys were established.

Legend

PRIMARY KEY

FOREIGN KEY

PRIMARY AND FOREIGN KEY

PROVINCIAS

PROVINCIAS (CÓDIGO, NOMBRE)

PRIMARY KEY: CÓDIGO

CCAAs

CCAA_s (CÓDIGO, NOMBRE, NOMBRE_PRESIDENTE, APELLIDO1_PRESIDENTE, APELLIDO2_PRESIDENTE, VIGENCIA, PARTIDO, CIF, TIPO_VIA, NOMBRE_VIA, NUM_VIA, COD_POSTAL, TELEFONO, FAX, WEB, MAIL)

PRIMARY KEY: CÓDIGO

DEUDAS_CCAA

DEUDAS_CCAA (CÓDIGO, AÑO, PIB, PIBC, RESULTADO, DEUDAVIVA)

PRIMARY KEY: CÓDIGO, AÑO

FOREIGN KEY: CÓDIGO (CCAA_s → CÓDIGO)

SCORING_CCAA

SCORING_CCAA (CÓDIGO, AÑO, POBLACION, RATING, TENDENCIA)

PRIMARY KEY: CÓDIGO, AÑO

FOREIGN KEY: CÓDIGO (CCAA_s → CÓDIGO)

CUENTAS_CCAA_GENERAL

CUENTAS_CCAA_GENERAL (CÓDIGO, AÑO, TRIMESTRE, INCR_PIB, N_EMPRESAS, CCAA_PIB, R_SOSTE_FINANCIERA, R_EFIC, R_RIGIDEZ, R_SOSTE_ENDEUDA, R_EJE_INGR_CORR, R_EJE_GASTOS_CORR, PAGOS_OBLIGACIONALES, RATIO DE EFICACIA RECAUDATORIA)

PRIMARY KEY: CÓDIGO, AÑO, TRIMESTRE

FOREIGN KEY: CÓDIGO (CCAA_s → CÓDIGO)

CUENTAS_CCAA_GENERAL_MENSUAL

CUENTAS_CCAA_MENSUAL (CÓDIGO, AÑO, MES, PARO, PMP, R_DCPP, DEUDAVIVA, DEUDA_INGR_COR, TRANSAC_INMOBILIARIAS)

PRIMARY KEY: CÓDIGO, AÑO, MES

FOREIGN KEY: CÓDIGO (CCAA_s → CÓDIGO)

CUENTAS_CCAA_INGRESOS

CUENTAS_CCAA_INGRESOS (CÓDIGO, AÑO, TIPO, PREV_INI, MOD_PREV_INI, PREV_DEF, DER_REC, RECAUDA_COR, RECAUDA_CER)

PRIMARY KEY: CÓDIGO, AÑO, TIPO

FOREIGN KEY: CÓDIGO (CCAA_s → CÓDIGO)

CUENTAS_CCAA_GASTOS

CUENTAS_CCAA_GASTOS (CÓDIGO, AÑO, TIPO, CRED_INI, MOD_CRED, CRED_TOT, OBLG_REC, PAGOS_COR, PAGOS_CER)

PRIMARY KEY: CÓDIGO, AÑO, TIPO

FOREIGN KEY: CÓDIGO (CCAA_s → CÓDIGO)

DIPUTACIONES

DIPUTACIONES (CÓDIGO, NOMBRE, CIF, TIPOVIA, NOMBREVIA, NUMVIA, CODPOSTAL, PROVINCIA, AUTONOMÍA, TELEFONO, FAX, WEB, MAIL)

PRIMARY KEY: CÓDIGO

FOREIGN KEY: PROVONCIA (PROVINCIAS → CÓDIGO), AUTONOMÍA (CCAA_s → CÓDIGO)

DEUDAS_DIP

DEUDAS_DIP (CÓDIGO, AÑO, INGRESOS, FONDLIQUIDOS, DERPENDCOBRO, DEUDACOM, DEUDAFIN, LIQUAJUST, INGRESOSCORR, GASTOSCORR)

PRIMARY KEY: CÓDIGO, AÑO

FOREIGN KEY: CÓDIGO (DIPUTACIONES → CÓDIGO)

SCORING_DIP

SCORING_DIP (CÓDIGO, AÑO, R1-13, R1-13_NAC, RATING, TENDENCIA)

PRIMARY KEY: CÓDIGO, AÑO

FOREIGN KEY: CÓDIGO (DIPUTACIONES → CÓDIGO)

CUENTAS_DIP_INGRESOS

CUENTAS_DIP_INGRESOS (CÓDIGO, AÑO, TIPO, PRES, DERE, RECA)

PRIMARY KEY: CÓDIGO, AÑO, TIPO

FOREIGN KEY: CÓDIGO (DIPUTACIONES → CÓDIGO)

CUENTAS_DIP_GASTOS

CUENTAS_DIP_GASTOS (CÓDIGO, AÑO, TIPO, PRES, OBLG, PAGOS)

PRIMARY KEY: CÓDIGO, AÑO, TIPO

FOREIGN KEY: CÓDIGO (DIPUTACIONES → CÓDIGO)

CUENTAS_DIP_PMP

CUENTAS_DIP_PMP (CÓDIGO, AÑO, TRIMESTRE, PMP)

PRIMARY KEY: CÓDIGO, AÑO, TRIMESTRE

FOREIGN KEY: CÓDIGO (DIPUTACIONES → CÓDIGO)

MUNICIPIOS

MUNICIPIOS (CÓDIGO, CIF, NOMBRE, PROVINCIA, AUTONOMÍA, NOMBREALCALDE, APELLIDO1ALCALDE, APELLIDO2ALCALDE, VIGENCIA, PARTIDO, TIPOVIA, NOMBRE VIA, NUMVIA, CODPOSTAL, TELEFONO, FAX, WEB, MAIL)

PRIMARY KEY: CÓDIGO

FOREIGN KEY: PROVINCIA (PROVINCIAS → CÓDIGO), AUTONOMÍA (CCAA's → CÓDIGO)

DEUDAS_MUN

DEUDAS_MUN (CÓDIGO, AÑO, INGRESOS, FONDLIQUIDOS, DERPENDCOBRO, DEUDACOM, DEUDAFIN, LIQUAJUST, INGRESOSCORR, GASTOSCORR)

PRIMARY KEY: CÓDIGO, AÑO

FOREIGN KEY: CÓDIGO (MUNICIPIOS → CÓDIGO)

SCORING_MUN

SCORING_MUN (CÓDIGO, AÑO, R1-13, R1-13_NAC, RATING, TENDENCIA)

PRIMARY KEY: CÓDIGO, AÑO

FOREIGN KEY: CÓDIGO (MUNICIPIOS → CÓDIGO)

CUENTAS_MUN_INGRESOS

CUENTAS_MUN_INGRESOS (CÓDIGO, AÑO, TIPO, PRES, DERE, RECA)

PRIMARY KEY: CÓDIGO, AÑO, TIPO

FOREIGN KEY: CÓDIGO (MUNICIPIOS → CÓDIGO)

CUENTAS_MUN_GASTOS

CUENTAS_MUN_GASTOS (CÓDIGO, AÑO, TIPO, PRES, OBLG, PAGOS)

PRIMARY KEY: CÓDIGO, AÑO, TIPO

FOREIGN KEY: CÓDIGO (MUNICIPIOS → CÓDIGO)

CUENTAS_MUN_PMP

CUENTAS_MUN_PMP (CÓDIGO, AÑO, TRIMESTRE, PMP, PARO, TRANSAC_INMOBILIARIAS)

PRIMARY KEY: CÓDIGO, AÑO, TRIMESTRE

FOREIGN KEY: CÓDIGO (MUNICIPIOS → CÓDIGO)

PROG_MUN

PROG_MUN (CÓDIGO, AÑO, AGSPC, SOP, OTE, MU, PC, SPEI, PGVPP, CRE, PVPA, RGTR, RR, GRSU, TR, LV, CSF, AP, PJ, P, SSPS, FE, S, E, D, C, AGP, IE, COM, TP, IT, IDI)

PRIMARY KEY: CÓDIGO, AÑO,

FOREIGN KEY: CÓDIGO (MUNICIPIOS → CÓDIGO)

USUARIOS

USUARIOS (CORREO, NOMBRE, CONTRASEÑA, ROL)

PRIMARY KEY: CORREO

Before reaching this database solution, the design had a series of phases trying to normalize and improve the data structure, minimizing the number of columns for each table. In order to execute these improvements, a set of decisions were taken as follow:

- **Single foreign key:** some Excel sheets have repeated identification data (CIF, Code, Name). In order to reduce the number of columns and construct an efficient query table relation graph, the identification column was reduced to the facility “code”.
- **Subdivide a single table into different smaller tables:** taking into account the data information and structure.
- **Table growth should be vertical, not horizontal:** in the Excel file, each time a new year related information wanted to be added for a given data, a new column was created following the pattern “data_year”. In order to have a reduced number of columns, for this structure we just need two columns to store any given data (“data” and “year”). This way, adding new data results in an increment of rows, and not columns.
- **Group information with the same kind of temporal relation to each data.** The different data have two kinds of temporality: *year* and *trimestral*. Each kind of data had to be stored on a different kind of table to reduce the number of columns.
- **Take advantage of patterns on financial information Excel sheets:** these spreadsheets contain a repetition pattern for income and another one for expenses for each party, which resulted in a repetition of same-kind columns only changing the party number (from 1 to 9 usually). As a solution, only 7 columns were created: one for the identification of the party number for either income or expenses; and 6 for the repetitive pattern

As a proof to these improvements, one of the largest spreadsheets, such as *financial information of Autonomous Communities*, was reduced from a single 749 column table into 4 tables adding up to 42 columns. An improvement like this reduced the column number by 94,39%.

4.2 DAO model (*Decentralised Autonomous Organisation*)

When interacting with the data within our application, both to collect it from Excel files or from the DB in order to show them on the screen, a series of specific classes was created using the DAO (*Decentralised Autonomous Organisation*) model.

At first, a set of classes was created, one for each kind of entity (*CCAA - Autonomous Community, Council, Municipality, ...*). In addition, other DAO classes were developed as an intermediate communication between the Excel files and the Database, and between the Database and the user interface.

These DAO Classes are the following:

- **DAOCargador.php:** identifies the entity of the excel file that has been uploaded and derives the data to its corresponding import script (*detailed later on this document*).
- **DAOConsultor.php:** central query class. When a query is received, it's referred to the corresponding DAO class, depending on the entity to which the query has been made, (*DAOConsultorCCAA, DAOConsultorDiputación, ...*).
- **DAOConsultorCCAA.php:** handles Autonomous Communities' queries.
- **DAOConsultorDiputacion.php:** handles Councils' queries.
- **DAOConsultorMunicipio.php:** handles Municipalities' queries.
- **DAOConsultorProvincia.php:** handles Provinces' queries.
- **DAOUsuario.php:** handles Users' queries.

As observed in the previous list, there is a DAO class dedicated to the users of the system, who are the only ones who can access a simple and dedicated control panel where they are given the possibility of uploading said Excel files to insert or update the data. These users are closed and the web does not allow them to be created through it, but they have been added manually into the database so that there are no unwanted and uncontrolled insertions.

4.3 Algorithm for automatic insertion and update of data in the database

The procedure for inserting and updating data in the database is done through scripts located in the *imports* folder (Fig.03).

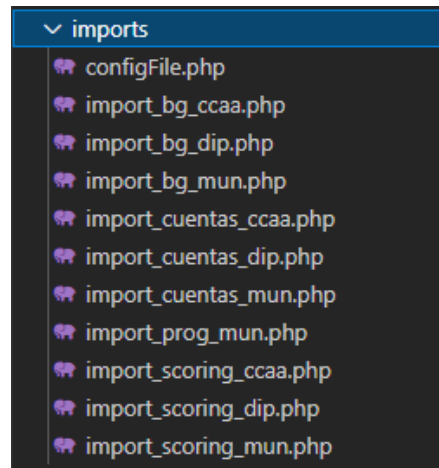


Fig.03: Scripts from *imports* folder

The content of the scripts is as follows:

- **configFile.php**: contains the configuration parameters of all scripts in general.
- **Import_bg_ccaa.php**: class that inserts or updates the data from the *BLOQUE GENERAL* sheet of the Excel of autonomous communities.
- **Import_bg_dip.php**: class that inserts or updates the data from the *BLOQUE GENERAL* sheet of the Excel of councils.
- **Import_bg_mun.php**: class that inserts or updates the data from the *BLOQUE GENERAL* sheet of the Excel of municipalities.
- **Import_cuentas_ccaa.php**: class that inserts or updates the data from the *CUENTAS* sheet of the Excel of autonomous communities.
- **Import_cuentas_dip.php**: class that inserts or updates the data from the *CUENTAS* sheet of the Excel of councils.
- **Import_cuentas_mun.php**: class that inserts or updates the data from the *CUENTAS* sheet of the Excel of municipalities.
- **Import_prog_mun.php**: class that inserts or updates the data of the *PROGRAMA DE GASTO* sheet of the municipalities spending program Excel.
- **Import_scoring_ccaa.php**: class that inserts or updates the data from the *SCORING* sheet of the Excel of autonomous communities.
- **Import_scoring_dip.php**: class that inserts or updates the data from the *SCORING* sheet of the Excel of autonomous councils.
- **Import_scoring_mun.php**: class that inserts or updates the data from the *SCORING* sheet of the Excel of autonomous municipalities.

