MÓDULO PROYECTO

CFGS Desarrollo de Aplicaciones WEB
Informática y Comunicaciones

Factory Management

Tutor individual: Jose Antonio Carrascal Alderete

Tutor colectivo: Ana Rosa Hoyos Teran

Año: 2024/2025

Fecha de presentación: 20 de Mayo de 2025

Nombre y Apellidos: Unai Benito Sánchez

Email: unai.bensan@educa.jcyl.es

1.   Identificación del proyecto

It is a project to manage aspects of a factory, in the part of employee would be a management of the products (add, edit and delete) and access to the internal chat of the factory and, in the part of the bosses, would be a management of the products, the graphs of production of these, access to the internal chat of their factory, management of the employees of the factory, estimation of necessary quantity of a product at the beginning of the following day not to exhaust the stock of this one and a control of the data of the factory (location, name, quantity of employees and a view of the bosses of the factory).

2.   Organización de la memoria

3.   Descripción general del proyecto

Factory management project, in which employees can manage their factory's products and access their factory's chat. When an employee adds a product, it remains in review, pending a boss of his factory to validate the product, but, edit and delete products you can do it without problem. The boss can do the same with the products but they are also the ones able to validate them, they can also access the company chat and they can also see a production graph of the products, an estimate of the amount of products they need at the beginning of the next day to avoid running out of stock, a visualization of the aspects of their factory and finally they have access to a visualization of the employees of their factory, modify their name or their access email and see their connection status (online or offline).

3.1.   Objetivos

My goal in developing this project is to strengthen my skills in the programming languages used, as well as to be able to add things and make a big project by making everything work well. Other than that, I have tried to expand my creativity by being able to come up with things that I could add to the project that would look good.

3.2.   Cuestiones metodológicas

First I made the base functionality that I wanted the web page to have, later I thought about what more things I could implement because it did not seem to me enough functionality so I thought about what I could put in and how to implement it and I looked for information to do it.

3.3.   Entorno de trabajo (tecnologías de desarrollo y herramientas)

I deployed my tfg with a remote host on my pc.
I have used several technologies:
-   PHP: Backend part of the application.
-   JavaScript, CSS and Bootstrap: Frontend part of the application.
-   Python: Handles the section of the prediction of how much of the product would be needed the next day (or also called "Demand Prediction").
-   Ruby: I have created in the "Ruby" language a file that, when I run it, I enter a URL and it transforms it into a QR code, to make it easier for people to enter the application on the day of the presentation.

And IDE:
-   I've used Visual Studio Code, it's the one I've been using for the whole 2nd year of DAW and it's the one I'm most comfortable with and the one I find easiest to use.

4. Descripción general del producto

4.1. Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar.

- System limits: It is an autonomous application that works by itself.
- Basic functionalities: It performs user registration and login, product management, application user management, interactive chat between users and visualization of aspects of the factory.
- Users: There are 2 types of users, the employees, who are the ones with product management functionality and access to the interactive chat of the application, and the bosses, who are the ones with the "admin role", who can manage the whole application.
- The most notable implementation of my project in terms of design may be in the data section of the factory, depending on the number of bosses in the factory, which appears a certain number of 3d models of people. In case there are 2 bosses in that factory, 2 3d models of people appear. This has been implemented with a thing called Three.js, used for the implementation of 3d models in a web page.

4.2. Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas.

I have used a MVC (Model-View-controller) architecture used with PHP to manage both the model and the view because certain aspects of the view needed to be implemented in my application with PHP.

Main techniques:

- Password encryption: The password is encrypted when saving it to make it more secure.
- Production graphs: Using the input and output records in the database, graphs are generated indicating the variation of a product in the factory.
- Demand prediction: Calculates the quantity of a product that should be available at the beginning of the next day to avoid running out of stock.

Work methods and techniques:

- The process is constantly being reviewed and improved: As I finished implementing something, I would test it several times and fix any bugs that there might be constantly.
- Sequential work: analysis → design → implementation → testing: Once I finished implementing and reviewing something, I would think and research how to implement the next functionality I wanted to implement in the application.
- Draft versions are created to validate ideas: In a new repository I would do the testing of the big changes I was implementing so, in case everything went wrong, it wouldn't break my project.
- It requires exploring new technology or solving complex problems: Like for the implementation of Ruby, which I didn't know much about, and Python, which I used a long time ago but there were things I didn't remember.
- UML modeling: use cases, class diagrams, sequence, activity
- Database design: entity-relationship model.
- Interface design: wireframes, prototyping with Figma.
- Version control: use of Git and GitHub.

4.3. Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha

At the time of deploying the application I investigated and discovered Cloudflare, installing it on the computer (Using Windows 11 through Powershell and testing it. First you must start the servers you need for your application (as can be the Angular, Apache…), then access the CMD and start from there the application with cloudflare deployment commands, which will generate a tunnel from your ip and a url that accesses the project through that tunnel. Because the url generated by this service is a very long url (since it is free you cannot choose your own url, it generates a succession of 4 random words separated by dashes "-" and then ".trycloudflare.com/(path to your project)"), I have created the QR generation file written in Ruby language so that people in the presentation do not have to enter that long url and just by scanning the code will serve them (I will also use a page called "bit.ly", which has limited free uses per month, to shorten the url in case people prefer to write a short url instead of scanning the qr).

5.   Planificación y presupuesto

Planning

The project planning has been distributed over several months, following a structure organized by key tasks. An approximate description of the time devoted to each phase is shown below:

Month	Principal activity	Estimated hours
January	Requirements analysis and planning	20 h
February	Design and database optimization	30 h
March	Backend and frontend optimization	60 h
April	Extras implementation and searching new ideas	50 h
May	Final tests, bug fixes, deployment and documentation.	40 h
Total	—	**200 h**

Budget

The following is an estimate of the total cost of the project, considering the hours of work, tools and services used:

Code development costs:

- Dedicated hours: 200 h

- Estimated price: 8 €/h

- Subtotal: 1.600 €

Cost of software and tools:

Herramienta	Observaciones	Coste aproximado
Windows 11 Pro	Profesional license	145 €
Visual Studio Code	Free	0 €
XAMPP / WAMP	Free	0 €
Git + GitHub	Free (public repository)	0 €
Figma	Free (basic plan)	0 €
Total software	—–-	**145 €**

Hardware cost:

Element	Observations	Estimated cost
Portable computer	Mid-range device	700 €
Total hardware		**700 €**

Hosting cost:

Service	Provider	Anual estimated plan	Cost
Shared host	Hostinger	Basic plan (~2,49 €/month)	29,88 €
Domain .com	Hostinger	Included on the plan	0 €
Total hosting			**29,88 €**

Final budget summary:

Concept	Estimated cost
Code development	1.600 €
Software and tools	145 €
Hardware	700 €
Hosting and deployment	29,88 €
Total of the project	**2.474,88 €**

Gantt diagram:

|	January	February	March	April	May	Time spent
Analysis						5 h
Planning						15 h
Database optimization						10 h
Design optimization						20 h
Backend optimization						30 h
Frontend optimization						30 h
Extras implementation						50 h
Deployment						10 h
Final tests, bug fixes and documentation						30 h
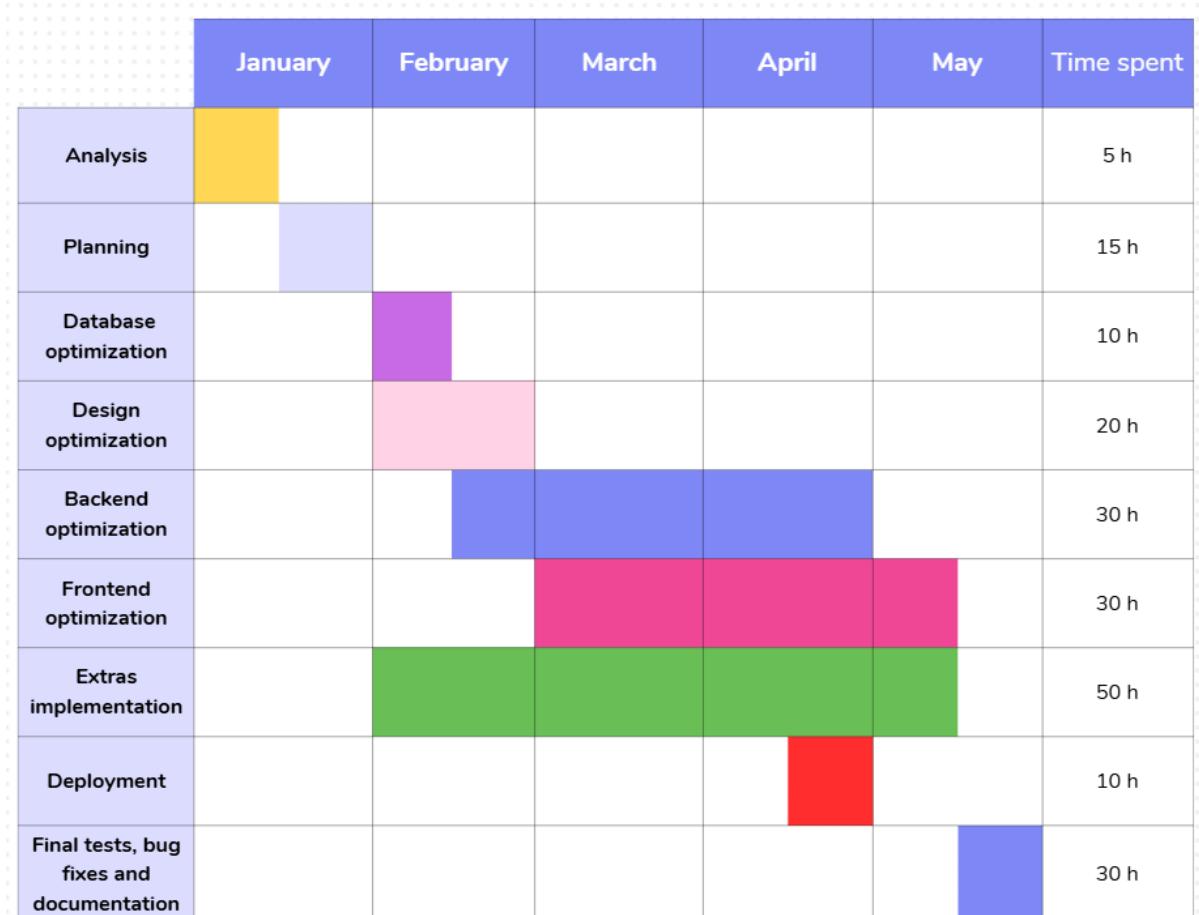
6. Documentación Técnica: análisis, diseño, implementación y pruebas.

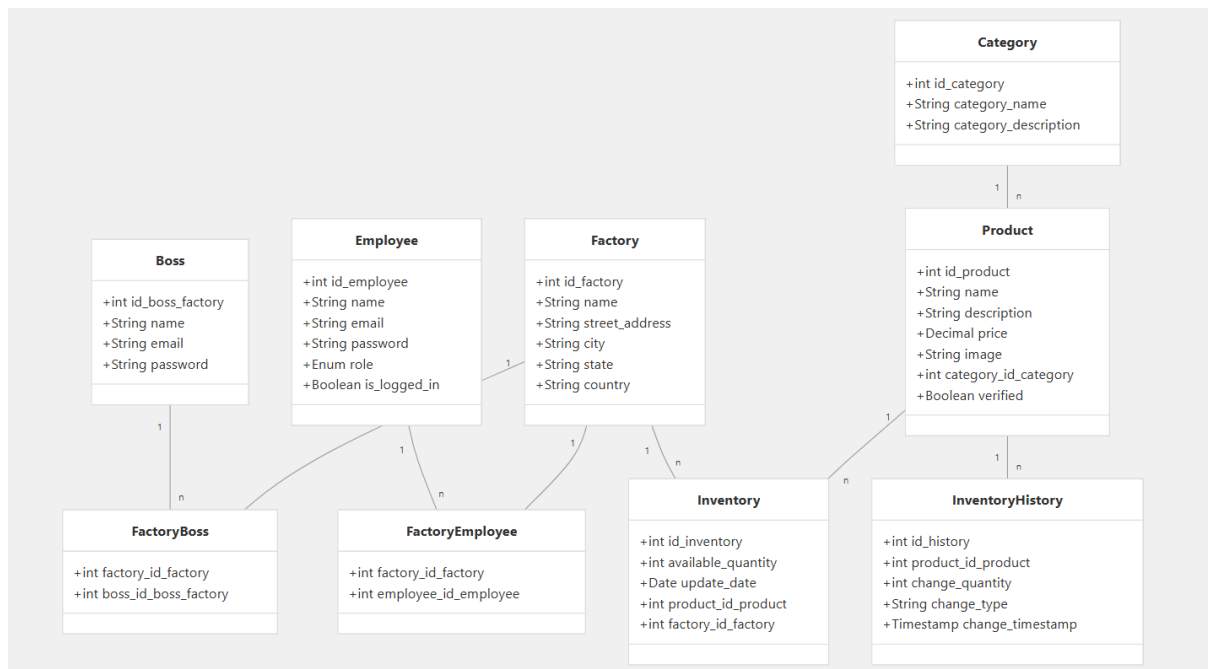## 6.1. Especificación de requisitos

At least there has to be CRUD (Create-Read-Update-Delete) and things we have used in class (some of the languages like PHP or JavaScript, both implemented in my project). I was also asked for the confirmation messages to be custom messages instead of the typical web page alert.

