



Facultade de Informática

UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO  
GRAO EN ENXEÑARÍA INFORMÁTICA  
MENCIÓN EN ENXEÑARÍA DO SOFTWARE

# **Desarrollo de un sistema para obtener la trazabilidad comercial y gestión administrativa de una empresa**

**Estudiante:** David Pérez Piñeiro  
**Dirección:** María Martínez Pérez  
**Dirección:** José Manuel Gándara Portela

A Coruña, febreiro de 2021.



### **Agradecimientos**

Quiero agradecer a mis padres, mi hermana y mi pareja todo el apoyo que me han brindado durante la realización de este proyecto. Al igual que el ánimo que me han dado todos mis amigos.

También agradecer a los directores de este proyecto, José Gándara y María, por todas las directrices indicadas para lograr llegar a este punto.



## **Resumen**

El presente TFG surge de la necesidad de complementar un ERP personalizado con la integración de los módulos de gestión comercial y administrativa. Una empresa, dedicada a la comercialización de piedra al por mayor, ya cuenta con un ERP propio, pero desea ampliar sus funcionalidades para poder tener un mayor control y trazabilidad de sus productos. Además, quiere mejorar la toma de decisiones con información detallada de sus ventas.

El objetivo, es que un comercial de la empresa, pueda disponer; de la realización de pedidos desde cualquier lugar, así como de la información necesaria de los clientes y ventas, y todo ello en tiempo real. También, que el departamento de administración tenga una parte de consulta y definición de KPIs. Para esto se crea una aplicación web con todas estas funcionalidades.

Estos sistemas facilitarán el trabajo a los usuarios de la aplicación web, pudiendo definir mejor las estrategias de venta y la toma de decisiones.

## **Abstract**

This final project arises out of the need to complement a personalized ERP with the integration of commercial and administrative management modules. A stone wholesale company already has its own ERP, but wants to expand its functionalities in order to have greater control and traceability of its products. In addition, it wants to improve the decision-making with detailed information on its sales.

The objective is that any commercial employee of the company, can have the orders that were made from anywhere, as well as the necessary information on customers and sales, and all this in real time. Also, that the administration department can take part in the consultation and definition of KPIs. For this, a web application is created with all these functionalities.

These systems will facilitate the work of the web application users, being able to better define sales strategies and decision-making.

### **Palabras clave:**

- ERP
- KPI
- MVC
- Trazabilidad
- Dashboard
- Cuadro de mando



# Índice general

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Contextualización . . . . .	1
1.2	Alcance y objetivos . . . . .	2
1.3	Estructura de la memoria . . . . .	3
<b>2</b>	<b>Estado del arte</b>	<b>7</b>
2.1	Sage Business Cloud . . . . .	7
2.2	Microsoft Dynamics 365 . . . . .	8
2.3	Odoo . . . . .	8
2.4	Conclusiones . . . . .	9
<b>3</b>	<b>Fundamentos tecnológicos</b>	<b>11</b>
3.1	Sistema operativo . . . . .	11
3.2	IDE . . . . .	11
3.3	Control de versiones . . . . .	12
3.4	TortoiseGit . . . . .	12
3.5	Lenguajes de programación . . . . .	12
3.6	Base de datos . . . . .	13
3.7	Draw.io . . . . .	13
<b>4</b>	<b>Metodología</b>	<b>15</b>
4.1	Análisis . . . . .	15
4.2	Diseño . . . . .	16
4.2.1	Patrón Model-View-Controller (MVC) . . . . .	16
4.2.2	Patrón Value Object (VO) . . . . .	17
4.2.3	Patrón Data Access Object (DAO) . . . . .	18
4.3	Implementación . . . . .	18
4.4	Pruebas . . . . .	18

---

<b>5</b>	<b>Planificación y costes</b>	<b>21</b>
5.1	Planificación . . . . .	21
5.2	Estimación de costes: . . . . .	22
5.2.1	Recursos humanos . . . . .	22
5.2.2	Recursos materiales . . . . .	23
<b>6</b>	<b>Análisis de requisitos</b>	<b>25</b>
6.1	Fase previa . . . . .	25
6.1.1	Necesidades . . . . .	25
6.1.2	Usuarios . . . . .	26
6.2	Primera iteración . . . . .	27
6.3	Segunda iteración . . . . .	27
6.4	Tercera iteración . . . . .	27
6.5	Cuarta iteración . . . . .	28
6.6	Conclusiones . . . . .	28
<b>7</b>	<b>Modelado E/R</b>	<b>31</b>
<b>8</b>	<b>Diseño</b>	<b>37</b>
8.1	Primera iteración . . . . .	37
8.2	Segunda iteración . . . . .	37
8.3	Tercera iteración . . . . .	38
8.4	Cuarta iteración . . . . .	38
<b>9</b>	<b>Implementación</b>	<b>45</b>
9.1	Primera iteración . . . . .	45
9.2	Segunda iteración . . . . .	48
9.3	Tercera iteración . . . . .	49
9.4	Cuarta iteración . . . . .	51
<b>10</b>	<b>Pruebas</b>	<b>59</b>
10.1	Pruebas unitarias . . . . .	59
10.2	Pruebas de integración . . . . .	59
10.3	Pruebas de sistema . . . . .	60
10.4	Pruebas de aceptación . . . . .	60
<b>11</b>	<b>Conclusiones</b>	<b>61</b>
11.1	Conclusiones . . . . .	61
11.2	Lecciones aprendidas . . . . .	62