These scripts are used in the import functionality from the website. This option is located in the control panel of the profile page, and to access it, the user must log in. Once there, the Excel file to be imported is added and the *Import* button is clicked.

In this way, the website is redirected to the file *procesarSubida.php*, where the *carga* function of the *DAOCargador* object is called. Within this function, the uploader will use the corresponding imports depending on the type of Excel file, whether it is from autonomous communities, councils or municipalities (Fig.04).

```

public function carga($file){
    if($this->checkFile($file)){
        $cargador = Cargador::getInstance();
        $cargador->setPath($file);
        $tmp_name = ($cargador->getPath()['tmp_name']);
        $realname = ($cargador->getPath()['name']);
        $filenamestr = explode('_',(explode('.', $realname))[0]);
        if(strtolower($filenamestr[2])=='ccaa'){
            if($this->import_ccaas($tmp_name, $realname)){
                return true;
            }
        }
        else if(strtolower($filenamestr[2])=='dip'){
            if($this->import_dips($tmp_name, $realname)){
                return true;
            }
        }
        else if(strtolower($filenamestr[2])=='mun'){
            if(strtolower($filenamestr[4])=='prog'){
                if($this->import_prog($tmp_name, $realname)){
                    return true;
                }
            }
            else if($this->import_muns($tmp_name, $realname)){
                return true;
            }
        }
    }
    return false;
}

```

Fig.04: Load function from DAOCargador.php

Each import is private, since the loader is the only object that should have access to it.

```

private function import_ccaas($tmp_name,$realname){
    if((new Importer_bg_ccaa()->import_bg_ccaa($tmp_name) && (new Importer_cuentas_ccaa()->import_cuentas_ccaa($tmp_name)&&(new Importer_scoring_ccaa()->import_scoring_ccaa($tmp_name, $realname))){
        return true;
    }
    return false;
}

```

Fig.05: Import function for autonomous communities in DAOCargador

```

private function import_dips($tmp_name,$realname){
    if((new Importer_bg_dip()->import_bg_dip($tmp_name) && (new Importer_cuentas_dip()->import_cuentas_dip($tmp_name)&&(new Importer_scoring_dip()->import_scoring_dip($tmp_name, $realname))){
        return true;
    }
    return false;
}

```

Fig.06: Import function for councils in DAOCargador

```

private function import_muns($tmp_name,$realname){
    if((new Importer_bg_mun()->import_bg_mun($tmp_name, $realname) && (new Importer_cuentas_mun()->import_cuentas_mun($tmp_name)&&(new Importer_scoring_mun()->import_scoring_mun($tmp_name, $realname))){
        return true;
    }
    return false;
}

```

Fig.07: Import function for municipalities in DAOCargador

```

private function import_muns($tmp_name,$realname){
    if((new Importer_bg_mun()->import_bg_mun($tmp_name, $realname) && (new Importer_cuentas_mun()->import_cuentas_mun($tmp_name)&&(new Importer_scoring_mun()->import_scoring_mun($tmp_name, $realname))){
        return true;
    }
    return false;
}

```

Fig.08: Import function for the spending program of the municipalities in DAOCargador

Within each import class the PHPSpreadSheet library is used, loading the Excel file with the *load* function of the IOFactory class (Fig.09).

Then, by means of two for loops, the script will go through row by row in each iteration to save the data in the array *fields* in case it is the first row of the Excel, and in *values* in the rest of iterations. It should be noted that, before saving the data in the array, a character cleanup is performed to remove all its special characters (Fig.10).

```
//Cargamos el archivo en la variable de documento "doc"
$path = $filename;
$doc = IOFactory::load($path);

$hoja = $doc->getSheet(0);

$rows = $hoja->getHighestDataRow();
$cols = $hoja->getHighestDataColumn();
$cols = Coordinate::columnIndexFromString($cols);
```

Fig.09: Loading the Excel document

```
for($x = 1; $x < $rows + 1; $x++){
    for($y = 1; $y <= $cols; $y++){
        $valor = $hoja->getCellByColumnAndRow($y,$x);
        //Para eliminar espacios vacios y enters (limpieza de caracteres del archivo excel)
        $valor = preg_replace('/\s+/', ' ', html_entity_decode(preg_replace('/_x{0-9a-fA-F}{4}_/', '&#x$1;', $valor)));
        if($x>1)
            $values[$y-1]=$valor;
        else
            $fields[$y-1]=strtoupper($valor);
    }
}
```

Fig.10: Reading Excel data

Once the script finishes reading a row, if it is not the first one, then it will start saving the data of the array *values* in variables to later insert them in the database (Fig.11).

On the other hand, according to Esteban, Excel files of the same type will always have the same structure, so there is no risk of script failure due to a possible format change.

```

if($x>1 && $values[0]!="") {

    $CODIGO_CCAA = $values[0];
    $NOMBRE_CCAA = addslashes($values[1]);

    $NOMBREPRESIDENTE = addslashes($values[3]);
    $APELLIDO1PRESIDENTE = addslashes($values[4]);
    $APELLIDO2PRESIDENTE = addslashes($values[5]);

    if($values[6]!=""){
        $tiempo = date_create_from_format("d/m/Y",$values[6]);
        $VIGENCIA = date_format($tiempo,"Y/m/d");
    }
    else{
        $VIGENCIA=$values[6];
    }

    $PARTIDO = addslashes($values[7]);
    $CIF = $values[8];
    $TIPOVIA = addslashes($values[9]);
    $NOMBREVIA = addslashes($values[10]);
    $NUMVIA = $values[11];
    $CODPOSTAL = $values[12];
    $TELEFONO = $values[13];
    $FAX = $values[14];
    $WEB = $values[15];
    $MAIL = $values[16];
}

```

Fig.11: Storage of Excel data into variables in import_bg_ccaa.php

To insert or update the information in the database, the script first checks if there is a row in the database with the same primary keys as the row to be inserted. If it already exists in the database, then it proceeds to update the existing row, and if not, it inserts it (Fig.12).

```

$sql = "SELECT CODIGO FROM ccaas WHERE CODIGO='$CODIGO_CCAA'";
$result = mysqli_query($conn,$sql);
if(!$result){
    echo mysqli_error($conn);
    return false;
}
if(mysqli_num_rows($result)==0){
    $insert = "INSERT INTO ccaas VALUES ('$CODIGO_CCAA','$NOMBRE_CCAA',NULLIF('$NOMBREPRESIDENTE',''),NULLIF('$APELLIDO1PRESIDENTE',''),NULLIF('$APELLIDO2PRESIDENTE',''),
    NULLIF('$VIGENCIA',''),NULLIF('$PARTIDO',''),NULLIF('$CIF',''),NULLIF('$TIPOVIA',''),NULLIF('$NOMBREVIA',''),NULLIF('$NUMVIA',''),NULLIF('$CODPOSTAL',''),
    NULLIF('$TELEFONO',''),NULLIF('$FAX',''), NULLIF('$WEB',''),NULLIF('$MAIL',''))";
    if (!mysqli_query($conn, $insert)) {
        echo mysqli_error($conn);
        return false;
    }
}
else {
    //si ya existe la fila, entonces se actualiza el valor del campo con el nuevo valor dado del documento ($value)
    $update = "UPDATE ccaas SET NOMBRE = NULLIF('$NOMBRE_CCAA',''), NOMBRE_PRESIDENTE = NULLIF('$NOMBREPRESIDENTE',''), APELLIDO1_PRESIDENTE = NULLIF('$APELLIDO1PRESIDENTE',''),
    APELLIDO2_PRESIDENTE = NULLIF('$APELLIDO2PRESIDENTE',''),VIGENCIA = NULLIF('$VIGENCIA',''),PARTIDO = NULLIF('$PARTIDO',''),CIF = NULLIF('$CIF',''),
    TIPO_VIA = NULLIF('$TIPOVIA',''), NOMBRE_VIA = NULLIF('$NOMBREVIA',''),NUM_VIA = NULLIF('$NUMVIA',''),COD_POSTAL = NULLIF('$CODPOSTAL',''),
    TELEFONO = NULLIF('$TELEFONO',''),FAX = NULLIF('$FAX',''), WEB = NULLIF('$WEB',''),MAIL = NULLIF('$MAIL','') WHERE CODIGO = '$CODIGO_CCAA'";
    if (!mysqli_query($conn, $update)) {
        echo mysqli_error($conn);
        return false;
    }
}
}

```

Fig.12: Insertion or update in the database in import_bg_ccaa.php

Once finished updating or inserting the data in the DB, the array *values* is reset and the same process is repeated with the remaining rows until the imported Excel file is inserted or updated. If there was no error, the function will return a true value, otherwise, it will display an error message.

4.4 High fidelity prototype

After all the data had been inserted into the database, the report system started its development with a basic and simple interface, just to make sure the desired data was shown properly without any particular design.

Meanwhile, a series of iterations were made regarding the graphic design of the website in order to make sure the final design was decided before its implementation. These iterations were made using the High Fidelity Prototype Software: *Figma*. This software allowed a non-programming and easy to share solution to present the design, without making the potential changes a long and frustrating labour, since no programming is needed.

Firstly, a basic prototype was proposed to the project co-director and the design was developed from there on as a starting point. The following figure represents the initial design proposition:



Fig.13: index page prototype proposition using Figma

The selected colour palette is based on a set of blue tones, combined with a white clean background. This white background is related to transparency, a key concept for the website itself. The blue tones are used over to stand out. Also, blue colours are used on many government companies, specially because of its meanings related to security, calm, confidence, loyalty and trustworthiness

As previously said, the design suffered a numerous amount of changes on its data representation and location, thanks to the constant iterations; but not on the colour palette, since it became a general consensus.

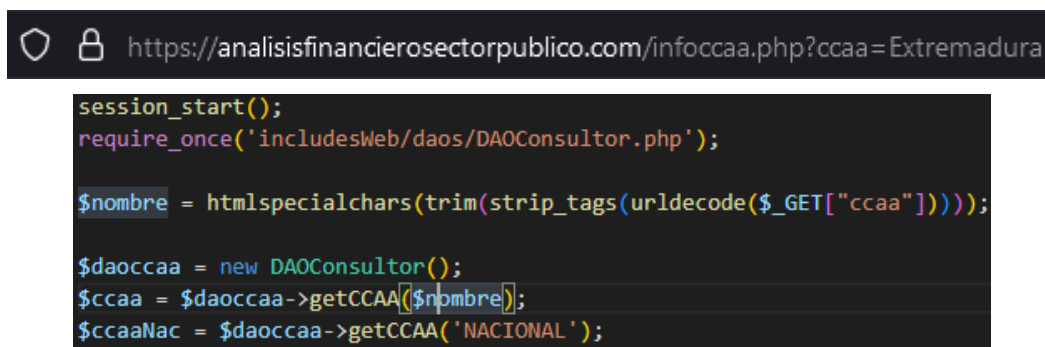
Some of these changes were to put the search bar on top of the website, make the menu and footer fixed while scrolling, and a small tweak on the soft blues.

Further in this document, the final design of the website is shown.

4.5 Reports

To generate the reports, it was done in two different ways.

In case of autonomous communities, the functionality is implemented in *infoccaa.php*, and before loading the HTML code, it is needed to obtain the information of the requested autonomous community. For this, the page receives the name through the URL and collects it by using the global variable GET, and in order to avoid possible attacks such as inserting malicious code in the variable, the special characters are cleaned. Then, *DAOConsultor* is called and this in turn calls the *DAOConsultorCCAA*, which is in charge of accessing the database to obtain the general and financial information of the autonomous community, a *CCAA* object is created and it is returned to *DAOConsultor*, and later to the *infoccaa.php* web page (Fig.14).



```
session_start();
require_once('includesWeb/daos/DAOConsultor.php');

$nombre = htmlspecialchars(trim(strip_tags(urldecode($_GET["ccaa"]))));

$daoccaa = new DAOConsultor();
$ccaa = $daoccaa->getCCAA($nombre);
$ccaaNac = $daoccaa->getCCAA('NACIONAL');
```

Fig.14: infoccaa file. Getter function for the facility and the country

Finally, the page loads the necessary variables in order to generate the graphics by using the Chartjs library, and load the HTML content.

In the case of councils and municipalities, the report generation script is slightly different. These reports are generated through *infomunicipio.php* and *infodiputacion.php*. Both entities share similar code structure since the data to be shown is basically the same, except the general information. For that reason, when detailing one's structure and functionality, the other one ends up being described too.