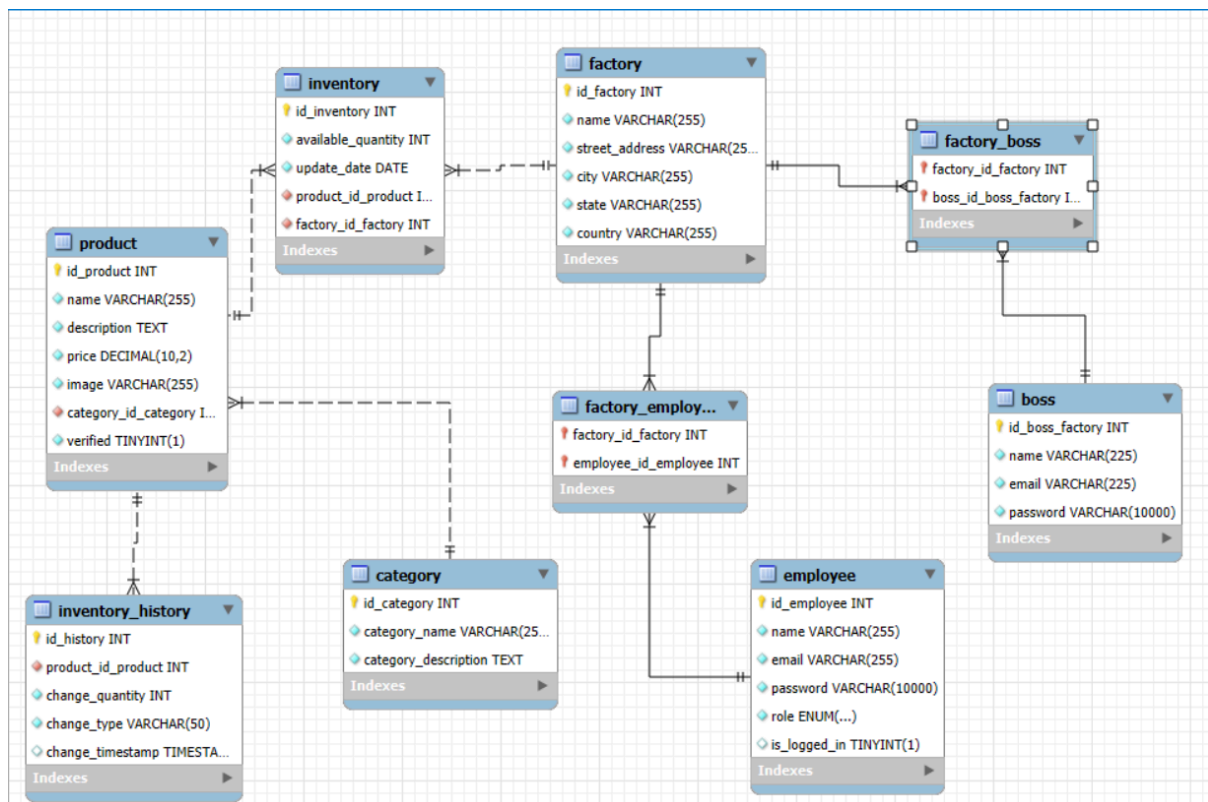## 6.2. Análisis del sistema

Use case diagram:
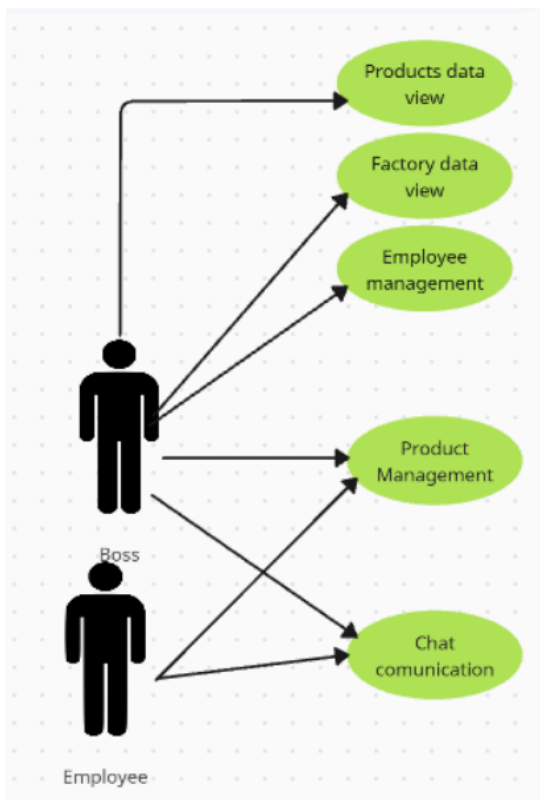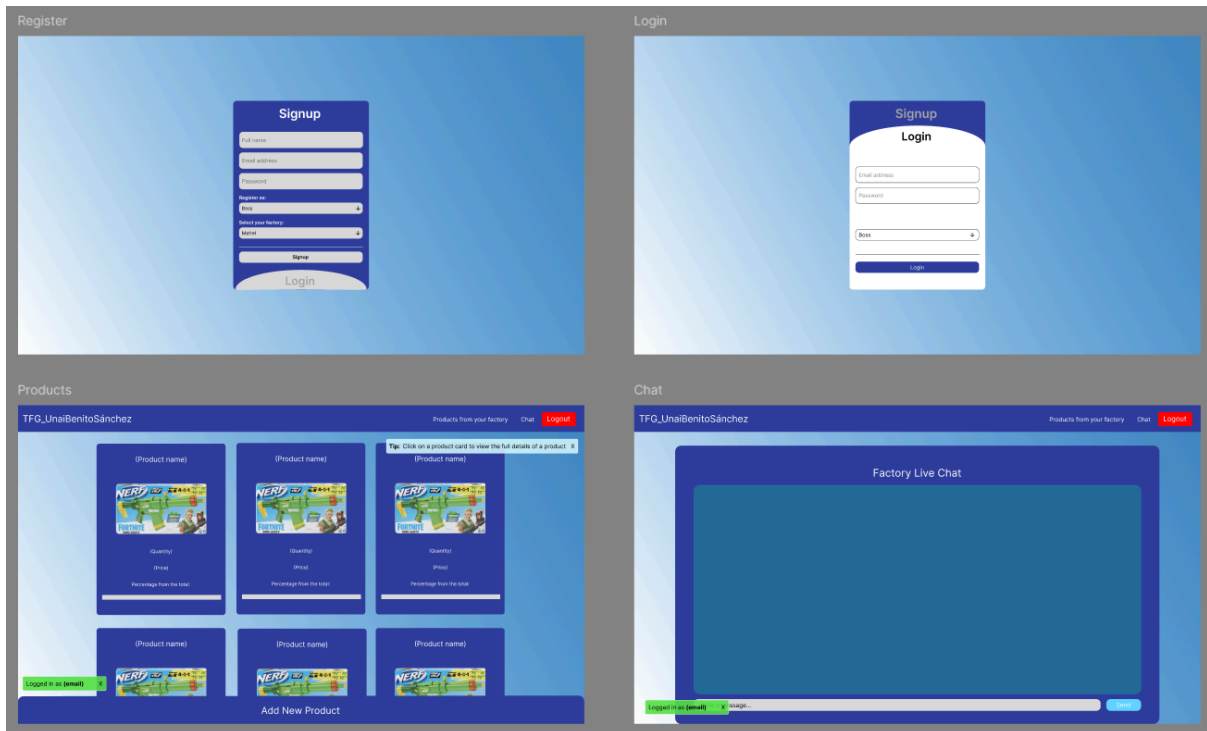
Class diagram:

6.3. Diseño del sistema

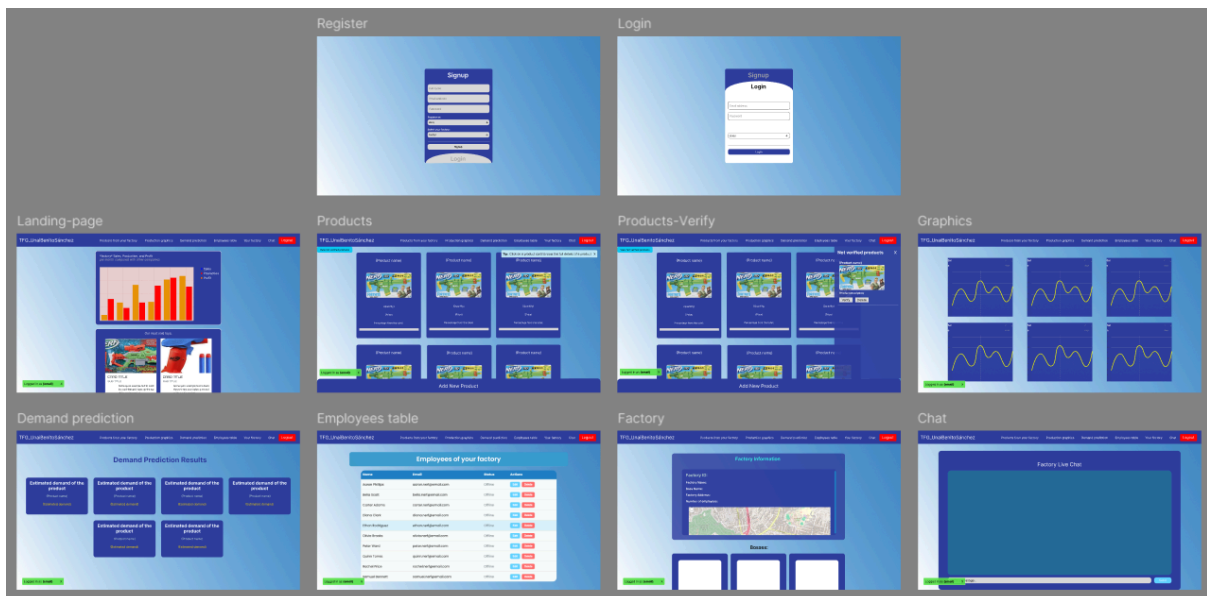## 6.3.1. Diseño de la base de datos

Database:

6.3.2.   Diseño de la interfaz de usuario

Employee view:

Boss view:

Link to see directly the Figma, where you can see better the components used several times, the view that the employee has and the view that the boss has, with the relations between pages declared in each one of them:
https://www.figma.com/design/jlWnoX6UQZfDML3iOC1VzP/TFG_UnaiBenitoS%C3%A1nchez?node-id=0-1&t=EyRbB2KBnsf8nKPv-1

6.3.3.   Diseño de aplicación

The structure of the application is as follows:

- It consists of a login/registration page as a start, there you can register either as a boss or an employee, there you register or log in and if you enter as a boss you will fall into a landing page from which you can access the other pages and boss and also you have the logout button so you can go back to the registration and login page, thus logging out. If you enter as an employee you will land on the product management page and then you can also access the chat and you also have the logout button and you can logout and go back to the registration/login page.
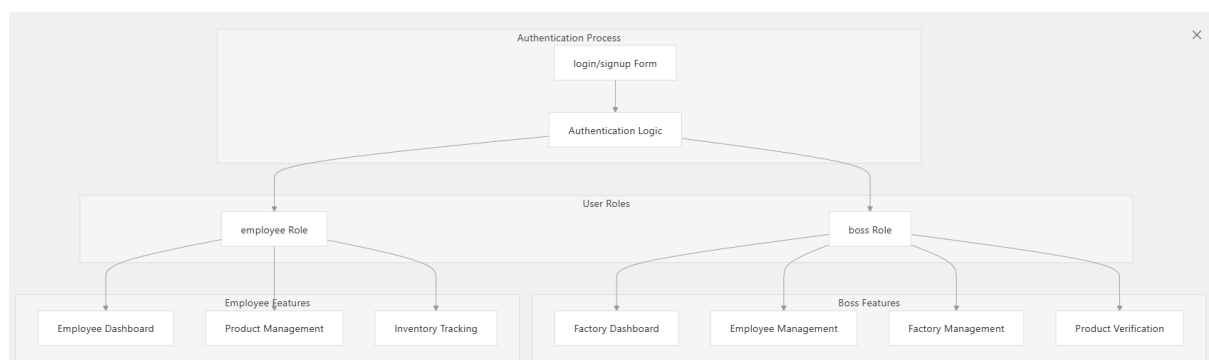
6.4.   Implementación
6.4.1.   Entorno de desarrollo

I have developed the application on a computer running Windows 10 operating system and subsequently upgraded to Windows 11 with the following features:

Device specifications:
- Processor: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
- RAM: 16.0 GB (15.7 GB usable)
- System Type: 64-bit operating system, x64-based processor

Windows specifications:
- Edition: Windows 10 Pro (before) / Windows 11 Pro (now)
- Version: 24H2

Programming software:

- PHP: Main language of the application, handles the view and model, the connection to the database and the bindings with the other languages.

- JavaScript: Handles part of the frontend to display the modals, do the fetches to the php's that collect things from the database and the display of the data on the page.

- CSS: The other part of the frontend of the application that puts the styles in the application.