ÍNDICE GENERAL

---

11.3 Líneas futuras . . . . .	62
<b>Lista de acrónimos</b>	<b>63</b>
<b>Bibliografía</b>	<b>65</b>



# Índice de figuras

---

3.1	Logo Windows 10. . . . .	11
3.2	Logo Apache Netbeans. . . . .	11
3.3	Logo Git. . . . .	12
3.4	Logo TortoiseGit. . . . .	12
3.5	Logo MySQL. . . . .	13
3.6	Logo Draw IO. . . . .	13
4.1	Representación del modelo incremental. . . . .	16
4.2	Representación del patrón MVC. . . . .	17
5.1	Diagrama de Gantt del proyecto. . . . .	22
5.2	Estimación coste humano. . . . .	23
5.3	Estimación coste material. . . . .	23
6.1	Casos de uso de la aplicación web. . . . .	29
7.1	Diagrama E/R de la parte de facturación y reservas. . . . .	35
7.2	Diagrama E/R de artículos. . . . .	36
8.1	Diagrama de clases del Login. . . . .	39
8.2	Diagrama de clases de Visor. . . . .	39
8.3	Diagrama de clases de Clientes. . . . .	40
8.4	Diagrama de clases de Ventas. . . . .	40
8.5	Diagrama de clases de Carrito. . . . .	41
8.6	Diagrama de clases de Definición KPIs. . . . .	42
8.7	Diagrama de clases del Cuadro de Mando. . . . .	43
9.1	Captura de los archivos generados en la primera iteración. . . . .	47
9.2	Captura de los archivos generados hasta la segunda iteración. . . . .	49

9.3	Captura de los archivos generados hasta la tercera iteración. . . . .	51
9.4	Captura de los archivos generados hasta la cuarta iteración. . . . .	53
9.5	Vista de la autenticación de usuario. . . . .	54
9.6	Vista del visor de productos de la empresa. . . . .	54
9.7	Vista de la lista de clientes. . . . .	55
9.8	Vista de la lista de ventas de un cliente. . . . .	55
9.9	Vista donde se añaden los productos al carrito. . . . .	56
9.10	Vista del carrito de un cliente. . . . .	56
9.11	Vista de la situación de los pedidos. . . . .	57
9.12	Vista donde se definen los KPIs. . . . .	57
9.13	Vista del Cuadro de Mando. . . . .	58

# Introducción

---

## 1.1 Contextualización

El mercado global, y en especial el entorno empresarial, está evolucionando de forma muy considerable en los últimos años, lo que obliga a las empresas a ser cada vez más competitivas. Estos cambios suelen relacionarse con la creciente globalización de la economía, la internacionalización de los mercados y la rapidez del cambio tecnológico.

Hoy en día, los sistemas de información y la tecnología, en particular los Enterprise Resource Planning (ERP), ocupan un papel imprescindible en el éxito global de la competencia para las grandes, medianas y pequeñas empresas. Por ello, las compañías se han visto obligadas a tener que adaptar sus sistemas de información para poder sobrevivir en el mercado, mantener su posición e incluso para diferenciarse de la competencia. Las empresas han ido incorporando las nuevas tecnologías de la información y de las telecomunicaciones, convirtiéndose estas en activos muy importantes para las mismas. Toda empresa que desee competir en el mercado actual debe considerar la “información” como una herramienta clave. Por esta razón, es necesario que las empresas dispongan de los sistemas de información adecuados para suministrar rápida y eficientemente esa información.

También es importante conseguir una adecuada adaptación y predicción de cambios. Esta se consigue gracias a la correcta integración de todos los departamentos. La organización que lleva a cabo la empresa, tiene como último fin garantizar la creación de valor de cara al cliente a través de una buena relación e interacción continua con los mismos. Si las empresas determinan de forma efectiva cómo relacionarse con sus clientes, estos asegurarán una mayor realización de ventas efectivas y por tanto una mayor rentabilidad para seguir creciendo. Por tanto, es importante establecer una buena estrategia de gestión comercial que puede asegurar el éxito de la empresa y de su política comercial. [1]

El presente Trabajo de Fin de Grado (TFG) se enmarca en una empresa gallega que se dedica a la comercialización de piedra al por mayor, en distintos formatos, pero especialmen-

te en formato de tabla serrada. La empresa ya se decantó en su momento por integrar un ERP personalizado. Eran conscientes de las ventajas que esto suponía para la empresa ya que cuenta con varios años de experiencia en un sector concreto y con unas características muy marcadas.

Este ERP propietario ya dispone de los siguientes módulos: gestión de compras, ventas, facturación, almacenes y logística. Con esto abarcan todo el proceso de compra y de venta, desde la solicitud de presupuestos hasta la facturación y gestión de pagos de facturas. Además, tiene un sistema de control de almacén que sirve para tener una trazabilidad total del producto, desde que entra hasta que sale de la empresa.

La idea de este TFG surge debido a la importancia que adquiere el departamento comercial dentro de la empresa y la necesidad de obtener un análisis de su actividad. El objetivo es poder mejorar este proceso de cara al futuro. El departamento comercial es el encargado de conseguir los pedidos de todos los clientes. Necesitan incorporar una nueva funcionalidad en el ERP para poder gestionar mejor las ventas de sus clientes. Además, el departamento de administración de la empresa, quiere optimizar mejor su trabajo. Para conseguir este objetivo necesita incorporar una parte de consulta y definición de Key Performance Indicator (KPI).

## 1.2 Alcance y objetivos

El objetivo principal del proyecto es incorporar al ERP actual los módulos de gestión comercial y gestión administrativa de la empresa mediante una aplicación web.

Para el módulo de gestión comercial, se busca que el comercial tenga un control total sobre los datos de sus clientes. Para ello, van a tener accesos entre otros datos, a toda la información de sus últimas compras, sus pagos pendientes, si está asegurado o el crédito disponible. Gracias a disponer de todos estos datos, la gestión para sus ventas es mucho más fácil. Además, un comercial podrá reservar directamente al almacén un producto concreto utilizando la aplicación web. Esto se consigue reservando la serie de un producto, que supondrá una modificación del ERP actual en la parte del sistema de venta.

Para el módulo de gestión administrativa, el objetivo es adquirir información interesante sobre la propia empresa para así mejorar el rendimiento de la misma. Esta información se consigue mediante unas métricas KPI que se utilizan para sintetizar la información sobre la eficacia y productividad de las acciones que se llevan a cabo en la empresa, con el fin de poder tomar decisiones y determinar aquellas que han sido más efectivas a la hora de cumplir con los objetivos. [2]

Por otro lado, los usuarios del ERP son todos los miembros de la empresa. Por tanto, cada miembro tendrá acceso a los datos e información que sean necesarios para su rol.