Firstly, the script detects the entity name through the url just like it does for autonomous communities. With that information, a set of objects is created after receiving the entity information through *DAOConsultor.php*. These variables store different information, relating to de entity general information, economic data for each of the 3 last years and, in case of municipalities, economic data of expense schedule. After receiving said data, it's stored in arrays in order to make it easier to access and show on screen in a table format.

For these entities, the basic structure of *name*, *table*, *graphics* is maintained from autonomous communities, in order to ensure internal consistency with the design.

4.6 PDF Generation

For the PDF reports generation, it was necessary to use an external open-source library called TCPDF. This library is found in the folder *includes/tcpdf* of the project and it is used in the scripts *pdfCCAA.php*, *pdfDIP.php* and *pdfMUN.php*.

The process begins when the user pushes the *Ver Informe* button and the corresponding script will be called (Fig.15).

```
<a href="pdfCCAA.php" target="_blank"><button type="button" id="verPDFCCAA">Ver Informe</button></a>
```

Fig.15: Tag that calls the PDF generation script in infoccaa.php

Then, in the script all the general and financial information from the facility and the country is collected from the database by calling the corresponding function of the *DAOConsultor* object (Fig.16).

```
/* RECOGEMOS LOS DATOS DE LA CCAA */
$daoccaa = new DAOConsultor();
$ccaa = $daoccaa->getCCAA($nombre);
$ccaaNac = $daoccaa->getCCAA('NACIONAL');
```

Fig.16: pdfCCAA file. Getter function for the facility and the country.

After that, the TCPDF library and all the general and financial data is copied to the created object. Finally, all the information is exported to a PDF format (Fig.17).

```
/* EXPORTACIÓN DEL ARCHIVO */
$pdf->lastPage();
ob_end_clean();

$pdf->Output('Informe_comunidad_autonoma.pdf', 'I');
```

Fig.17: pdfCCAA file. Information export

In this way, the PDF file is generated by the user in a separate window (Fig.18).

Informe de comunidad autónoma

Extremadura

Rating 2022: **B**
Tendencia: Estable¹

Rating 2021: **B**
Tendencia: Estable²

Información General

Presidente de la comunidad: Guillermo Fernández Vara
Vigencia: 2015-07-04
Partido político: PSOE
CIF: S06110011
Vía: PLAZA del Rastro S/N
Teléfono: 924003438
Código Postal: 6800
Fax: 924003441
Sitio web: www.juntaex.es
Correo electrónico: presidente@prs.juntaex.es

Datos económicos

Población (Año 2021): 1.059.501		PIB per cápita (Año 2017): 17.000	
Incremento del PIB de la comunidad	Incremento del PIB nacional	Paro	Paro nacional
<ul style="list-style-type: none">• 2020: -7.4%• 2019: 1.85%• 2018: 1.81%	<ul style="list-style-type: none">• 2020: -10.82%• 2019: 2.08%• 2018: 2.28%	<ul style="list-style-type: none">• 2021 (Trimestre 4): 18.94%• 2020 (Trimestre 4): 21.32%• 2019 (Trimestre 4): 23.48%	<ul style="list-style-type: none">• 2021 (Trimestre 4): 13.33%• 2020 (Trimestre 4): 16.13%• 2019 (Trimestre 4): 13.78%
Transacciones inmobiliarias	Transacciones inmobiliarias nacionales	Crecimiento de las empresas en la comunidad	Crecimiento de las empresas a nivel nacional
<ul style="list-style-type: none">• 2021 (Trimestre 4): 12.89%• 2020 (Trimestre 4): 12.89%	<ul style="list-style-type: none">• 2021 (Trimestre 4): 20.32%• 2020 (Trimestre 4): 19.96%	<ul style="list-style-type: none">• 2020: -0.51%• 2019: 1.2%• 2018: 1.89%	<ul style="list-style-type: none">• 2020: 1.22%• 2019: 0.76%• 2018: 1.68%

Fig.18: PDF file from Extremadura

4.7 Income and Expense Excel Generation

The export scripts are found in the *exports* folder (Fig.19).

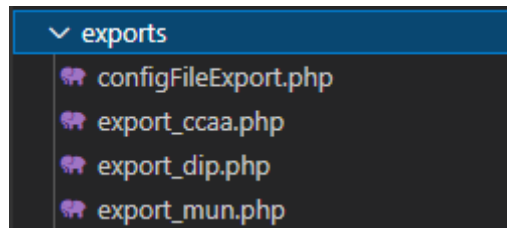


Fig.19: Export files location

- **configFileExport.php**: contains the configuration parameters for the export files.
- **Export_ccaa.php**: contains the export function of the incomes and expenses of the last three years of the autonomous communities.
- **Export_dip.php**: contains the export function of the incomes and expenses of the last three years of the councils.
- **Export_mun.php**: contains the export function of the incomes and expenses of the last three years of the municipalities.

Firstly, the export process starts when the user clicks on the *Exportar información* button found on the information page of the facility. Then, the script *procesarExportacion.php* is called, which would use the *export* function of the *DAOCargador* in order to create a Excel file in the project directory.

Nevertheless, as the file is unnecessary to be in local and it is required to be downloadable, the headers had to be changed to xlsx format after the file was created. By this way, the content of the local file will be exported to a dynamically generated downloadable Excel file instead of showing it in the page. Finally, the local Excel file is removed in order to free space (Fig.20).

```
<?php
require_once("includesWeb/daos/DAOCargador.php");

$nombre = urldecode(htmlspecialchars(trim(strip_tags($_REQUEST['nombre']))));
$type = htmlspecialchars(trim(strip_tags($_REQUEST['tipo'])));
$cargador = new DAOCargador();

$filename = $cargador->export($nombre,$type);
$nombre = str_replace(' ', '_', str_replace(',','',str_replace('-', '', strtolower($nombre)))).'_ingresos.xlsx';
header('Content-Type: text/xlsx; charset=utf-8');
header('Content-Disposition: attachment; filename="'.$nombre.'"');
if($filename!='') {
    flush();
    readfile($filename);
    unlink($filename);
}
exit();
?>
```

Fig.20: ProcesarExportacion script

Furthermore, *DAOCargador* behaves similarly to the import, that is, it runs a different script depending on the type of facility (Fig.21).

```

public function export($nombre, $type){
    $filepath = "";

    if($type=='ccaa'){
        $filepath = $this->export_ccaa($nombre);
    }
    else if($type=='dip'){
        $filepath = $this->export_dip($nombre);
    }
    else if($type=='mun'){
        $filepath = $this->export_mun($nombre);
    }

    return $filepath;
}

private function export_ccaa($ccaaNombre){
    return (new Exporter_ccaa())->export_ccaa($ccaaNombre);
}

private function export_dip($dipNombre){
    return (new Exporter_dip())->export_dip($dipNombre);
}

private function export_mun($munNombre){
    return (new Exporter_mun())->export_mun($munNombre);
}

```

Fig.21: Export script in DAOCargador

Within the export script, the PHPSpreadSheet library is used to create a new SpreadSheet (Fig.22) and store all the income and expense information in it. For this, the general and financial information of the facility is queried through the *DAOConsultor*.

```

$file = new Spreadsheet();
$active_sheet = $file->getActiveSheet();

```

Fig.22: SpreadSheet creation

```

$daoccaa = new DAOConsultor();
$ccaa = $daoccaa->getCCAA($ccaaNombre);

```

Fig.23: Getting information from autonomous community in export_ccaa.php

At the end, all the stored information is saved in an Excel file that is located inside the local project directory (Fig.24).

```

$writer = new Xlsx($file);
$filename = __DIR__.'/'.str_replace(' ', '_', str_replace(',', '', str_replace('-', '', strtolower($ccaa->getNombre())))).'_finanzas.xlsx';
$writer->save($filename);

```

Fig.24: Storing the SpreadSheet in Excel format

4.8 Queries

All the scripts related to the queries are found in the *includesWeb/daos* folder.

To begin with, the functionality is implemented on the queries page. Depending on the button the user pushes, a different script will be called:

- procesarConsultaCCAA.php: show all the results from the autonomous communities.
- procesarConsultaDIP.php: show all the results from the councils.
- procesarConsultaMUN.php: show all the results from the municipalities.

In all of them, the data from the checkboxes (Fig.25) and the filters (Fig.26) are stored in variables and they are sent to the corresponding function of the DAOConsultor (Fig.27).

```
// Hay un array de checkboxes. Si el usuario activa una checkbox, se guarda como true, sino como false.
// Asi se sabe que checkboxes ha seleccionado el usuario y cuales no
if(!empty($_REQUEST['endeudamientoCCAA_C'])) array_push($checked_boxes, true);
else array_push($checked_boxes, false);
if(!empty($_REQUEST['ahorronetoCCAA_C'])) array_push($checked_boxes, true);
else array_push($checked_boxes, false);
if(!empty($_REQUEST['pmpCCAA_C'])) array_push($checked_boxes, true);
else array_push($checked_boxes, false);
if(!empty($_REQUEST['dcpccAA_C'])) array_push($checked_boxes, true);
else array_push($checked_boxes, false);
if(!empty($_REQUEST['ingrnofinCCAA_C'])) array_push($checked_boxes, true);
else array_push($checked_boxes, false);
```

Fig.25: Checkboxes handling in procesarConsultaCCAA.php

```
// Revisa los filtros. Si el usuario ha seleccionado un filtro, se guarda su valor, sino se deja a NULL
if(!empty($_REQUEST['scoringCCAA']) && $_REQUEST['scoringCCAA']!= 'inicio')
    $scoring = htmlspecialchars(trim(strip_tags($_REQUEST['scoringCCAA'])));

if(!empty($_REQUEST['poblacionCCAA']) && $_REQUEST['poblacionCCAA']!= 'inicio')
    $poblacion = htmlspecialchars(trim(strip_tags($_REQUEST['poblacionCCAA'])));

if(!empty($_REQUEST['gastoCCAA']) && $_REQUEST['gastoCCAA']!= 'inicio')
    $gasto = htmlspecialchars(trim(strip_tags($_REQUEST['gastoCCAA'])));

if(isset($_REQUEST['selection'])){
    $choice = htmlspecialchars(trim(strip_tags($_REQUEST['selection'])));
    if($choice == 'SelectYear'){
        if(!empty($_REQUEST['anhoCCAA']) && $_REQUEST['anhoCCAA']!= 'inicio')
            $anho = htmlspecialchars(trim(strip_tags($_REQUEST['anhoCCAA'])));
    }
    else {
        if(!empty($_REQUEST['from']) && !empty($_REQUEST['to'])){
            if($_REQUEST['from']!= 'inicio')
                $from = htmlspecialchars(trim(strip_tags($_REQUEST['from'])));

            if($_REQUEST['to']!= 'inicio')
                $to = htmlspecialchars(trim(strip_tags($_REQUEST['to'])));
        }
    }
}
```

Fig.26: Filters handling in procesarConsultaCCAA.php

```
$ccaas = (new DAOConsultor())->consultarCCAAs($scoring, $poblacion, $endeudamiento, $ahorro_netto, $pmp, $choice, $anho, $from, $to, $dcpcc, $ingrnofin, $gasto, $checked_boxes);
```

Fig.27: Query function calling from procesarConsultaCCAA.php

In DAOConsultor, the corresponding DAO is called depending if the query was done from the autonomous communities, councils or municipalities (Fig.28).