- Bootstrap: Used to help me with the design in some of the sections of the web page. It's the one I've used the least, most of it is with CSS and JS.

- Python: Used with a Linear Regression to calculate the quantity of a product to have at the beginning of a day so I don't run out of stock of it.

- Ruby: Used for programming a website that when entering a url will generate a QR code to access it.

- Github: I have created a repository to save my project, do version control and have the repository on the devices used for development.

- MySQL Workbench: Control of the database, which I have developed with MySQL.

6.4.2. Estructura del código

<u>Python libraries:</u>

Standard libraries (already included with Python):
- JSON
- sys

These come by default with Python, so you don't have to install them.

External libraries (you do need to install):
- mysql-connector-python
  To connect to the MySQL database, install with:
  pip install mysql-connector-python

- numpy
  For handling numerical arrays, used here for model training, install with:
  pip install numpy
- scikit-learn (o sklearn)
  To use the linear regression model, install with:
  pip install scikit-learn

<u>Ruby libraries:</u>

Gems required:

- sinatra
  Lightweight web framework for Ruby, install with:
  gem install sinatra

- rqrcode
  Librería para generar códigos QR, instalar con:
  gem install rqrcode

- puma (optional but recommended)
  Library to generate QR codes, install with:
  gem install puma

- rack
  It is already dependence of Sinatra, but if missing, install with:
  gem install rack

<u>Orientation:</u>

My project is a combination of structured, event-driven and partially object-oriented programming.

6.4.3.   Cuestiones de diseño e implementación reseñables

The most remarkable implementation of my project in terms of design may be in the data section of the factory, depending on the number of bosses in the factory, a certain number of 3d models of people appear. In case there are 2 bosses in that factory, 2 3d models of people appear. This has been implemented with a thing called Three.js, used for the implementation of 3d models in a web page.

Cartamundi Shaker Road, East Longmeadow, MA 01028, USA

Bosses:
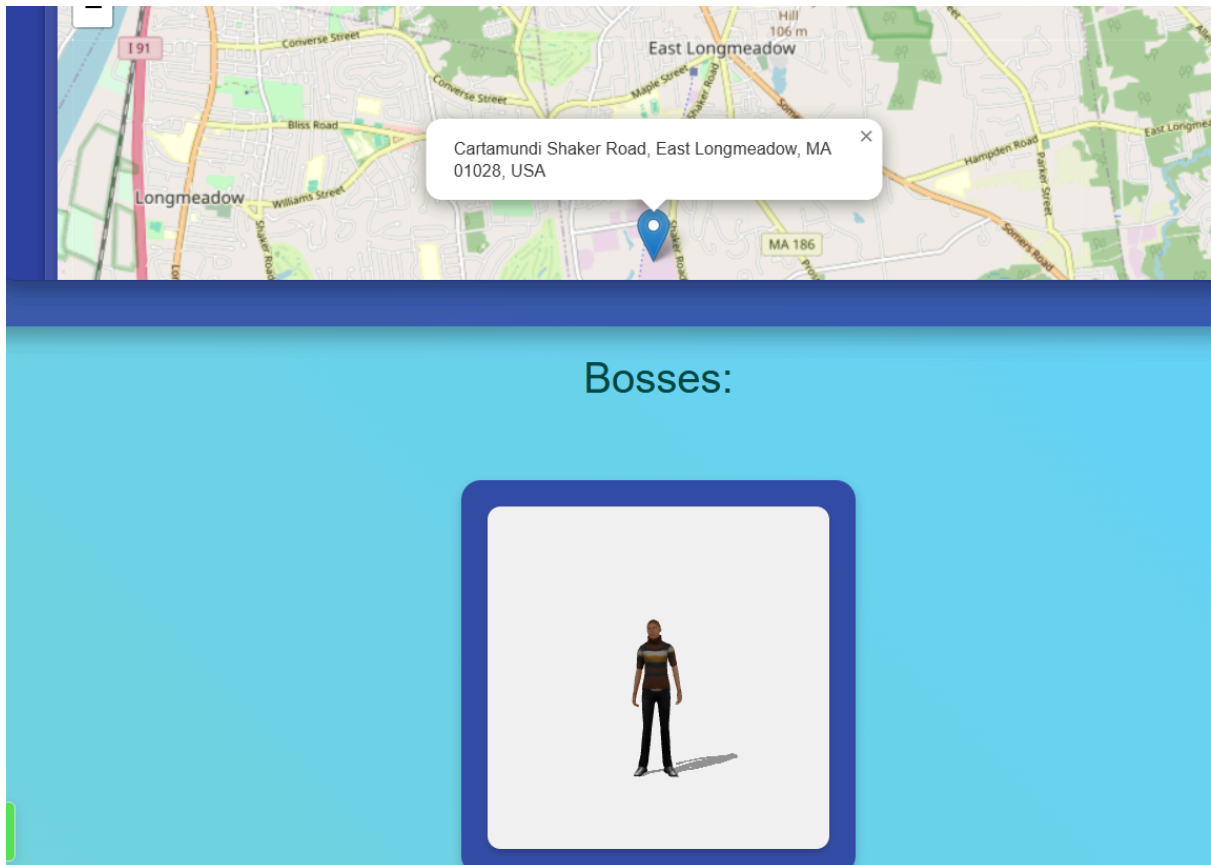
6.5. Pruebas

Try on Windows in Google Chrome:

(Width >1480):
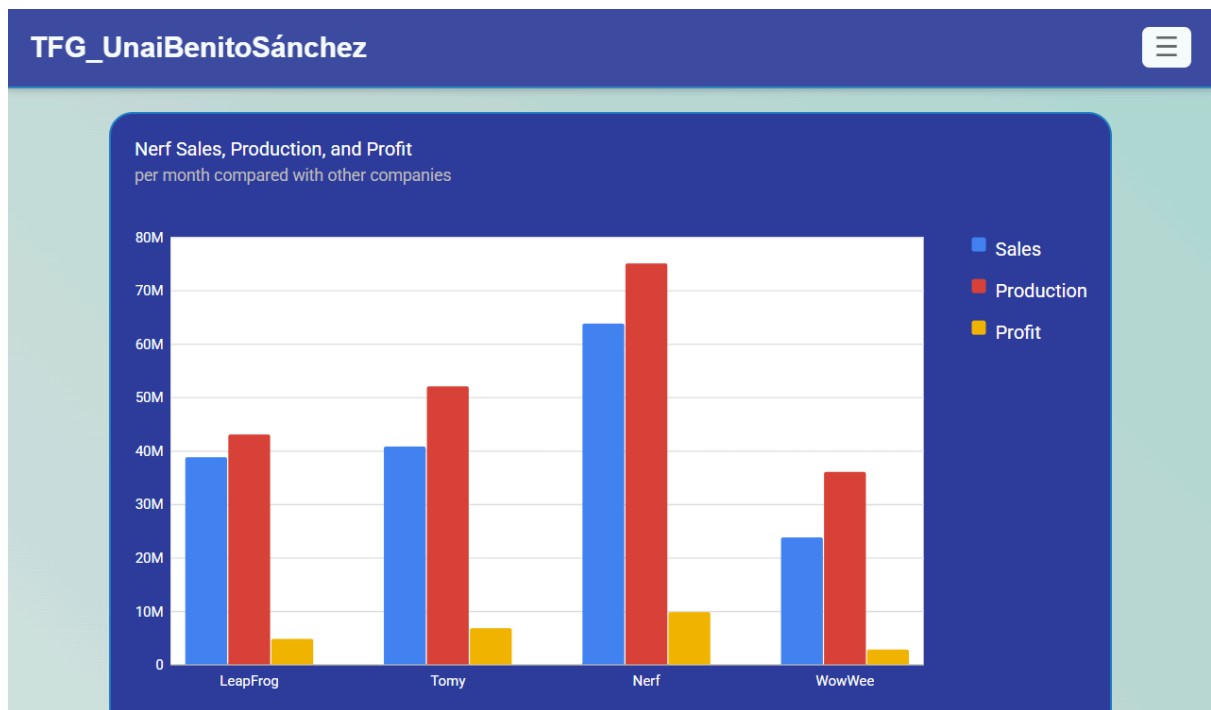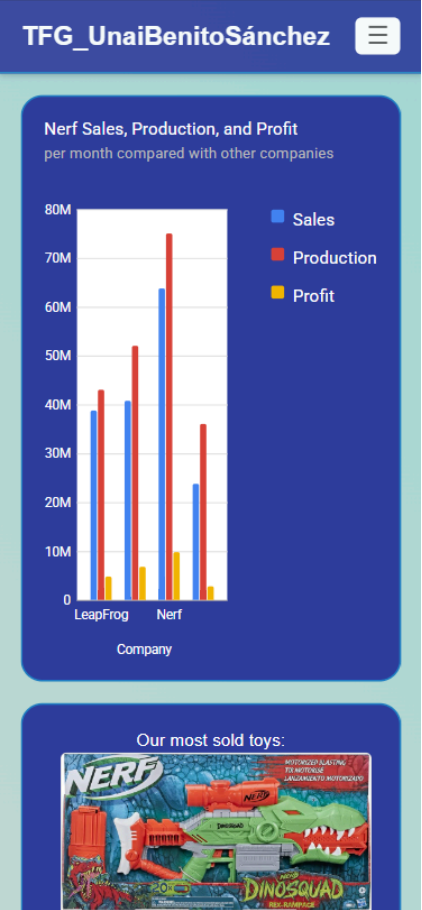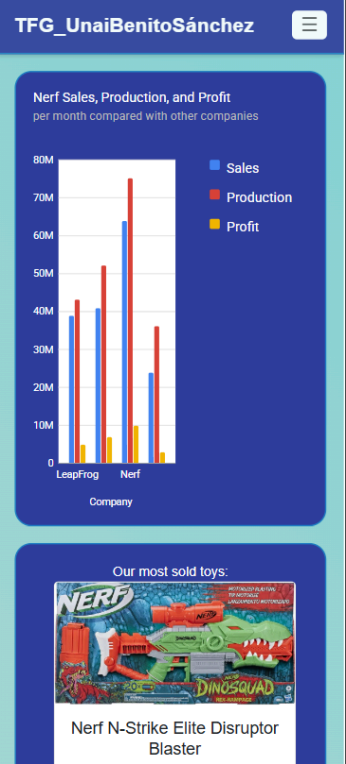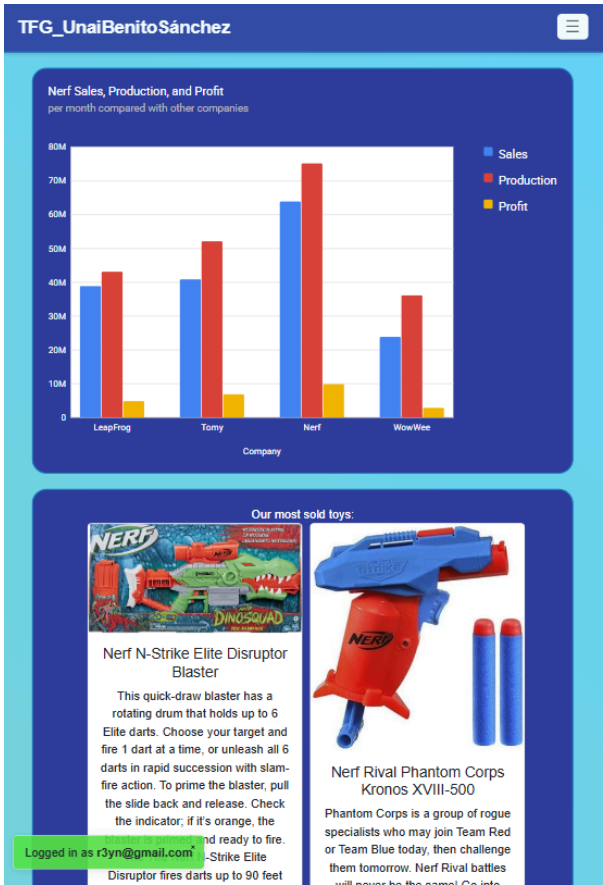
(Width <= 1480):

Try on BlackBerry PlayBox:

Try on Iphone 12 Pro:

Try on Samsung Galaxy S20 Ultra:

Try on Ipad Air:

7. Manuales de usuario

7.1. Manual de usuario

(Attached as a separate file)

7.2. Manual de instalación

(Attached as a separate file)

8. Conclusiones y posibles ampliaciones

-Conclusions:

Compared to the initial approach of the project, which was to make a management of the products and factory data, I have added more things, so I can take as a conclusion that maybe what I had thought in the beginning was not ambitious enough or I did not feel fully realized with what I had planned. I have to admit that I would have liked to add something else, but time has a limit and we can't always do everything we want. In case of using this project for something more professional is when I could implement the extensions and improvements discussed below.

Another conclusion that I have been able to draw is that it is good to be ambitious but that also must have a limit, since there have been times that I thought about adding things that later were very complicated and I was overwhelmed thinking that I would not have time to do the project by adding them.

-Possible expansions that i could do to improve the project:

- **1. Demand Forecasting System Enhancements**
  The current system uses simple linear regression to predict future product demand, could be extended with:
    - Implementation of more advanced machine learning algorithms (neural networks, time series models).
    - Incorporation of external variables such as seasons, events or market trends.
    - Development of more interactive visualizations of forecasts.