Los objetivos más concretos a llevar a cabo en este proyecto son los que se detallan a

continuación:

- Controlar de forma personalizada los datos a visualizar por parte de los distintos comerciales de la empresa.
- Controlar a los clientes con acceso a su historial de compras y situación financiera.
- Obtener una trazabilidad total de los productos de la empresa.
- Gestionar los pedidos mediante la aplicación Web de manera inmediata.
- Implementar de un Dashboard para la visualización de métricas empresariales y el seguimiento de KPIs.

### 1.3 Estructura de la memoria

La disposición de los capítulos en la memoria es la siguiente:

1. **Introducción.** En este capítulo se refleja una visión global del contexto del proyecto junto con los principales objetivos y el alcance de éste. Además, se muestra una estructura de la presente memoria. Los apartados de este capítulo son:
  - 1.1. Contextualización.
  - 1.2. Alcance y objetivos.
  - 1.3. Estructura de la memoria.
2. **Estado del arte.** En este capítulo se comentan otras herramientas disponibles en el mercado que podrían ser utilizadas para elaborar las funcionalidades de este proyecto. Además, se termina con una conclusión explicando la herramienta que ya está definida en este proyecto. Los apartados de este capítulo son:
  - 2.1. Sage Business Cloud.
  - 2.2. Microsoft Dynamics 365.
  - 2.3. Odoo.
  - 2.4. Conclusiones.
3. **Fundamentos tecnológicos.** Se expone lo utilizado en el proyecto, como sistema operativo, entorno y lenguaje de desarrollo, sistema de gestión de base de datos, etc. Se justifica el motivo de su elección frente al de otras alternativas.
4. **Metodología.** En este capítulo se explica la metodología seleccionada para este proyecto, detallando las distintas fases que tiene.

- 4.1. Análisis.
  - 4.2. Diseño.
  - 4.3. Implementación.
  - 4.4. Pruebas.
5. **Planificación y costes.** Se plasman las diferentes fases de desarrollo del proyecto. Explicando la planificación con su diagrama de Gantt y la estimación de costes.
- 5.1. Planificación.
  - 5.2. Estimación de costes.
6. **Análisis de requisitos.** Se explican las necesidades del proyecto y los usuarios de la aplicación. Además, de los requisitos para las distintas iteraciones del proyecto. Va a tener los siguientes apartados:
- 6.1. Fase previa
  - 6.2. Primera iteración.
  - 6.3. Segunda iteración.
  - 6.4. Tercera iteración.
  - 6.5. Cuarta iteración.
7. **Modelado E/R.** En este capítulo se explica el modelo Entidad-Relación que enmarca a toda la aplicación.
8. **Diseño.** En este capítulo se detallan los diseños elaborados para las distintas iteraciones del proyecto. Va a tener los siguientes apartados:
- 8.1. Primera iteración.
  - 8.2. Segunda iteración.
  - 8.3. Tercera iteración.
  - 8.4. Cuarta iteración.
9. **Implementación.** Para este capítulo se muestra toda la implementación que fue necesaria para crear la aplicación web. Se divide en los siguientes apartados:
- 9.1. Primera iteración.
  - 9.2. Segunda iteración.
  - 9.3. Tercera iteración.
  - 9.4. Cuarta iteración.



10. **Pruebas.** En este capítulo se reflejan las distintas técnicas de pruebas realizadas para validar la efectividad del proyecto. Se divide en cuatro tipos de pruebas:
  - 10.1. Pruebas unitarias.
  - 10.2. Segunda de integración.
  - 10.3. Tercera de sistema.
  - 10.4. Cuarta de aceptación.
  
11. **Conclusiones y lecciones aprendidas.** Se enumera todo lo realizado en el proyecto. También se explican las lecciones aprendidas.
  - 11.1. Conclusiones.
  - 11.2. Lecciones aprendidas.
  - 11.3. Líneas Futuras.



# Estado del arte

---

En este apartado se muestran una serie de herramientas que proporcionan las funcionalidades necesarias para llevar a cabo este proyecto en la actualidad. En este caso, el desarrollo de la aplicación ya está decidido. Se va a continuar ampliando las funcionalidades de un ERP personalizado de la empresa. De todas formas, también se muestran distintas alternativas que se podrían haber escogido.

## 2.1 Sage Business Cloud

Sage 200 [3] es una solución de gestión integral con un software estándar, modular y adaptable a todas las necesidades de gestión. Es capaz de gestionar en cualquier momento las cuentas y clientes, además de la fabricación, cadena de suministro, inteligencia empresarial, entre otras funciones. Con ella se toma el control de todos los procesos empresariales de principio a fin.

Entre todas las funcionalidades de las que dispone Sage, las más destacables con respecto a las necesidades de este proyecto son:

- Control de finanzas: permite tener un control completo del plan contable y trabajar en todos los aspectos financieros de la empresa.
- Control de almacenes: se basa en tener un mayor control de los inventarios gracias a funciones como la trazabilidad completa de la información, stock real y virtual, rentabilidad y evolución de los artículos.
- Gestión comercial: realiza una gestión completa del ciclo de compras y ventas, desde el aprovisionamiento hasta la gestión de contratos, expedientes y almacenes.
- CRM: solución integral para identificar, captar y fidelizar a los clientes, además de poder conocer su historial completo para saber lo que puede necesitar en cualquier momento.

- Soluciones conectadas: añade soluciones de movilidad para gestionar el catálogo de productos y los pedidos de clientes online e integrar las facturas recibidas automáticamente.

## 2.2 Microsoft Dynamics 365

Microsoft Dynamics 365 [4] es una aplicación que se integra a la perfección para ofrecer una solución completa con la que se puede gestionar toda la empresa, se incluyen las ventas, el marketing, los servicios, las operaciones y el comercio. Se consigue una perspectiva integral de la organización, lo que proporciona conocimientos en tiempo real y mejora los resultados.