- DAOConsultorCCAA.php
- DAOConsultorDIP.php
- DAOConsultorMUN.php

```
public function consultarCCAAs($scoring, $poblacion, $endeudamiento, $ahorro_neto, $mpm, $choice, $anho, $from, $to, $dcp, $ingrnofin, $gasto, $checked_boxes){
    $ccaas = array();
    $daoacca = new DAOConsultorCCAA();

    return $daoacca->consultarCCAAs(strtoupper($scoring), $poblacion, $endeudamiento, $ahorro_neto, $mpm, $choice, $anho, $from, $to, $dcp, $ingrnofin, $gasto, $checked_boxes);
}

public function consultarDIPs($scoring, $poblacion, $endeudamiento, $ahorro_neto, $fondliq, $choice, $anho, $from, $to, $autonomia, $mpm, $ingrnofin, $gasto, $checked_boxes){
    $dips = array();
    $daodip = new DAOConsultorDiputacion();

    return $daodip->consultarDIPs(strtoupper($scoring), $poblacion, $endeudamiento, $ahorro_neto, $fondliq, $choice, $anho, $from, $to, $autonomia, $mpm, $ingrnofin, $gasto, $checked_boxes);
}

public function consultarMUNs($scoring, $poblacion, $endeudamiento, $ahorro_neto, $fondliq, $choice, $anho, $from, $to, $autonomia, $provincia, $mpm, $ingrnofin, $gasto, $prog, $checked_boxes){
    $mun = array();
    $daomun = new DAOConsultorMunicipio();

    return $daomun->consultarMUNs(strtoupper($scoring), $poblacion, $endeudamiento, $ahorro_neto, $fondliq, $choice, $anho, $from, $to, $autonomia, $provincia, $mpm, $ingrnofin, $gasto, $prog, $checked_boxes);
}
}
```

Fig.28: Queries functions from DAOConsultor.php

Each function will have the same structure. Firstly, it sets the fields that were requested by checking the checkboxes that had been activated by the user (Fig.29), and later, it is added the conditions that were requested in the filters (Fig.30). This is done through the modifications of the variables *conditions*, *returning_values*, *group_by*, *order_by*, *joins*.

```
//Checkboxes
if($checked_boxes[0]){ //endeudamiento
    if($joins!=""){
        if($conditions!=""){
            $conditions = $conditions." AND ";
        }
        if(strpos($joins, 'scoring_ccaa ')!==false) $conditions = $conditions . "cuentas_ccaa_general_mensual.ANHO = scoring_ccaa.ANHO ";
        else if(strpos($joins, 'cuentas_ccaa_general ')!==false) $conditions = $conditions . "cuentas_ccaa_general_mensual.ANHO = cuentas_ccaa_general.ANHO ";
        else if(strpos($joins, 'cuentas_ccaa_gastos ')!==false) $conditions = $conditions . "cuentas_ccaa_general_mensual.ANHO = cuentas_ccaa_gastos.ANHO ";
        else if(strpos($joins, 'cuentas_ccaa_ingresos ')!==false) $conditions = $conditions . "cuentas_ccaa_general_mensual.ANHO = cuentas_ccaa_ingresos.ANHO ";
        else if(strpos($joins, 'deudas_ccaa ')!==false) $conditions = $conditions . "cuentas_ccaa_general_mensual.ANHO = deudas_ccaa.ANHO ";
    }
    if(strpos($returning_values, 'ANHO')===false) $returning_values = $returning_values . ",cuentas_ccaa_general_mensual.ANHO";
    if(strpos($returning_values, 'c2.MES')===false) $returning_values = $returning_values . ", c2.MES";
    $returning_values = $returning_values."c2.DEUDA.VIVA INGR_COR";
    if(strpos($joins, 'INNER JOIN cuentas_ccaa_general_mensual ON cuentas_ccaa_general_mensual.CODIGO=ccaas.CODIGO')===false) $joins = $joins . "INNER JOIN cuentas_ccaa_general_mensual ON cuentas_ccaa_general_mensual.CODIGO=ccaas.CODIGO";
    if(strpos($joins, 'INNER JOIN (SELECT c2.CODIGO, c2.ANHO, c2.MES, c2.DEUDA.VIVA INGR_COR FROM cuentas_ccaa_general_mensual c2 INNER JOIN (SELECT c2_aux.CODIGO, c2_aux.ANHO, MAX(c2_aux.MES) MAX_MES2 FROM c2_aux WHERE c2_aux.CODIGO=c2.CODIGO AND c2_aux.ANHO=c2.ANHO) c2_aux ON c2.CODIGO=c2_aux.CODIGO AND c2.ANHO=c2_aux.ANHO) c2 ON c2.CODIGO=c2_aux.CODIGO AND c2.ANHO=c2_aux.ANHO')===false) $joins = $joins . "INNER JOIN (SELECT c2.CODIGO, c2.ANHO, c2.MES, c2.DEUDA.VIVA INGR_COR FROM cuentas_ccaa_general_mensual c2 INNER JOIN (SELECT c2_aux.CODIGO, c2_aux.ANHO, MAX(c2_aux.MES) MAX_MES2 FROM c2_aux WHERE c2_aux.CODIGO=c2.CODIGO AND c2_aux.ANHO=c2.ANHO) c2_aux ON c2.CODIGO=c2_aux.CODIGO AND c2.ANHO=c2_aux.ANHO) c2 ON c2.CODIGO=c2_aux.CODIGO AND c2.ANHO=c2_aux.ANHO";
    if(strpos($order_by, 'ANHO DESC')===false) $order_by = $order_by . "cuentas_ccaa_general_mensual.ANHO DESC, ";
    if(strpos($group_by, 'ANHO')===false) $group_by = $group_by . ",cuentas_ccaa_general_mensual.ANHO ";
}
}

if($checked_boxes[1]){ //ahorro neto
    if($joins!=""){
        if($conditions!=""){
            $conditions = $conditions." AND ";
        }
        if(strpos($joins, 'scoring_ccaa ')!==false) $conditions = $conditions . "cuentas_ccaa_general.ANHO = scoring_ccaa.ANHO ";
        else if(strpos($joins, 'cuentas_ccaa_general_mensual ')!==false) $conditions = $conditions . "cuentas_ccaa_general.ANHO = cuentas_ccaa_general_mensual.ANHO ";
        else if(strpos($joins, 'cuentas_ccaa_gastos ')!==false) $conditions = $conditions . "cuentas_ccaa_general.ANHO = cuentas_ccaa_gastos.ANHO ";
        else if(strpos($joins, 'cuentas_ccaa_ingresos ')!==false) $conditions = $conditions . "cuentas_ccaa_general.ANHO = cuentas_ccaa_ingresos.ANHO ";
        else if(strpos($joins, 'deudas_ccaa ')!==false) $conditions = $conditions . "cuentas_ccaa_general.ANHO = deudas_ccaa.ANHO ";
    }
    if(strpos($returning_values, 'ANHO')===false) $returning_values = $returning_values . ",cuentas_ccaa_general.ANHO";
    $returning_values = $returning_values.",cuentas_ccaa_general.R_SOSTE_FINANCIERA";
    if(strpos($joins, 'INNER JOIN cuentas_ccaa_general ON cuentas_ccaa_general.CODIGO=ccaas.CODIGO')===false) $joins = $joins . "INNER JOIN cuentas_ccaa_general ON cuentas_ccaa_general.CODIGO=ccaas.CODIGO ";
    if(strpos($order_by, 'ANHO DESC')===false) $order_by = $order_by . "cuentas_ccaa_general.ANHO DESC, ";
    if(strpos($group_by, 'ANHO')===false) $group_by = $group_by . ",cuentas_ccaa_general.ANHO ";
}
}
```

Fig.29: Checkboxes handling in *consultarCCAAs* function, in DAOConsultorCCAA.php

```

// Filtros
if(!empty($scoring)){
    if($conditions!=""){
        $conditions = $conditions." AND ";
    }
    if($joins!=""){
        $tmp=$conditions;
        if(strpos($joins, ' cuentas_ccaa_general_mensual ')!==false) $conditions = $conditions . "scoring_ccaa.ANHO = cuentas_ccaa_general_mensual.ANHO";
        else if(strpos($joins, ' cuentas_ccaa_general ')!==false) $conditions = $conditions . "scoring_ccaa.ANHO = cuentas_ccaa_general.ANHO";
        else if(strpos($joins, ' cuentas_ccaa_gastos ')!==false) $conditions = $conditions . "scoring_ccaa.ANHO = cuentas_ccaa_gastos.ANHO";
        else if(strpos($joins, ' cuentas_ccaa_ingresos ')!==false) $conditions = $conditions . "scoring_ccaa.ANHO = cuentas_ccaa_ingresos.ANHO";
        else if(strpos($joins, ' deudas_ccaa ')!==false) $conditions = $conditions . "scoring_ccaa.ANHO = deudas_ccaa.ANHO";

        if($tmp!=$conditions) $conditions = $conditions . " AND ";
    }

    $conditions = $conditions."scoring_ccaa.RATING = '$scoring' ";
    if(strpos($returning_values, 'ANHO')===false) $returning_values = $returning_values . ",scoring_ccaa.ANHO";

    $returning_values = $returning_values.",scoring_ccaa.ANHO, scoring_ccaa.RATING";

    if(strpos($joins, 'INNER JOIN scoring_ccaa ON scoring_ccaa.CODIGO=ccaas.CODIGO')===false) $joins = $joins . "INNER JOIN scoring_ccaa ON scoring_ccaa.CODIGO=ccaas.CODIGO ";
    if(strpos($order_by, 'ANHO DESC')===false) $order_by = $order_by . "scoring_ccaa.ANHO DESC, ";
    if(strpos($group_by, 'ANHO')===false) $group_by = $group_by . ",scoring_ccaa.ANHO ";
}

```

Fig.30: Filters handling in *consultarCCAAs* function, in *DAOConsultarCCAA.php*

At the end of the function, these elements will be concatenated with the query and sent to the database (Fig.31). The results will be stored in an array and returned to the *DAOConsultor* function, and the *DAOConsultor* function will return it to the *procesarConsultaCCAA.php*, *procesarConsultaDIP.php* or *procesarConsultaMUN.php* file.

```

$sql = "SELECT ccaas.CODIGO, ccaas.NOMBRE $returning_values FROM ccaas $joins $conditions GROUP BY ccaas.CODIGO $group_by ORDER BY $order_by ccaas.CODIGO ASC";

```

Fig.31: SQL query in *DAOConsultarCCAA.php*

Finally, the results are shown in a table format divided by years in descending order, that is, from the most recent year to the oldest.

4.9 API

After the basic requirements for the project were fully developed, some time was dedicated to start a basic but totally functional API. Said functionality would allow any user with programming skills to send HTTP requests to our website in order to receive data in a standardised JSON format.

In order to maintain the integrity of our Database's data, only HTTP GET requests are available, since any other kind of request would translate into a modification of the information. This functionality is open for future development but has been declined for the project's due date.

This API was developed fully in PHP and its set of requests is based on an adaptation of previously developed queries to obtain data for *Reports* or *Query* pages.

Requests can be made via a created *api.php* page for the public website as follows:

analisisfinancierosectorpublico.com/api.php?request=<request>

If the request needs any additional information in order to be processed, it can be added at the end of the previous request as follows

&name=<name>

Said set of requests is as follows:

<i>Request</i>	<i>Description</i>	<i>Format</i>
getAllFacilities	Returns a list of the names of all existing Autonomous Communities, Concils and Municipalities	<i>request=getAllFacilities</i>
getAllProvincias	Returns a list of all Provinces	<i>request=getAllProvincias</i>
getProvinciaById (id)	Returns the Province for the given ID	<i>request=getProvinciaById&id=id</i>
getAllCCAAs	Returns a list of the names of all Autonomous Communities	<i>request=getAllProvincias</i>
getCCAAById (id)	Returns the Autonomous Community for the given ID	<i>request=getCCAAById&id=id</i>
getCCAA (nombre)	Returns all general and economic data for a given Autonomous Community by name	<i>request=getCCAA&name=name</i>
getMunicipio (nombre)	Returns the general info for a given Municipality by name	<i>request=getMunicipio&name=name</i>
getEconomiaMUN (código, año)	Returns the economic data for a given Municipality by ID and year	<i>request=getEconomiaMUN&id=id &year=year</i>
getProgMUN (código)	Returns the expenses program for a given Municipality by ID	<i>request=getCCAA&id=id</i>
getDiputacion (nombre)	Returns the general info for a given Council by name	<i>request=getDiputacion&name=name</i>
getEconomiaDIP (código, año)	Returns the economic data for a given Council by ID and year	<i>request=getEconomiaDIP&id=id &year=year</i>

The code for this functionality is quite straightforward. First of all it detects the method of the request, filtering only by GET method and dismissing any other kind. After the filter, a single “switch” organised by a case for each possible request is implemented, also detecting and notifying when a given request is nonexistent. In case of a match with one of the available requests, the data is translated into a JSON format and given back to the user.

```

1 <?php
2 session_start();
3 header('Content-Type: application/json');
4 require('includesWeb/daos/DAOConsultor.php');
5
6 //La variable $_SERVER tiene entre otros, la petición HTTP que se está realizando a la web
7 $metodo = $_SERVER['REQUEST_METHOD']; //Solo nos interesa "GET"
8
9 /* EJEMPLO VARIABLE */
10 // $id = $_REQUEST['id'] ?? null; //Si no se nos proporciona el ID en la petición, por defecto lo dejamos en null
11
12 $peticion = $_REQUEST['request'] ?? null;
13 $id = $_REQUEST['id'] ?? null;
14 $nombre = $_REQUEST['name'] ?? null;
15 $anio = $_REQUEST['year'] ?? null;
16
17 if($metodo == "GET"){
18     switch($peticion){
19         case "getAllFacilities":
20             $dao = new DAOConsultor();
21             $peticion = new stdClass();
22
23             $result = $dao->getAllFacilities();
24
25             $peticion->getAllFacilities = $result;
26             echo json_encode($peticion);
27             break;
28
29         case "getAllProvincias":
30             $dao = new DAOConsultor();
31             $peticion = new stdClass();
32
33             $result = array();
34             $provincias = $dao->getAllProvincias();
35             foreach($provincias as $provincia){
36                 $element = new stdClass();
37                 $element->Codigo = $provincia->getCodigo();
38                 $element->Nombre = $provincia->getNombre();
39                 $element->Autonomia = $provincia->getCCAACode();
40                 array_push($result, $element);
41             }
42
43             $peticion->getAllProvincias = $result;
44             echo json_encode($peticion);
45             break;
46

```

Fig.32: Request filter in api.php

In order to test the HTTP Requests on our API, we used the program *Postman*. With its clear and simple interface we could generate plenty of requests and make sure the results were just as we expected.