- **2. Notifications and Alerts System:**
  Currently there is no alert system. A system could be implemented that:
    - Notifies when inventory is below certain limits
    - Alerts on products in high demand based on forecasts
    - Send reminders about pending tasks to employees

- **3. Chat System Expansion:**
  The current system allows basic communication between members of a factory.
  chat.php:

- Could be expanded with: Ability to share files and photos.
- Creation of thematic groups within the factory
- Integration of push notifications
- Conversation history and search

- **4. Mobile Application Development**
    - Creation of iOS and Android versions to enable on-the-go management and monitoring
    - Ability to scan barcodes/QR codes to update inventory from cell phones
    - Push notifications for important alerts

- **5. Product Management Enhancements**
Current system allows managing commodities, could be extended with:
    - More detailed product categorization
    - Customizable labeling system
    - Batch and expiration date tracking Management of physical locations in the warehouse
- **6. Integration with External Systems**
    - API for integration with ERP systems
    - Connection with supplier platforms to automate orders Integration with e-commerce systems for inventory synchronization

- **7. Logistics and Shipping Module**
    - Tracking of shipments to customers
    - Management of distribution routes
    - Calculation of shipping costs Route optimization

- **8. Financial Management Module**
    - Tracking of production costs
    - Profitability analysis by product
    - Financial forecasts based on demand forecasts
    - Automatic invoice generation

- **9. Enhanced Authentication and Security**
Current system has basic authentication, could be extended with:
    - Implementation of Single Sign-On (SSO) two-factor authentication with corporate services
    - Audit trail of critical actions
    - More granular roles and permissions.

- **10. Advanced Analysis Panel**
    - Customizable dashboards for different roles Interactive data visualizations
    - Export of reports in different formats

- Configuration of customized KPIs for monitoring

- **11. Equipment Maintenance System**
  - Machinery life cycle tracking
  - Preventive maintenance scheduling Incident and repair registration
  - Automatic alerts for scheduled maintenance

- **12.Human Resources**
  - Management for the Factory Time and Shift Control
  - Competency and Skills Management
  - Optimized Task Assignment
  - Performance Evaluation

9. Bibliografía

WebTechnologies

- **PHP:**

  - PHP Group. (2024). PHP: Hypertext Preprocessor. https://www.php.net/

  - Welling, L., & Thomson, L. (2022). PHP and MySQL Web Development (6th ed.). Addison-Wesley Professional.

  - Lerdorf, R., Tatroe, K., & MacIntyre, P. (2023). Programming PHP (4th ed.). O'Reilly Media.

  - Lockhart, J. (2023). Modern PHP: New Features and Good Practices (2nd ed.). O'Reilly Media.

  - Stauffer, M. (2022). Laravel: Up & Running: A Framework for Building Modern PHP Apps (3rd ed.). O'Reilly Media.

  - Zandstra, M. (2022). PHP Objects, Patterns, and Practice (6th ed.). Apress.

  - PHP Manual. (2024). PHP Security Best Practices: https://www.php.net/manual/en/security.php

  - Sklar, D. (2022). Learning PHP: A Gentle Introduction to the Web's Most Popular Language. O'Reilly Media.

  - Sturgeon, P., & Lockhart, J. (2023). PHP The Right Way. https://phptherightway.com/

- **CSS:**

  - Bootstrap Team. (2023). Bootstrap v5.3.2: The world's most popular front-end open source toolkit. https://getbootstrap.com/docs/5.3/

  - McFarland, D. (2023). CSS: The Missing Manual (5th ed.). O'Reilly Media.

  - Frain, B. (2022). Responsive Web Design with HTML5 and CSS (4th ed.). Packt Publishing.

- Meyer, E. A., & Weyl, E. (2021). CSS: The Definitive Guide (5th ed.). O'Reilly Media.

- Gasston, P. (2022). The Book of CSS3: A Developer's Guide to the Future of Web Design (3rd ed.). No Starch Press.

- Duckett, J. (2021). HTML and CSS: Design and Build Websites (2nd ed.). Wiley.

- World Wide Web Consortium (W3C). (2024). CSS Specifications. https://www.w3.org/Style/CSS/specs.en.html

- Holzschlag, M. E. (2023). The Zen of CSS Design: Visual Enlightenment for the Web (2nd ed.). New Riders.

- Verou, L. (2022). CSS Secrets: Better Solutions to Everyday Web Design Problems (2nd ed.). O'Reilly Media.

- Lindley, C. (2021). Front-End Developer Handbook. O'Reilly Media.

- **JavaScript:**

  - jQuery Foundation. (2023). jQuery JavaScript Library v3.6.4: The Write Less, Do More, JavaScript Library. https://jquery.com/

  - Bootstrap Team. (2023). Bootstrap JavaScript Bundle v5.3.2. https://getbootstrap.com/docs/5.3/getting-started/introduction/#js

  - Popper.js Team. (2019). Popper.js v1.14.7: Positioning Engine. https://popper.js.org/

  - Flanagan, D. (2023). JavaScript: The Definitive Guide (7th ed.). O'Reilly Media.

  - Simpson, K. (2022). You Don't Know JS Yet (2nd ed.). O'Reilly Media.

  - Crockford, D. (2018). JavaScript: The Good Parts. O'Reilly Media.

  - Haverbeke, M. (2023). Eloquent JavaScript (4th ed.). No Starch Press.

  - Resig, J., Bibeault, B., & Maras, J. (2022). Secrets of the JavaScript Ninja (3rd ed.). Manning Publications.

  - Frisbie, M. (2022). Professional JavaScript for Web Developers (5th ed.). Wrox.

- MDN Web Docs. (2024). JavaScript Reference.
  https://developer.mozilla.org/en-US/docs/Web/JavaScript

- JavaScript modern tutorial. https://es.javascript.info/

- **<u>Bootstrap:</u>**

  - Bootstrap Team. (2023). Bootstrap v5.3.2: The world's most popular front-end open source toolkit. https://getbootstrap.com/

  - Spurlock, J. (2023). Bootstrap: Responsive Web Development (5th ed.). O'Reilly Media.

  - Shenoy, A., & Prabhu, A. (2022). Bootstrap 5 for Beginners: Learn By Doing with Examples. Packt Publishing.

  - Cochran, D. (2022). Twitter Bootstrap Web Development How-To (3rd ed.). Packt Publishing.

  - Libby, A. (2022). Beginning Responsive Web Design with Bootstrap 5. Apress.

  - Krause, J. (2020). Bootstrap 4 Cookbook. Packt Publishing.