De todas las funcionalidades que ofrece, las más interesantes para el proyecto son:

- Gestión de finanzas: permite conectar los datos de contabilidad, ventas, compras, inventario e interacciones con los clientes para obtener una visión integral del negocio.
- Mejor servicio al cliente: asigna prioridades a los clientes potenciales según su potencial de ingresos. Hace un seguimiento de todas las interacciones con el cliente.
- Disponibilidad: funciona desde cualquier dispositivo de forma fluida y segura.

## 2.3 Odoo

Odoo [5] es un conjunto de aplicaciones que ayudan a abordar todas las necesidades comerciales en una solución única, rentable y modular. Las aplicaciones de Odoo están perfectamente integradas entre sí, lo que permite automatizar completamente los procesos comerciales y conseguir ahorros y beneficios en la empresa.

Con Odoo se pueden conseguir cumplir los objetivos con las siguientes funcionalidades:

- Análisis de la empresa: permite personalizar los dashboards y obtener una vista completa del negocio. También permite profundizar con informes en tiempo real y diagramas de flujo.
- Gestión de clientes: conseguir el historial completo de las actividades relacionadas con cada cliente (oportunidades, pedidos, pagos, etc.).
- Gestión de ventas: permite facturar según las cantidades entregadas o solicitadas, en función del tiempo y los materiales; gestionar los términos de pago por cliente o por factura y realizar el seguimiento de los vencimientos de las facturas fácilmente.
- Gestión del inventario: permite escanear los productos del almacén al utilizar el escáner de código de barras; controlar los paneles y realizar el seguimiento y se puede gestionar

fácilmente un inventario único o un entorno complejo de múltiples almacenes al activar las características por encargo.

## **2.4 Conclusiones**

Como conclusión, se puede destacar que, con cualquiera de estas herramientas se puede llegar a implementar los objetivos fijados para este TFG, sin embargo, con ninguno se puede conseguir una aplicación tan personalizada y concreta de la empresa como con el ERP personalizado. Además, este proyecto ya parte de la base de un ERP personalizado, que fue escogido en su momento por las características tan marcadas que tiene la empresa y los años de experiencia de los que dispone. Destacar también que algunas de las funcionalidades que actualmente trabaja la empresa no se pueden implementar en las herramientas anteriormente mencionadas.



# Fundamentos tecnológicos

---

En este capítulo se recogen las diferentes herramientas y tecnologías empleadas para la elaboración del proyecto. Justificando el motivo de su elección frente al de otras alternativas.

## 3.1 Sistema operativo

Se ha utilizado Windows 10 como sistema operativo [6]. Este es el instalado en el ordenador donde se desarrolla la aplicación del proyecto. Ha sido escogido por encima de otros debido al nivel de conocimiento y agilidad que se tiene sobre el mismo.



Figura 3.1: Logo Windows 10.

## 3.2 IDE

Para el desarrollo de la aplicación web se ha utilizado Apache Netbeans 12.0 como entorno de desarrollo integrado [7]. Netbeans permite el uso de un amplio rango de tecnologías de desarrollo para aplicaciones web. Da soporte a lenguajes de todo tipo y puede instalarse en distintos sistemas operativos.

Se ha escogido por el largo periodo de uso con este entorno de desarrollo.



Figura 3.2: Logo Apache Netbeans.

### 3.3 Control de versiones

Un sistema de control de versiones permite registrar todos los cambios por los que van pasando nuestros archivos, de modo que podamos consultar estos cambios más adelante. También permite colaborar con otras personas, trabajar en paralelo en diferentes características sin molestar, resolver conflictos, tener copias de seguridad de los proyectos sin riesgo de perderlos y muchas otras cosas [8].

Para el desarrollo de software es necesario contar con un sistema de control de versiones y este es el más utilizado hoy en día. Por eso, se ha escogido Git por encima de otras opciones.



Figura 3.3: Logo Git.

### 3.4 TortoiseGit

TortoiseGit es un cliente de control de revisión Git, implementado como una extensión de shell de Windows y basado en TortoiseSVN. Es un software gratuito lanzado bajo la Licencia Pública General de GNU [9]. Se encarga de gestionar elementos almacenados en un repositorio local a lo largo del tiempo.

Se ha optado por utilizar esta herramienta para facilitar la gestión con Git.



Figura 3.4: Logo TortoiseGit.

### 3.5 Lenguajes de programación

Como lenguajes de programación para el desarrollo del proyecto, se ha decidido por usar:

- **PHP:** lenguaje de código abierto muy popular, adecuado para el desarrollo web y que puede ser incrustado en HTML.
- **HTML:** lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto.



- **JavaScript:** lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes, etc.
- **AJAX:** conjunto de técnicas de desarrollo web que permiten que las aplicaciones web funcionen de forma asíncrona, procesando cualquier solicitud al servidor en segundo plano.

Han sido escogidos estos lenguajes por la experiencia y agilidad que se tienen sobre ellos. Son lenguajes muy utilizados por la comunidad para el desarrollo web, por tanto, son muy cómodos para poder buscar cualquier tipo de información sobre ellos.

### 3.6 Base de datos

Para la gestión de bases de datos relacional se ha optado por utilizar MySQL. Está desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation, y está considerada como la base de datos de código abierto más popular del mundo. Utiliza SQL como lenguaje de programación diseñado para actualizar, obtener, y calcular información en bases de datos relacionales [10].

El uso de bases de datos es necesario para guardar los datos de toda la aplicación. El ERP personalizado del que dispone la empresa, ya utilizaba MySQL como gestor de bases de datos, por lo que su elección ha sido sencilla.



Figura 3.5: Logo MySql.

### 3.7 Draw.io

Es una aplicación que permite crear todo tipo de diagramas. Se ha utilizado para elaborar los diagramas de clases, el diagrama de casos de uso y el diagrama de Gantt de este proyecto. Ha sido escogido por ser muy intuitivo y gratuito.