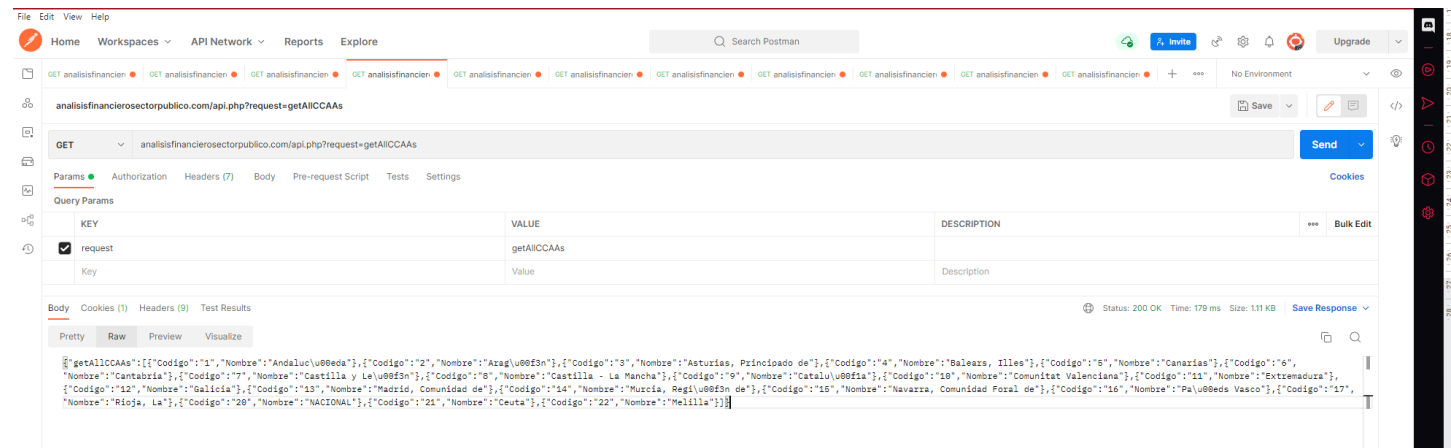


Fig.33: Example of HTTP Request using Postman

4.10 Usage of Hosting

One of the requirements for the project was to make it accessible to anyone online, on an open and free-to-use website. For that reason, the co-director contracted a hosting service on IONOS with a personalised domain name.

IONOS offers a MySQL database managed with *Phpmyadmin* as a simple access interface. In addition, it also has an uploading page for the web code, and a website control panel with a set of extra available configurations, such as the web HTTPS certificate.

For the range of the project, the service that was decided to pay is a portion of a shared server, which means that the web application shares the server with other domains and websites. This helps reduce the cost of the project, but also brings a few inconvenients, such as the impossibility to configure the server to a free will, in order to adapt it perfectly to the project needs. Some of these limitations are, for example, the impossibility to upgrade the server memory limit, waiting time for a petition and maximum used memory to upload a file. Since these problems can't be solved without a full control of the server, the project had to be adapted and won't allow, for example, an insertion of a big file of data, such as the file for Municipalities, which is the biggest file, and the one that requires a server modification.

These previously explained problems could all be solved in the future if the project changes its location to a dedicated server, allowing a full configuration of it. The issue is that it requires an increment of the general cost of the project.

4.11 Individual work

Aldair

To begin with, Aldair was responsible for developing the reports interface of the autonomous communities, where all the financial and general information is shown with the corresponding graphics by using ChartJs, an open-source library. Regarding its design, most of the decisions were taken following the preferences of Esteban, who normally made suggestions about the structure of the interface, such as the location of the charts, its content, the information to be shown, etc.

In addition, he implemented the export functionality whose scripts are in the *exports* folder, in order to export all the income information into an Excel file. In order to do this, it was decided to use the PHPSpreadSheet open-source library, as well as ChartJs. It was decided to maintain the export functionality in another folder different to the imports folder in order to have the functionalities as separated as possible.

Furthermore, he creates the DAOs scripts:

- *DAOCargador*
- *DAOConsultor*
- *DAOConsultorCCAA*
- *DAOConsultorDIP*
- *DAOConsultorMUN*

This was done in order to manage the different accesses to the database.

It was decided to create a general script called *DAOConsultor*, in order to centralise all the processes. By this way, every time an access to the database is needed, then *DAOConsultor* is called, and it will call the corresponding function from a specific DAO depending on the type of public facility. Regarding *DAOCargador*, it has the same task as the *DAOConsultor*, but in this case it will manage both the import and export functionalities. Even though these scripts were created by Aldair, they could be also modified by Miquel, as well as the *cargador*, *municipio*, *ccaa*, *diputacion*, *config*, *provincia* and *usuario* objects, located in the same *includesWeb* folder. It should be noted that the *cargador* object is implemented following the Singleton pattern, since it is considered to have better only one instance of the loader, instead of creating multiple instances of the same object.

Additionally, some Javascript functions were included by Aldair. For instance, the possibility to change the type of chart in some graphics of the autonomous community interface. However, Miquel was who had more responsibility in this work, as well as the styles, as it is explained later.

Apart from that, he developed the login interface, in addition to a specific profile interface that will appear depending if the logged account is an administrator or a manager.

- If the user is an administrator, the user could manage, create and delete users, apart from importing Excel files and modify its personal user data.
- If the user is a manager, then it can only import Excel files and modify its personal user data.

This login interface was originally created since the import functionality should be located somewhere restricted to authorised users, otherwise it could be free to use, which can provoke that any user could import undesired files and insert garbage in the database. However, as the access to this section is limited, the possibility for conventional users to register must be forbidden. Therefore, if a new user wants to be added to the database, the administrator must create it in an exclusive section of the control panel. The administrator must set the user name, email, role and password of the new account. It is important that once the account is created, its new owner should change its password for security reasons.

At last, he implemented the queries and results interface. It was decided to divide the filters between autonomous communities, councils and municipalities since Esteban preferred to have the results of the facilities separated. Throughout the development of the web application, the number of filters and checkboxes were increasing and decreasing depending on Esteban preferences.

Miquel

On the other hand, Miquel has been in charge of the other side of the development. As previously said, most part of the previous decisions were taken during voice or video call with Aldair.

These decisions are as follows:

- Technologies and programming languages: including the decision to use PHP for the development of the base of the project, a SQL database and JavaScript for other functionalities.
- Raw data analysis
- Database design
- Reports design style

Apart from this previous to development decisions, some of the functionalities were fully developed by both of the project developers, Miquel and Aldair:

- **Import scripts:** since the raw data analysis was done jointly, it was decided that the import script had to be developed jointly too. This functionality is one of the most important one, since it allows the page to automatically import data to the database. Because it's such a central part, in order to make sure it was as most efficient as possible.
- **Database structure:** in addition to the import script, the structure previously designed for the database had to be implemented using SQL on *phpmyadmin*.

Obviously since the project is developed by both of the developers, if some help was needed from any of the two parts on part of the code, the other one tried to make the task easier.

Since Aldair was in charge of the development of the reports for Autonomous Communities, Miquel implemented this functionality for both Councils and Municipalities. For these ones, as explained in the point *4.5 Reports*, the structure for both is similar in its queries, tables and graphics. In order to maintain internal consistency with the reports of autonomous communities, the code for scoring, tables and graphics follows the same structure and characteristics.

After the report page was completed, the new functionality to be developed was an automatically generated export file of the said PDF format. This functionality required a previous investigation and selection of tools, because Miquel never had created a PDF file using PHP, JavaScript or HTML.

In order to create these PDF adaptations for the report page, it was decided to choose one of the existing PHP libraries that would facilitate the work. Some of the libraries took into account were the following:

- **FPDF**
- **mPDF**
- **DOMPDF**
- **TCPDF:** library selected for the project, because of its spread documentation and open source licence

In addition, Miquel was responsible for the graphic design of the website. This includes the initial proposition of design made in Figma and the HTML and CSS implementation of the final designs. The final designs are described in the next point *6. Instructions for use* and include the design of the search bar, navigation, footer and login.

These designs were a result of a set of iterations of meetings with the co-director, who proposed a few changes on the location of some elements.

Moreover, he also developed the functionality of the search bar in full JavaScript. This option searches the written text on the bar and shows a list of matching entities. It was decided that this would appear on the top part of the website.

All the page styles were also coded by Miquel in CSS, applying the colours decided in the design iterations on Figma. These styles were detailed in order to try to maintain the web in order in several screen sizes.

Finally, he also developed the API for the website. As previously said in *4.9 API*, the handled HTTP Get Requests are developed as an adaptation of the queries previously created, since this functionality only serves as a base for the future.

As previously mentioned by Aldair, him and Miquel participated in the weekly meetings with the project directors in order to maintain the project on schedule and explain the new functionalities developed.

5. Evaluation

5.1 Evaluation with final users

During the development of the website, feedback was received both from Esteban and from colleagues in his company NosterEconomia. However, their opinions were from a business point of view and also as conventional users, who will initially be the main users of the website, apart from journalists. In this way, the following changes were proposed in chronological order:

1. Adding colours to the percentages of the ratios, in order to help users understand the seriousness of the data.
2. Change the location of the search engine from being on the home page to being positioned at the top of the page and in a fixed way, i.e., to keep the search engine always on all the pages of the website.
3. Change the location of the filters and checkboxes on the query page, inverting their positions. In this way, the filters would be displayed first instead of the checkboxes, which would be moved to the bottom.
4. Show on the query results page the filters that the user had previously selected, so that the user would know at all times which filters had been selected, as well as being able to show the total number of results that had been found in the database that met those conditions.
5. Export the information from the facility to an Excel file, so that the information can be stored locally.
6. Adding a set of conclusions based on the rating of the facility in the two most recent years. In addition, another set of conclusions was also proposed under the most recent scoring based on the percentages of the most recent ratios of the facility.

All these proposed changes and improvements were successfully implemented on the website. However, the fifth request was left incomplete, as it was requested to export not only all information on the income and expenses, but also information on the ratios and scoring. It is expected that this option will be extended in the future.

6. Instructions for use

6.1 Access to the site

In order to access the website, the user needs to put the URL <https:// analisisfinancierosectorpublico.com/index> in any web browser (Google Chrome, Mozilla Firefox, Opera, Microsoft Edge, etc.).

6.1.1 Main page

The application has been designed with usability in mind, following a simple and intuitive design. For this, a default structure formed by two differentiated headers, content and a footer is used (Fig.32).

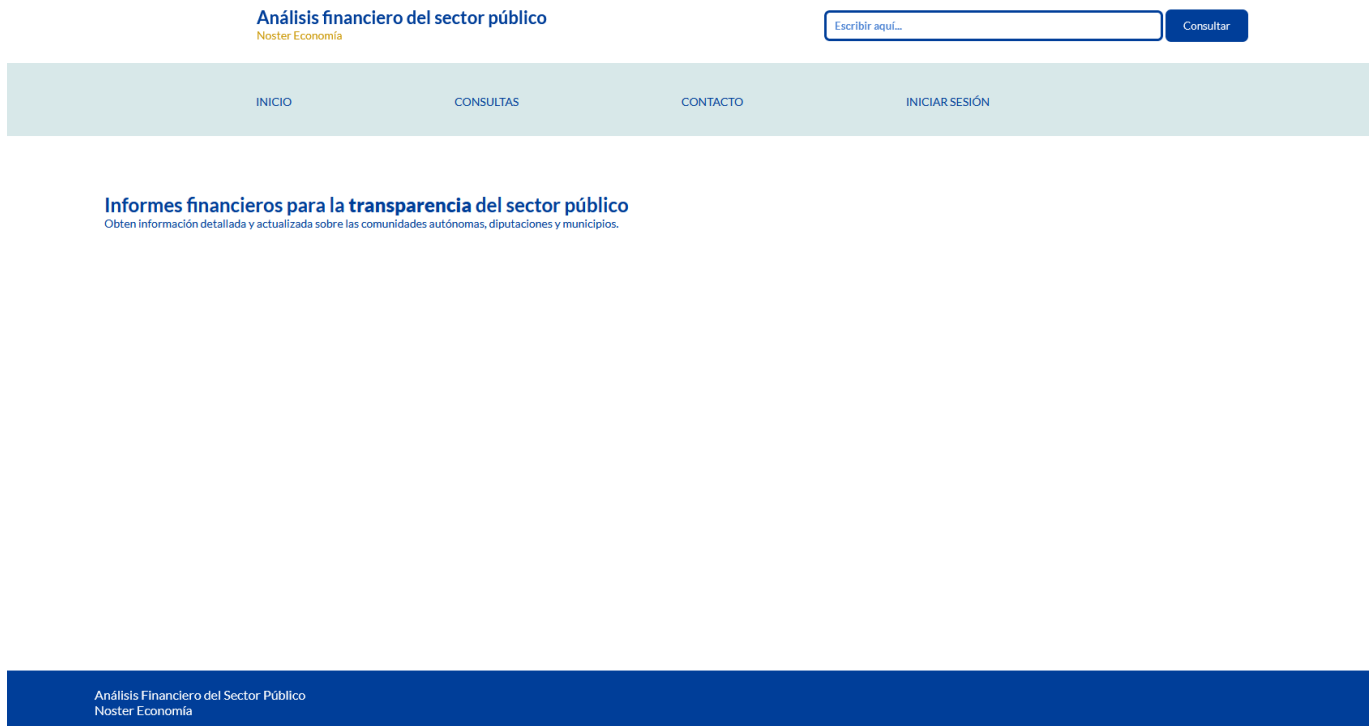


Fig.34: Main interface

- The first header contains an internal browser in order to search any facility, which can be an autonomous community, council or municipality.
- The second header includes all the different pages the website has, which are the main page, the query, contact and profile page.