  - Bootstrap Documentation. (2023). Bootstrap Grid System. https://getbootstrap.com/docs/5.3/layout/grid/

  - Bootstrap Documentation. (2023). Bootstrap Components. https://getbootstrap.com/docs/5.3/components/

  - Bootstrap Documentation. (2023). Bootstrap Utilities. https://getbootstrap.com/docs/5.3/utilities/

  - Percival, J. (2022). Building Responsive Web Applications with Bootstrap 5. Packt Publishing.

3D Graphics and Visualization

- **Three.js:**

    - Three.js Authors. (2019). Three.js v108: JavaScript 3D Library.
      https://threejs.org/

    - Three.js Authors. (2018). Three.js GLTFLoader r92: Three.js GLTF File Format
      Loader. https://threejs.org/docs/#examples/en/loaders/GLTFLoader

    - Dirksen, J. (2023). Learn Three.js: Programming 3D Animations and
      Visualizations for the Web (4th ed.). Packt Publishing.

    - Parisi, T. (2022). Programming 3D Applications with HTML5 and WebGL: 3D
      Animation and Visualization for Web Pages (3rd ed.). O'Reilly Media.

    - Cabello, R. (2022). Three.js Cookbook. Packt Publishing.

    - Three.js Documentation. (2023). Three.js Examples.
      https://threejs.org/examples/

    - Three.js Documentation. (2023). Three.js Manual. https://threejs.org/docs/

    - Janssen, A. (2022). Practical WebGL: Learn Interactive 3D Graphics with
      Three.js. Apress.

    - Takahashi, I. (2021). WebGL Insights. CRC Press.

    - Vukićević, V., & Belmonte, N. (2022). WebGL Programming Guide: Interactive
      3D Graphics Programming with WebGL (2nd ed.). Addison-Wesley
      Professional.

Backend languages

- **Python:**

    - Oracle Corporation. (2024). MySQL Connector/Python: MySQL driver written
      in Python. https://dev.mysql.com/doc/connector-python/en/

    - NumPy Developers. (2024). NumPy: The fundamental package for scientific
      computing with Python. https://numpy.org/

- Scikit-learn Developers. (2024). Machine Learning in Python. https://scikit-learn.org/

- Lutz, M. (2023). Learning Python (6th ed.). O'Reilly Media.

- Ramalho, L. (2022). Fluent Python (2nd ed.). O'Reilly Media.

- McKinney, W. (2022). Python for Data Analysis (3rd ed.). O'Reilly Media.

- Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media.

- VanderPlas, J. (2023). Python Data Science Handbook (2nd ed.). O'Reilly Media.

- Müller, A. C., & Guido, S. (2022). Introduction to Machine Learning with Python: A Guide for Data Scientists (2nd ed.). O'Reilly Media.

- Sweigart, A. (2023). Automate the Boring Stuff with Python (3rd ed.). No Starch Press.

- **Ruby:**

  - Sinatra Contributors. (2023). Sinatra: DSL for quickly creating web applications in Ruby. http://sinatrarb.com/

  - Whomwah and Contributors. (2023). RQRCode: A library for encoding QR Codes in Ruby. https://github.com/whomwah/rqrcode

  - Thomas, D., & Hunt, A. (2020). Programming Ruby 3.0: The Pragmatic Programmers' Guide (4th ed.). Pragmatic Bookshelf.

  - Carlson, L., & Richardson, L. (2022). Ruby Cookbook: Recipes for Object-Oriented Scripting (3rd ed.). O'Reilly Media.

  - Cooper, P. (2021). Beginning Ruby 3: From Beginner to Pro (4th ed.). Apress.

  - Black, D. A. (2019). The Well-Grounded Rubyist (3rd ed.). Manning Publications.

  - Olsen, R. (2018). Design Patterns in Ruby (2nd ed.). Addison-Wesley Professional.

- Tate, B. A., & Hibbs, C. (2022). Ruby on Rails: Up and Running (3rd ed.). O'Reilly Media.

- Provenzano, C. (2021). Ruby Best Practices. O'Reilly Media.

- Harris, M., & Haase, A. (2022). Ruby Under a Microscope: An Illustrated Guide to Ruby Internals (2nd ed.). No Starch Press.

<u>Database and cloud service</u>

- ## <u>MySQL</u>

  - Oracle Corporation. (2024). MySQL Database. https://www.mysql.com/

  - DuBois, P. (2022). MySQL (8th ed.). Addison-Wesley Professional.

  - Bell, C. (2023). MySQL for Developers (2nd ed.). Apress.

  - Kofler, M. (2021). The Definitive Guide to MySQL 8. Apress.

  - Schwartz, B., Zaitsev, P., & Tkachenko, V. (2022). High Performance MySQL (4th ed.). O'Reilly Media.

  - Tahaghoghi, S., & Williams, H. (2022). Learning MySQL and MariaDB (2nd ed.). O'Reilly Media.

  - MySQL Documentation. (2024). MySQL Reference Manual. https://dev.mysql.com/doc/refman/8.0/en/

  - MySQL Documentation. (2024). MySQL Performance Schema. https://dev.mysql.com/doc/refman/8.0/en/performance-schema.html

  - MySQL Documentation. (2024). MySQL Security. https://dev.mysql.com/doc/refman/8.0/en/security.html

  - Pachev, S. (2022). Understanding MySQL Internals (2nd ed.). O'Reilly Media.

- ## <u>Cloudflare:</u>

  - Cloudflare, Inc. (2024). Content Delivery Network (CDN) Services. https://www.cloudflare.com/cdn/

  - Cloudflare Documentation. (2024). Cloudflare Developer Documentation. https://developers.cloudflare.com/

- Clinton, D. G. (2022). Cloudflare for Developers. Packt Publishing.

- Cloudflare Documentation. (2024). Cloudflare Workers.
  https://developers.cloudflare.com/workers/

- Cloudflare Documentation. (2024). Cloudflare Pages.
  https://developers.cloudflare.com/pages/

- Cloudflare Documentation. (2024). Cloudflare Security.
  https://developers.cloudflare.com/security/

- Cloudflare Documentation. (2024). Cloudflare DNS.
  https://developers.cloudflare.com/dns/

- Williams, C., & Gillmor, D. (2022). Content Delivery Networks: Fundamentals,
  Design, and Evolution. Wiley.

- Cloudflare Documentation. (2024). Cloudflare DDoS Protection.
  https://developers.cloudflare.com/ddos-protection/

- Cloudflare Documentation. (2024). Cloudflare Bandwidth Alliance.
  https://www.cloudflare.com/bandwidth-alliance/

10. Anexos

- Full detailed information of the project made with "DeepWiki":

https://deepwiki.com/UnaiBenitoSanchez/TFG_UnaiBenito_FactoryManagement/1-Overview

- Installation manual:

📄 TFG_UnaiBenitoSánchez_InstallationGuide

- User manual:

📄 TFG_UnaiBenitoSánchez_UserManual