Figura 3.6: Logo Draw IO.



# Metodología

---

La metodología del desarrollo software es el conjunto ordenado y bien estructurado de procesos que aseguran, a los desarrolladores, la realización oportuna, en tiempo y forma, de un software de calidad adecuado a las necesidades. Como objetivos se pueden destacar los siguientes:

- Disponer de un método sistematizado para abordar el proceso de desarrollo
- Asegurar el cumplimiento de plazos y costes
- Documentar adecuadamente el sistema

A pesar de que, las metodologías de desarrollo de software no varían en lo esencial, existen varios modelos o marcos de trabajo distintos. Estos métodos de trabajo han sido creados para satisfacer necesidades específicas en los proyectos. Los más destacados en la actualidad son: el modelo en cascada, el prototipado, el desarrollo iterativo e incremental y la metodología Scrum, entre otros [11].

En este proyecto se ha seguido un enfoque basado en incrementos (Figura 4.1 de la página 16), de manera que en cada iteración se obtiene una versión funcional del producto y se realizan secuencias basadas en el modelo en cascada. Estas secuencias se basan en realizar cuatro fases distintas por cada iteración: análisis, diseño, implementación y pruebas. En los siguientes apartados de este capítulo se comentan brevemente cada una de estas fases.

## 4.1 Análisis

La principal tarea a llevar a cabo en esta fase es la obtención de los requisitos, sobre todo funcionales, que debe cumplir la aplicación a desarrollar. En este proyecto se han realizado distintas entrevistas con el cliente y jefe de proyecto para adquirir todos los requisitos. En el capítulo 6 se detalla lo que se realiza en esta fase para las distintas iteraciones de la aplicación web.

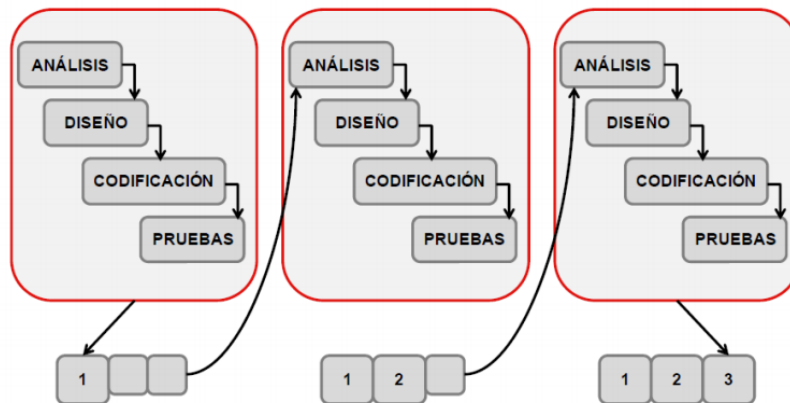


Figura 4.1: Representación del modelo incremental.

## 4.2 Diseño

En esta fase se busca una solución conceptual que satisfaga los requisitos obtenidos en la fase de análisis.

La documentación en esta fase se realiza siguiendo la notación estándar definida por el Lenguaje Unificado de Modelado (UML) para mostrar el diseño de la aplicación. Se obtendrán diversos diagramas de clases del diseño que representan tanto las clases que componen la aplicación como sus relaciones. En el capítulo 8 se muestran los distintos diagramas que se realizan en esta fase para las distintas iteraciones de la aplicación web.

Para el diseño conceptual de las bases de datos se emplean diversos diagramas Entidad-Relación (E/R), que se detallan en el capítulo 7.

Además, se utilizan patrones de diseño para solucionar los distintos problemas que se pueden presentar durante el diseño de la aplicación. En los siguientes apartados se explican algunos de los patrones utilizados.

### 4.2.1 Patrón Model-View-Controller (MVC)

Para el diseño de esta aplicación se ha seguido el patrón MVC que consta de tres componentes: modelo, vista y controlador.

Este patrón proporciona una separación clara entre el modelo (lógica de negocio), y la vista (interfaz gráfica), gracias a un controlador que los mantiene desacoplados. El funcionamiento básico de este patrón se puede apreciar en la figura 4.2 (página 17).

A continuación se explica brevemente cada uno de los componentes.

- **Modelo:** es el que define los datos que debe contener la aplicación. Se encarga de insertar, actualizar, eliminar o consultar cualquier dato a través de la base de datos.

- **Vista:** sirve como representación visual del modelo. Se encarga de mostrar los datos de la aplicación.
- **Controlador:** recibe las peticiones del usuario, y se encarga de solicitar los datos al modelo y comunicárselos a la vista para que los represente visualmente.

En cuanto a las principales ventajas que ofrece el patrón MVC se destacan las siguientes:

- Separación clara de donde tiene que ir cada tipo de lógica, facilitando el mantenimiento y la escalabilidad de la aplicación.
- Sencillez para crear distintas representaciones de los mismos datos.
- Reutilización de los componentes.

Al existir una clara separación entre las capas, es posible tener dos grupos distintos de programadores, uno para la capa de negocio y otro para la capa de presentación. De esta manera se puede conseguir un desarrollo rápido de la aplicación.