The browser includes an autocomplete mechanism such that it shows all the matching names. At the same time, it shows the facility type (*CCAA*, *DIPUTACIÓN* or *MUNICIPIO*) (Fig.33). This mechanism activates when the user types three letters or more. In this way, the user can find the wanted entity more easily, besides knowing all the options available on the website, therefore contributing to the usability of the website.

The image shows a search interface with a text input field containing 'mad' and a 'Consultar' button. Below the input field, a list of suggestions is displayed:

- Extremadura (CCAA)
- Madrid, Comunidad de (CCAA)
- Madrigueras (MUNICIPIO)
- Paterna del Madera (MUNICIPIO)
- Madrigal de las Altas Torres (MUNICIPIO)

Fig.35: Autocomplete mechanism

6.1.2 Reports page

Once the user selects the facility, it is redirected to the information page. Depending on the type of facility, the information page will be redirected to different pages, but all of them will show similar information.

Firstly, the webpage will show its ratings of the last two years, in addition to a set of conclusions about the financial situation of the facility based on the most recent thirteen ratios. General data is shown, such as the name of its president, current political party, CIF, address, phone number, post code, fax, website link and email. After that, all the economical information is shown, such as its benefits, expenses and ratio percentages of the last three years. Apart from that, it is possible to download this financial information as a PDF document, in addition to exporting the benefits and expenses into an Excel document.

All the ratios will come with graphics created by the open-source library Chart.js. Most of the graphics will have a fixed bar format, but in some of them it is possible to change the format to line chart. For instance, the PMP charts for the information pages of autonomous communities (Fig.34).

It should be noted that ratings and ratios are colored depending on whether they are beneficial or detrimental data for the public facility. This, in terms of usability, is a disadvantage for people who suffer from colour blindness. Therefore, a set of conclusions was added at the end of the most recent rating (Fig. 35). These are based on the most recent data for each ratio, which are always found in the rightmost part of each ratio (Fig.36).

Análisis financiero del sector público
Noster Economía

Escribir aquí... [Consultar](#)

INICIO CONSULTAS CONTACTO INICIAR SESIÓN

Comunidad Autónoma
Nota: el rating se corresponde con los últimos datos disponibles.

Madrid, Comunidad de
Rating 2022

B

Tendencia: Estable
La entidad muestra una situación financiera holgada, con un nivel relativamente bajo de riesgo, que se ha comportado de forma estable durante los últimos años.

Buen resultado presupuestario. Alto nivel de deuda financiera. Sin capacidad de ahorro pero en niveles aceptables. Muy bajo apalancamiento operativo. Bajo nivel de carga financiera. Pago de facturas lento. Muy eficiente en términos de gastos ordinarios. Muy buena previsión de ingresos. Baja ejecución presupuestaria de gastos. Bajo nivel de deuda comercial. Elevado nivel de pagos sobre gastos reconocidos. Muy buen nivel de eficacia recaudatoria.

Rating 2021

B

Tendencia: Estable
La entidad muestra una situación financiera holgada, con un nivel relativamente bajo de riesgo, que se ha comportado de forma estable durante los últimos años.

[Ver Informe](#) [Exportar información](#)

Datos generales
 Presidente de la comunidad: Isabel Díaz Ayuso
 Vigencia: 2019-08-19
 Partido político: PP
 CIF: 1760001E
 Vía: PLAZA Puerta del Sol, 7
 Teléfono: 915015192
 Código Postal: 28013
 Fax: 915020247
 Sitio web: www.madrid.org
 Correo electrónico: atencionalciudadano@madrid.org

Datos económicos

Población (Año 2021): 6.751.251 habs.	PIB per cápita (Año 2017): 34.000€
---------------------------------------	------------------------------------

Análisis Financiero del Sector Público
Noster Economía

Fig.36: Information interface example about Comunidad de Madrid

Madrid, Comunidad de
Rating 2022



Tendencia: Estable

La entidad muestra una situación financiera holgada, con un nivel relativamente bajo de riesgo, que se ha comportado de forma estable durante los últimos años.

Buen resultado presupuestario. Alto nivel de deuda financiera. Sin capacidad de ahorro pero en niveles aceptables. Muy bajo apalancamiento operativo. Bajo nivel de carga financiera. Pago de facturas lento. Muy eficiente en términos de gastos ordinarios. Muy buena previsión de ingresos. Baja ejecución presupuestaria de gastos. Bajo nivel de deuda comercial. Elevado nivel de pagos sobre gastos reconocidos. Muy buen nivel de eficacia recaudatoria.

Fig.37: Conclusions example from Comunidad de Madrid



Fig.38: Ratio information example from Comunidad de Madrid

6.1.3 Queries page

On the other hand, the *Consultas* section is used to show a list of facilities that accomplish a set of filters and checkboxes that the user selects. The filters are used to show a list of facilities based on some conditions established by the user, whereas the checkboxes are used to let the user select which fields must be shown in the results (Fig.39).

The filters options can be:

- Scoring
- Población
- Comunidad Autónoma
- Provincia

The checkboxes options can be:

- Endeudamiento
- Ahorro neto
- Fondos líquidos
- PMP (Período Medio de Pago)
- Deuda comercial pendiente de pago
- Ingresos no financieros
- Tipo de gasto
- Programa de gasto

Comunidades Autónomas | Diputaciones | Municipios

Consultas de comunidades autónomas

Filtros
Para aquellas entidades seleccionadas según los siguientes filtros...

Consultar por año Consultar por intervalo de años

Scoring
Seleccione un rating

Población
Seleccione un tramo de población

Mostrar
Mostrar los siguientes datos

Seleccionar todo
 Endeudamiento Ahorro neto PMP Deuda comercial pendiente de pago Ingresos no financieros

Tipo de gasto
Seleccione un tramo

Consultar

Fig.39 : Filters interface for autonomous communities

This page is divided into three subsections, depending on the type of facilities the user wants to filter: autonomous communities, councils and municipalities. Each one will have its own filters and checkboxes. Each section has a comment informing the user about its usefulness, and the orientation is always vertical, combining form radio buttons, checkboxes and selection lists, which contribute to the usability of the website.

Once the filters and conditions are set, the button is pressed and the results will be shown.

At the beginning, the selected filters by the user are shown so that the user can remember the filters it has chosen. Subsequently, the application shows the number of data that have been found in the entire database and displays them in tabular form. They are divided by years in descending order, that is, from the most recent year to the oldest, and in each year the number of results obtained in each year is shown (Fig. 40). In case the user has chosen to view a schedule of expenditure of municipalities, at the end of each year, the total sum of expenditure of all municipalities is shown.

Filtros seleccionados

Scoring: B

Se han encontrado 60 resultados

2021

11 resultados

	Nombre	Scoring	Endeudamiento	Ahorro neto	PMP	Deuda comercial pendiente de pago	Nivel de ingresos no financieros
1	Andalucía	B	127.26%	-10.7%	14.81 días	10.44%	31.962.443.040€
2	Balears, Illes	B	157.63%	-6.45%	49.48 días	13.04%	5.938.349.420€
3	Canarias	B	64.91%	-21.13%	13.44 días	7.27%	10.953.229.590€
4	Cantabria	B	130.95%	-8.74%	13.76 días	9.58%	2.861.795.990€
5	Castilla y León	B	147.95%	-8.45%	19.15 días	4.98%	9.367.381.440€
6	Extremadura	B	113.67%	-12.98%	7.42 días	5.94%	5.144.481.240€
7	Galicia	B	112.63%	-7%	16.59 días	7.72%	10.822.998.220€
8	Madrid, Comunidad de	B	146.31%	-4.09%	79.76 días	2.84%	24.402.637.530€
9	Navarra, Comunidad Foral de	B	116.03%	-74.25%	23.95 días	5.89%	4.774.760.110€
10	País Vasco	B	91.3%	4.27%	15.63 días	10.05%	12.578.091.180€

Análisis Financiero del Sector Público
Noster Economía

Fig.40: Results interface for autonomous communities

6.1.4 Contact page

In the Contact page, the email of the enterprise, NosterEconomía, is provided in order to let the users to send emails (Fig.41). Due to the lack of time, this page could not be developed as far as it was planned initially, however there are clear ideas to improve it, as it is explained in 7.2.

Análisis financiero del sector público
Noster Economía[INICIO](#)[CONSULTAS](#)[CONTACTO](#)[INICIAR SESIÓN](#)**Contacto**

noster.economia@gmail.com

Análisis Financiero del Sector Público
Noster Economía

Fig.41: Contact interface

6.1.5 Profile page

Finally, in the Profile section there is a form in order to put the email and the password (Fig.42). The possibility for users to sign up and create an account in the website is disabled, since it could give the users the possibility to enter to the control panel and insert any type of undesirable data to the database by using the import option. Therefore, the only option to enter to the profile page is to obtain an account given by the administrator.

En términos de usabilidad, se ha habilitado un botón para permitir al usuario ver la contraseña para revisar si la está tecleando correctamente o no.

It is important to emphasise that to protect the security of the accounts, all passwords are hashed in the database and every time the administrator creates a new account, the password is also hashed to prevent it from being stored unencrypted and easily obtained by attackers.

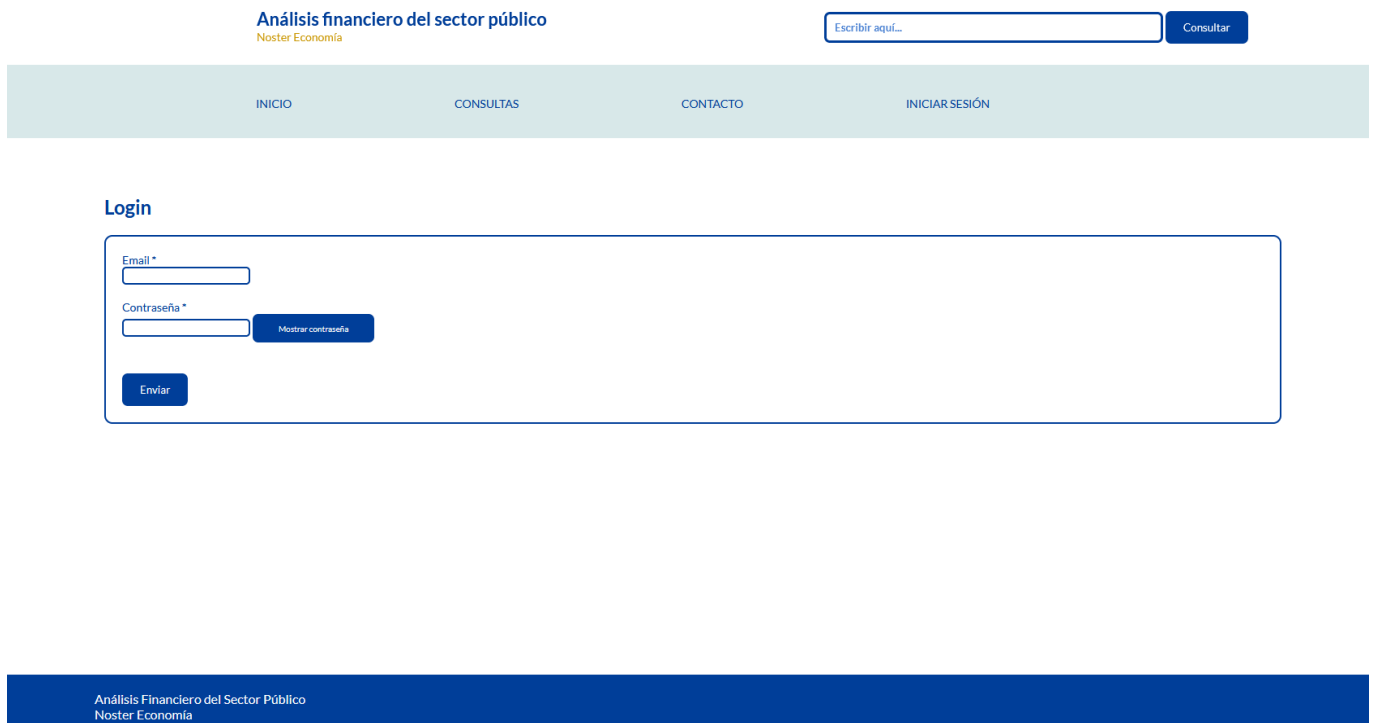


Fig.42: Login interface

Once the credentials are put and validated by the website, the user is redirected to the profile, where it is possible to modify the account information, such as the name, email or password. In the website there is a little role hierarchy, in which there two type of users:

- **Manager**, who can import Excel files in order to update the database (Fig.43).
- **Administrator**, who can add and remove accounts, apart from import Excel files (Fig.44).