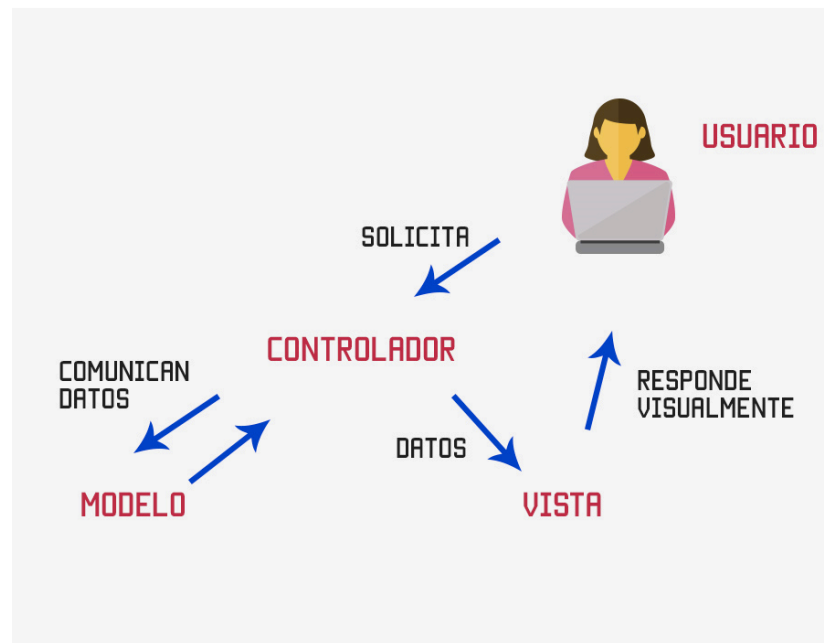


Figura 4.2: Representación del patrón MVC.

#### 4.2.2 Patrón Value Object (VO)

Con este patrón se pretende agrupar en una clase un conjunto de atributos procedentes de uno o varios objetos de dominio. El uso de este patrón facilita el intercambio de datos entre las distintas capas de la aplicación, reduciendo el número de peticiones de una capa a otra.

### 4.2.3 Patrón Data Access Object (DAO)

La idea de este patrón es desacoplar la lógica de negocio de la lógica de acceso a datos, de manera que se pueda cambiar fácilmente la fuente de datos de la aplicación. Además con el uso del patrón DAO, se agrupa toda la lógica de acceso a datos de una entidad del dominio dentro de una clase. Esto facilita su creación, depuración y mantenimiento.

## 4.3 Implementación

En esta fase se pasa del diseño de la aplicación que se realizó en la anterior fase al lenguaje de programación para crear la aplicación. Dependiendo del tipo de aplicación se emplean distintos lenguajes y herramientas. Para la aplicación de este proyecto se ha utilizado:

- Mysql para la gestión de bases de datos con el lenguaje SQL.
- PHP, JavaScript, HTML y AJAX para el desarrollo de la aplicación web.
- Git como sistema de control de versiones.
- Netbeans como entorno de desarrollo.
- TortoiseGit como cliente para Git open source.
- Draw.io para realizar los diferentes diagramas.
- Toad for MySQL para la administración de las bases de datos en MySQL.

En el capítulo 9 se detalla lo que se realiza en esta fase para las distintas iteraciones de la aplicación web.

## 4.4 Pruebas

En esta fase se elaboran diferentes pruebas con el fin de reducir el número de errores de la aplicación. Se realizan para comprobar que funciona correctamente el programa y que cumple los requisitos fijados en la fase de análisis.

Las pruebas que se realizan son de distinto tipo según el grado de desarrollo del sistema. Se llevan a cabo pruebas de unidad, de integración, de aceptación y de sistema.

Con las pruebas unitarias se comprueba el correcto funcionamiento de un módulo de código. Sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Estas pruebas se realizan para comprobar las distintas funciones empleadas en la capa modelo de la aplicación. Son las primeras pruebas que se hacen.

Las pruebas de integración sirven para verificar que todos los subsistemas encajan correctamente y funcionan juntos. Se deben contemplar todas las actividades de la aplicación y todas las acciones posibles. Para ello se comprueban que todas las llamadas entre los controladores y vistas funcionan correctamente, al igual que el intercambio de datos entre el modelo y el controlador.

Mediante las pruebas de aceptación y sistema se verifica que las funcionalidades del sistema de cara a los usuarios finales son las adecuadas y que el sistema alcanza un nivel adecuado de usabilidad.

Llevando a cabo todo este conjunto de pruebas se permite reducir considerablemente la existencia de errores del sistema y, por lo tanto se aumenta la fiabilidad del sistema, sin embargo en ningún caso se garantiza la inexistencia de los mismos.





# Planificación y costes

---

El presente capítulo recoge la planificación del proyecto y su estimación de costes. Se consideran las distintas etapas que debe presentar el proyecto para llevarse a cabo y el cálculo aproximado de su precio.

## 5.1 Planificación

La planificación de un proyecto es la ordenación sistemática de las tareas para lograr un objetivo, donde se expone lo que se necesita hacer y cómo debe llevarse a cabo [12].

El objetivo de este proyecto es realizar una aplicación web con las funcionalidades que se describen más adelante. Basándose en proyectos de una envergadura similar, se ha estimado una duración de 6 semanas.

Este TFG ha comenzado el 29 de septiembre de 2020, y tras dedicarle 40 horas semanales (8 diarias) se ha finalizado el día 6 de noviembre de 2020. Además, el jefe de proyecto le ha dedicado unas 4,5 horas de media a la semana.

El proyecto se divide en seis fases, que se explica el contenido de cada uno de ellas a continuación junto con un diagrama de Gantt (figura 5.1).

- **Fase previa:** se realiza la descripción del proyecto, la obtención de requisitos y la familiarización del entorno de la empresa. En ella se diseña el calendario que organiza el trabajo a realizar por las siguientes fases. Tras la obtención de requisitos, se ha acordado hacer cuatro iteraciones. Cada una con las distintas fases que se han descrito en el capítulo 4, donde se explica la metodología a seguir.
- **Iteraciones:** se realizan cuatro iteraciones distintas. En todas ellas se empieza con una reunión con el cliente y jefe de proyecto para profundizar sobre las distintas funcionalidades. Además, se muestran los resultados de la iteración anterior para cerciorarse de su buen funcionamiento y sugerir mejoras. En ellas se termina creando la aplicación



- Jefe de proyecto: 11,53 €/hora.

Quedando la estimación de costes de los recursos humanos de la siguiente manera:

Perfil Prof.	Horas	Coste
Analista programador	232	2.403,52 €
Jefe de proyecto	27	331,31 €
		2.734,83 €

Figura 5.2: Estimación coste humano.

### 5.2.2 Recursos materiales

Los recursos materiales empleados fueron los siguientes:

- Se ha utilizado un ordenador portátil con procesador Intel Core i5 y 8GB de RAM para el desarrollo del software.
- Se han incorporado tres Ipad Wifi + Cellular para los comerciales, donde disponen de conexión a la aplicación web.

Quedando la estimación de costes de los recursos materiales de la siguiente manera:

Producto	Cantidad	Coste
Portátil	1	800,00 €
Ipad Wifi + Cellular	3	1.557,00 €
		2.357,00 €

Figura 5.3: Estimación coste material.

Por tanto, el coste total del proyecto es aproximadamente **5.091,83 €**.



# Análisis de requisitos

---

En la fase de análisis, la principal tarea a llevar a cabo es la obtención de los requisitos, sobre todo funcionales, que debe cumplir la aplicación a desarrollar. En la ingeniería de sistemas, un requisito es un dato necesario en la fase de diseño de un producto. En este caso, los requisitos dictan la norma sobre lo que debe ejecutar el producto a crear [14].

Para este TFG se empieza con una fase previa, donde se realizan las primeras reuniones con el cliente y jefe de proyecto, dando así a conocer todos los requisitos del proyecto.

Más adelante, en este capítulo, vamos a ir detallando las distintas funcionalidades que se fueron completando por cada iteración. Al principio, de cada una de estas, se han realizado reuniones con el cliente y jefe de proyecto, para comprobar el trabajo ya realizado y analizar lo que queda por hacer.

## 6.1 Fase previa

A continuación, se muestran las necesidades que solicita el cliente para este proyecto y los usuarios que van a manejar la aplicación.

### 6.1.1 Necesidades

La aplicación web cubre los procesos necesarios para que un comercial de la empresa pueda disponer de: la realización de pedidos desde cualquier lugar y la información necesaria de los clientes y ventas. También dispone de una parte de consulta y definición de KPIs por parte del departamento de administración.

- **Información del cliente:** se obtiene información relevante sobre un comprador. Un comercial, antes de realizar un pedido, debe estar informado de la situación económica del cliente. Para ello, el comercial necesita conocer si un cliente dispone de un seguro que le cubra los gastos en caso de impago, el importe total que le asegura, el plazo desde

que se genera la factura hasta que se pueda reclamar ese dinero, lo que tiene pendiente de cobrar de la empresa y el margen del que dispone para seguir comprando.

- **Información de ventas:** se extrae la información de las ventas de un cliente. Para un comercial, es interesante conocer las ventas que ya le ha realizado a un cliente para poder preparar una buena estrategia de venta.
- **Gestión de pedidos:** se generan los pedidos de un cliente. Un comercial, requiere el catálogo de la empresa de una forma intuitiva y cómoda para mostrarle a un cliente. Además, necesita que sea útil para añadir distintos artículos a un pedido. La empresa elabora productos con características únicas. Si alguno de estos se quiere reservar para un cliente, el comercial precisa poder bloquearlo para que no pueda ser vendido.
- **Situación de pedidos:** se muestra la situación de los pedidos. Todos los comerciales de la empresa, necesitan saber en todo momento cual es la situación en la que se encuentran los pedidos de sus clientes.
- **Definición de KPIs:** se configuran los distintos KPIs acompañados de una representación gráfica. El departamento administrativo de la empresa, precisa obtener información relevante de su situación económica. Esto se consigue con la creación de dashboard para reflejar, mediante una representación gráfica, las principales métricas o KPIs que intervienen en la consecución de los objetivos.

### 6.1.2 Usuarios

Existen dos tipos de usuarios que van utilizar la aplicación web.

- **Comercial:** se ocupa de hacer llegar los bienes y servicios producidos por la empresa hasta el consumidor, intentando satisfacer sus necesidades. Con la creación de los distintos requisitos del proyecto, podrá disponer de una mayor cantidad de información que le ayudará para convencer a los clientes.
- **Administrador:** sus funciones principales son organizar, dirigir, coordinar, controlar y evaluar la situación de la empresa. Podrá mejorar gran parte de sus funciones con la configuración de KPIs y su representación gráfica.

El sistema va a estar configurado para que cada usuario únicamente pueda realizar las funciones asociadas a su trabajo diario.

## 6.2 Primera iteración

En la primera iteración se constituye el primer entregable de nuestra aplicación web. En ésta, se añaden las funcionalidades del login de usuario y la visualización de todos los productos clasificados por los diferentes tipos y categorías que tienen.

Este programa no dispone del registro de usuario, debido a que solo van a acceder a esta aplicación los comerciales y los administradores de la empresa. Estos ya están dados de alta con su login y contraseña a través del ERP personalizado.

El sistema de autenticación de usuarios, permite que un comercial o jefe de la empresa se identifique, para poder acceder a las distintas funcionalidades de las que va a disponer la aplicación. Dependiendo del rol que tenga el usuario va a disponer de una información más o menos restringida. Si un usuario no está logueado, no va a poder realizar ninguna operación, ni va a poder visualizar ningún tipo de información.

La vista de todo el catálogo de la empresa, sirve para que un comercial que se encuentre con un cliente, le vaya mostrando los distintos productos de la empresa categorizados por sus distintos tipos de mineral, colección, nombre de familia, calidad, formato, acabado o espesor.

## 6.3 Segunda iteración

Para la segunda iteración, se añaden los requisitos necesarios para la consulta de datos de los clientes.

En primer lugar, se agrega la visualización de la lista de compradores. Este listado va a ser distinto dependiendo del rol del usuario. Si se trata de un comercial, solamente aparecen los clientes que tiene a su cargo, sin saber el resto que hay en la empresa. En cambio, si se trata de un administrador, si que va a tener la información de todos.

En segundo lugar, se incorpora la vista con la información detallada de un cliente. Para ello, se mostrará los datos de interés para un comercial de un comprador concreto, como puede ser su límite de crédito, si está asegurado, el riesgo vivo, etc.