Tu perfil

Nombre: admin
Email: admin@hotmail.com
Contraseña: admin4444
Rol: Administrador

Panel de control

Subir archivo

Selecciona el archivo:

Usuarios

Fig.43: Profile interface (Administrator account)

Tu perfil

Nombre: manager
Email: manager@hotmail.com
Contraseña: manager2022
Rol: Gestor

Panel de control

Subir archivo

Selecciona el archivo:

Fig.44: Profile interface (Manager account)

In addition, there is the import option where the user can upload the Excel files to update the database. The import option only accepts Excel files, otherwise they are discarded.

6.2 Access to the repository

The public repository can be accessed by through the following URL

<https://github.com/MiquelVera/AnalisisFinancieroSectorPublico>

The whole code can be downloaded by clicking the option Code → Download ZIP (Fig.45)

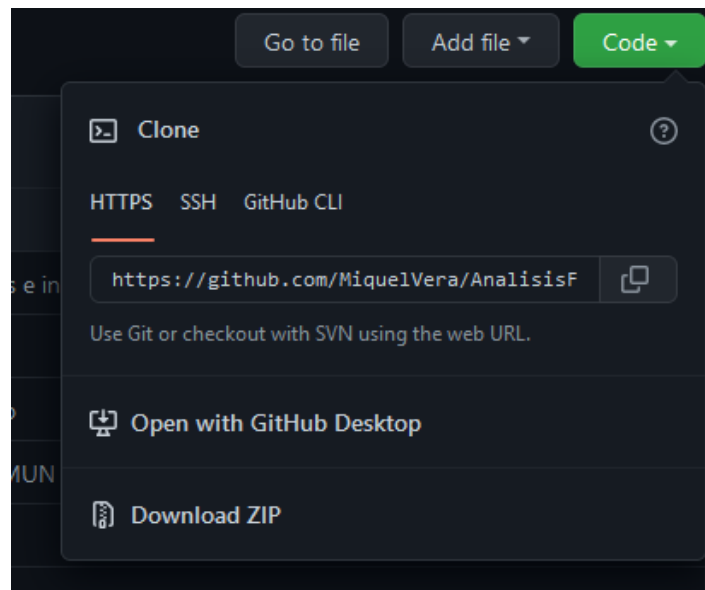


Fig.45: Download ZIP file

6.3 Usage of the code locally

In order to run the web application locally, it is necessary to download XAMPP (X, Apache, MariaDB/MySQL, PHP, Perl) with the latest version (<https://www.apachefriends.org/download.html>). Once the folder is downloaded, it must be decompressed (Fig.46).

XAMPP is an open-source package that contains mainly the MySQL database management system, the Apache web server and the PHP and Perl programming languages.



Versión	¿Qué está incluido?.	Suma de comprobación	Tamaño
7.4.29 / PHP 7.4.29	¿Qué está incluido?.	md5 sha1	Descargar (64 bit) 159 Mb
8.0.18 / PHP 8.0.18	¿Qué está incluido?.	md5 sha1	Descargar (64 bit) 161 Mb
8.1.5 / PHP 8.1.5	¿Qué está incluido?.	md5 sha1	Descargar (64 bit) 164 Mb

Fig.46: XAMPP download page

After that, a window will appear in case the user has an antivirus, informing that some functions of XAMPP will be disabled because of this antivirus (Fig.47).

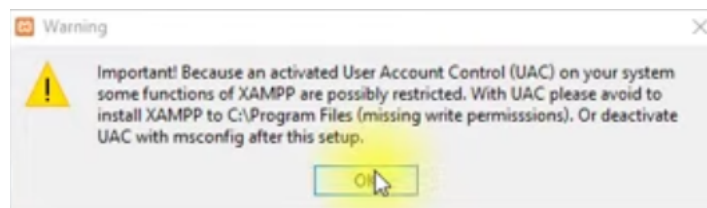


Fig.47: XAMPP warning

The following window will show all the XAMPP components. In this case, at least the Apache, MySQL, PHP and phpMyAdmin options must be activated. The rest are optionally but it is recommended to maintain them activated (Fig.48).

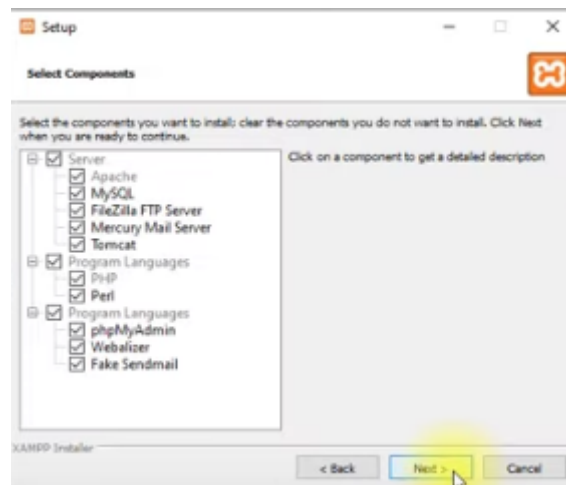


Fig.48: XAMPP components

Then, the installation folder is selected. It is also recommended to leave the default folder (Fig.49).

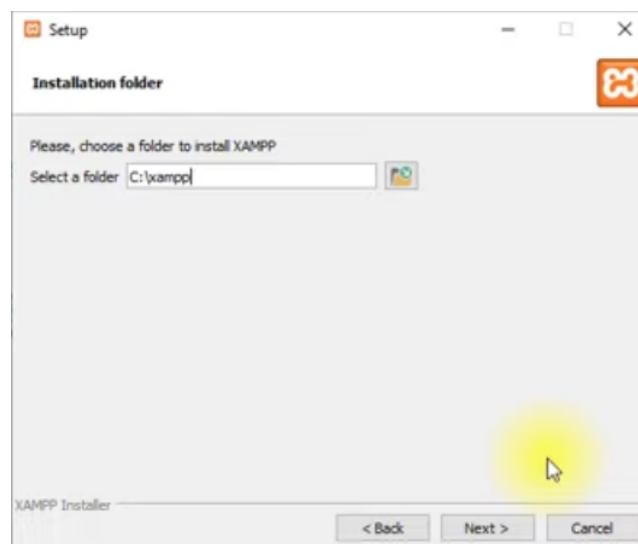


Fig.49: Installation folder selection

The following window is about Bitnami for XAMPP. The checkbox must be disabled (Fig.50).

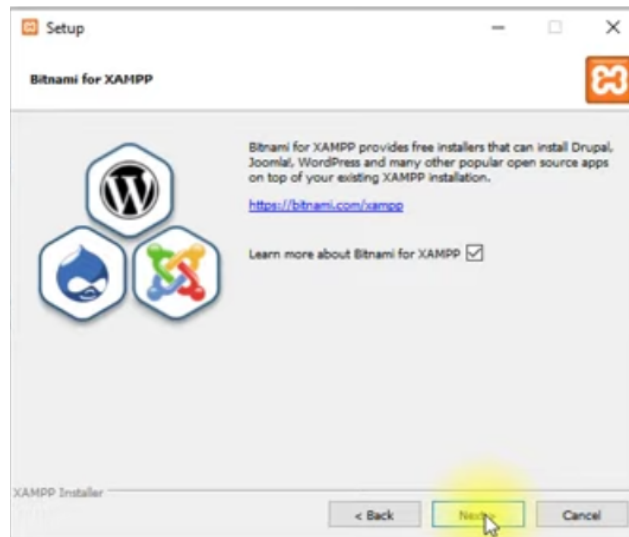


Fig.50: Bitnami for XAMPP

After that, the *Next* button from the rest of the windows must be clicked until the installer is closed. Then, the control panel opens automatically (Fig.51).

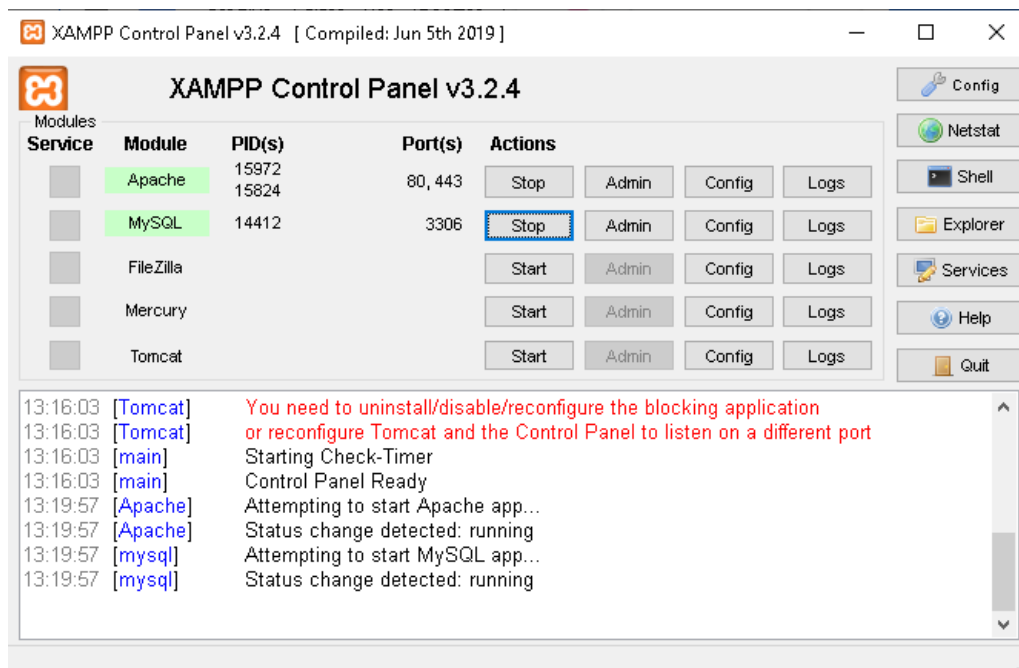


Fig.51: XAMPP control panel

The Apache and MySQL options should be started by clicking their corresponding *Start* buttons and it is waited until both servers are in green. This panel control can also be accessed by executing the `C:\xampp\xampp_control.exe`.

In order to prove the web application in XAMPP, the downloaded project file must be decompressed in the `C:\xampp\htdocs` folder. After that, it is needed to create a local database with all the information from the website there. For this, three SQL files must be imported in phpMyAdmin in order. All of them are located in the folder *bd*:

1. `Db_s_01_create.sql`: creates the database.

2. Dbs_01_tables.sql: creates all the tables of the database, establishing the primary and foreign keys, and the relationships between them.
3. Dbs_01_data_ccaa.sql: inserts the data of the autonomous communities into the corresponding tables.
4. Dbs_01_data_dip.sql: inserts the data of the councils into the corresponding tables.
5. Dbs_01_data_mun_general.sql: inserts the general information of the municipalities into the corresponding tables.
6. Dbs_01_data_mun_finanzas.sql: inserts the economic information of the municipalities into the corresponding tables.
7. Dbs_01_data_mun_deuda_scoring.sql: inserts the scoring and debt information of the municipalities into the corresponding tables.
8. Dbs_01_data_usuarios.sql: inserts the data from the users into the corresponding tables. In fact, this file can be imported at any moment after the creation of the tables, since the users table does not have any relation with the other ones.

The municipalities data had to be divided into three SQL subfiles due to the large size of some of their tables, such as *municipios*.

To import all the previous files, the user must enter the local database by entering the following URL in any browser: localhost/phpmyadmin (Fig.52).

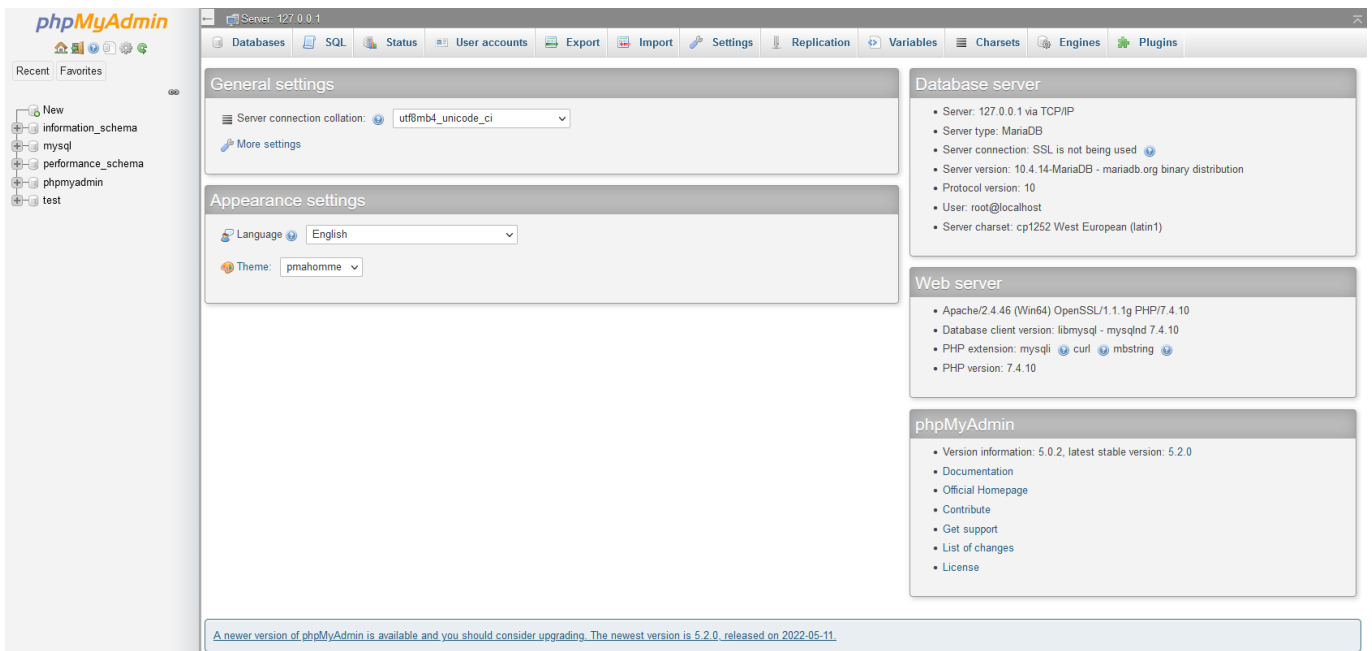


Fig.52: PhpMyAdmin control panel

Then, enter the *Import* option and click to the *Browse* button in order to import the *dbs_01_create.sql* file into phpmyadmin. Finally, after clicking *Go*, the database is created (Fig.53).

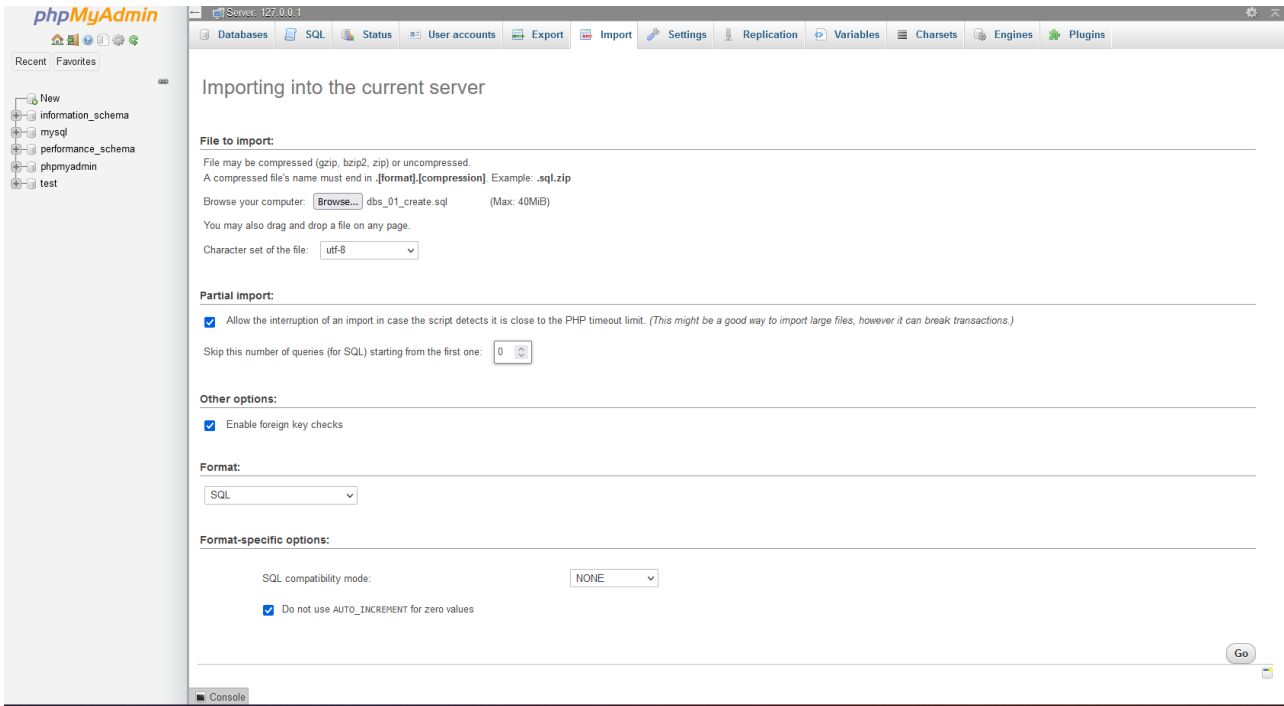


Fig.53: Import option in phpmyadmin

Then, the `db_01_tables.sql` should be imported into the already created database, but it is necessary to enter the database (Fig.54).

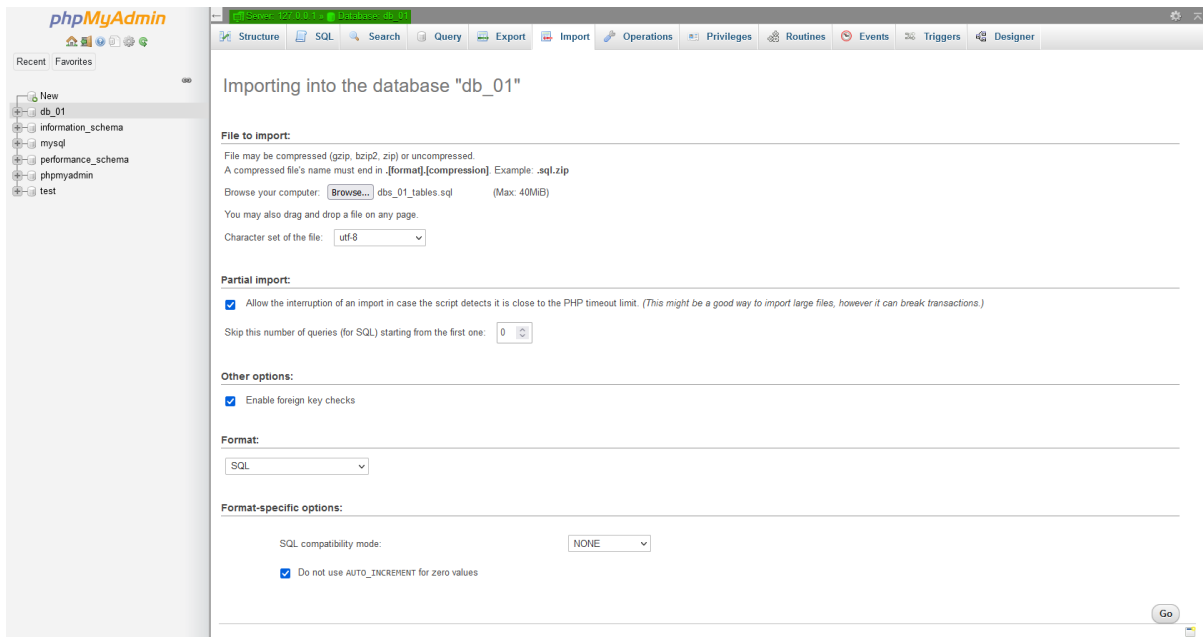


Fig.54: Import option inside the `db_01` database

Then, enter the *Import* option and click to the *Browse* button in order to import the `db_01_tables.sql` file into phpmyadmin. Finally, after clicking *Go*, the tables are created inside the `db_01` database (Fig.55).

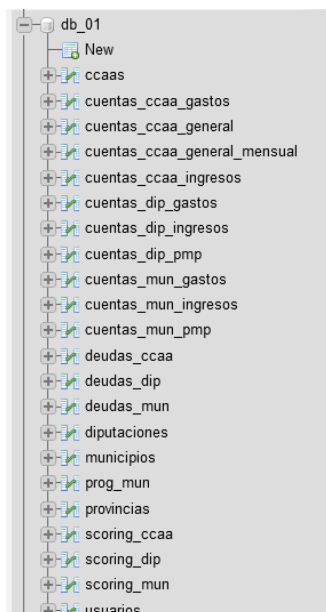


Fig.55: Tables from dbs_01 database

Once the tables are created, the remaining files are imported in the same way and following the previously mentioned order.

Finally, it is put in the browser the URL *localhost/AnalisisFinancieroSectorPublico/index.php* and it will be redirected to the webpage locally. After this, the web application will work as in the server.

It is important to say that once the user wants to finish the execution of the web application, it is not enough to only close the panel control, but it must be disabled the Apache and MySQL servers by clicking the *Stop* button (Fig.51).

7. Conclusions

7.1 Conclusions

During the development of the application, it could be observed that obtaining public sector financial data is complicated, because although all data are public, the sources are very diverse and disorganised. This makes the data collection process tedious.

On the other hand, it was also necessary to understand certain economic concepts in order to develop the application correctly, such as knowing the meaning of all the ratios of public entities, the units that should be used to represent these ratios.

In this project it has been possible to apply several knowledge acquired during the degree, such as the design, creation and management of relational databases, web programming with PHP and Javascript, web security to avoid possible attacks, usability engineering, software engineering and agile methodologies to be able to identify the functionalities and organise the work through sprints, and knowledge to know and decide which open source libraries should be used to avoid any copyright infringement.

This Final Degree Project allows users to have access to data in an easier way, knowing more easily the financial situation of a given public entity. In addition, it helps users to better understand the economy of the public entity by using a very intuitive range of grades, from A (the highest grade) to E (the lowest grade), as well as including graphs that allow a clearer visualisation of the evolution of the ratios of an autonomous community, county council and municipality.

There has also been added the possibility of making queries on the website through the queries page or through the API.

In addition, the website allows authorised users to update their data continuously by importing new datasets into the database via the control panel.

7.2 Future of the project

First of all, it is considered relevant to have an application that can be adaptable to any screen size. For this reason, it is planned in the most recent future to modify the styles in order to make the web application responsive.

Additionally, once the previous goal is achieved, in order to improve the positioning and indexing of the application on the Internet, SEO techniques will be used. However, it is also necessary to have a dedicated server in IONOS in order to accomplish it, because the server was shared during this project development.

In addition, it is planned to improve the contact page by implementing a form and allowing the users to send emails to NosterEconomia about doubts and complaints. Initially, it was supposed to be implemented. However, it was discarded mainly because a captcha is necessary to avoid bots, and it involves registering in the reCAPTCHA web page <https://www.google.com/recaptcha/admin/create>, sending the website to Google to receive the keys and implementing it in the code.

On the other hand, another element that is planned to be implemented in the long term is an interactive map in the main page. In this way, the user will be able to search for a place in a more interactive way. The map must show autonomous communities, councils and municipalities as the user zooms in the map. Once the user clicks on one facility, it would show the most recent rating and all their ratio scores, and if the user clicks again on the facility name, then it is redirected to the facility information webpage.

Besides, the PDF is also expected to be improved by implementing graphics and more conclusions by analysing the ratios, apart from some other stetical changes to the document, such as the header, footer and content organisation. Furthermore, another feature that is planned to improve is the API by adding more requests, since it has implemented general getters from the facilities for the moment.

Finally, another feature that is likely to be implemented in the long term future is a regressive function and Big Data, in order to predict the most probable economical situation and rating of the facility in the following years.

8. Licence

8.1 Licence

As this project has been developed as our thesis and its basic motivation is to make spanish's public facilities economic data more accessible, we find convenient to apply a free software licence

At the same time, our code's licence depends on the licence of the set of libraries used for its development. Said licences are the following:

- **TCPDF:** *GNU LGPL (Lesser General Public Licence)*
- **PHPSpreadSheet:** MIT
- **ChartJS:** MIT

MIT Licence gives us free use of the software and full liberty on the licence we apply on our project. On the other hand, GNU LGPL also allows us to use the software as we like as long as the project's redistribution or modifications use the same licence.

Taking all this into account, we find it convenient that our project maintains this **GNU Lesser General Public Licence**, allowing users to use it as they wish, and only affecting its licence once they modify or redistribute it.

9. Bibliography

9.1 Bibliography

- <https://www.w3schools.com/>
- <https://stackoverflow.com/>
- <https://developer.mozilla.org/es/>
- Documentation
 - PHP library: <https://www.php.net/>
 - PHPSpreadSheet library: <https://phpspreadsheet.readthedocs.io/en/latest/>
 - TCPDF library: <https://tcpdf.org/docs/>
 - Chartjs library: <https://www.chartjs.org/docs/latest/>
 - XAMPP: <https://www.apachefriends.org/es/index.html>
 - Visual Studio Code: <https://code.visualstudio.com>
 - IONOS: <https://www.ionos.es>
 - Postman: <https://www.postman.com>