Por último, se añade la lista de ventas para un cliente en concreto. Esto también es información interesante para un comercial de la empresa, ya que dispone de los detalles de las últimas ventas de un comprador.

## 6.4 Tercera iteración

En esta iteración se desarrolla todo lo relacionado con la creación de un pedido. Para ello se añaden las funcionalidades de: incluir productos en un carrito, visualizar carrito, modificar carrito, confirmar la venta de esos productos y visualizar la situación de los pedidos.

Un comercial, cuando gestiona una venta, va incluyendo productos que le interesan al cliente en su carrito. Una vez añadidos todos los artículos deseados, se revisan esos datos para modificar o eliminar alguno de ellos, y se confirma ese pedido para ser tramitado.

Algunos productos tienen unas características registradas que los hacen únicos. Estos productos disponen de un número de serie concreto que los identifica. Cuando un cliente desea una serie concreta de un producto, esta va a ser bloqueada por la empresa para que no se realice su venta a otro cliente.

Además, un comercial puede consultar cual es la situación actual de los pedidos que ha realizado para todos sus clientes.

## 6.5 Cuarta iteración

En esta última iteración, se realizan las funcionalidades relacionadas con la gestión administrativa de la empresa. En ella se van a producir dos funcionalidades; una referida al cuadro de mando de la empresa y otra a la definición de KPI's. A esta parte sólo podrán acceder los empleados que son administradores de la empresa.

Un Cuadro de Mando está configurado por KPIs acompañados de una representación gráfica, de forma que se puede acceder a la información de una manera muy visual y ágil. Este tipo de herramienta, permite optimizar los procesos de toma de decisiones tanto estratégicas como tácticas. En esta aplicación se han incorporado indicadores económicos de distintas categorías: de ventas totales, de ventas por comercial y de ventas por tipo de mineral del producto.

Por lo tanto, el encargado de la administración va a poder definir los KPIs de la empresa para después visualizarlos a través del cuadro de mando.

## 6.6 Conclusiones

Con la aplicación web creada, los comerciales podrán saber cualquier información que deseen y realizar todo tipo de pedidos desde el propio lugar en el que se encuentre con el cliente. Además, los administradores podrán consultar el cuadro de mando de la empresa desde cualquier lugar, al igual que toda la información relacionada con los pedidos y los clientes.

A continuación, se muestra un diagrama que recoge las funcionalidades que puede realizar cada tipo de usuario de la aplicación web (figura 6.1 de la página 6.1).



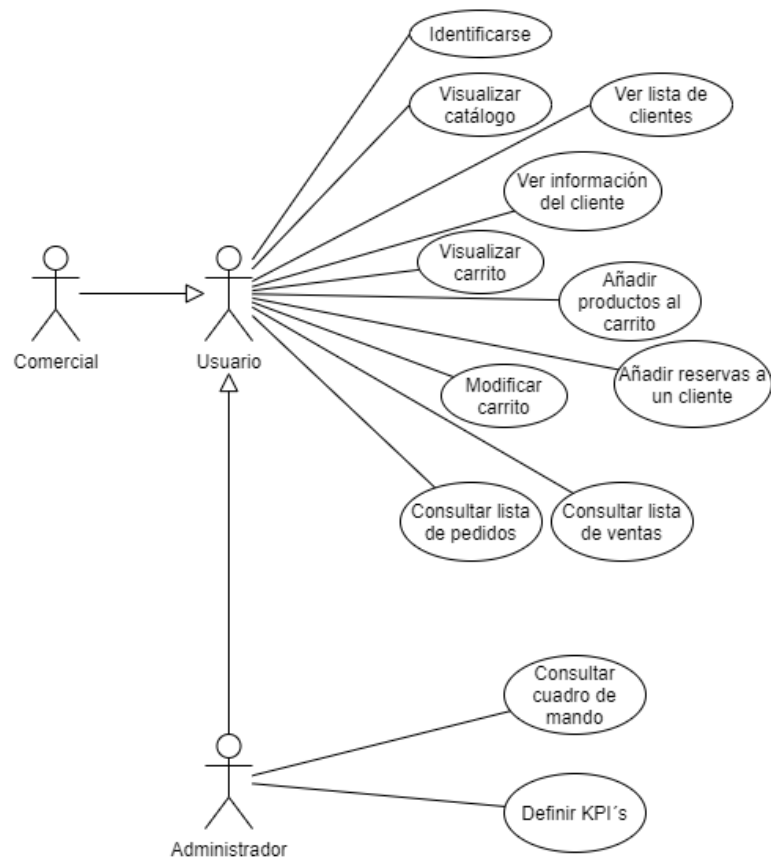


Figura 6.1: Casos de uso de la aplicación web.



# Modelado E/R

---

En este capítulo se muestra, en detalle, el modelo Entidad-Relación (E/R) que recoge la información de todo el proyecto. Gran parte de las entidades del modelo forman parte del ERP personalizado que ya dispone la empresa. Se han añadido algunas más para poder realizar la aplicación web del presente proyecto.

A continuación, se muestran dos diagramas E/R; uno con las entidades más relacionadas con el tema de reservas y facturación, Figura 7.1 en la página 35, y otro con las entidades más relacionadas con los artículos, Figura 7.2 en la página 36. Para una visualización de los diagramas más cómoda, se ha decidido no mostrar todos los atributos de las entidades.

Las tablas que han sido añadidas para desarrollar el TFG son: Reservas, Reservas\_lin, Reservas\_lineas\_series, Carritos, Carritos\_lin y Carritos\_lineas\_series.

Se han añadido algunos campos en las tablas ya existentes en el ERP personalizado como es el rol de los comerciales, que va a servir para restringir su acceso a algunas partes de la aplicación. También el bloquea\_stock que va a servir para comunicar a la empresa si la serie de un artículo está reservada y, por tanto, no se puede vender.

- **Usuarios:** tabla con los datos necesarios para acceder a la aplicación. En ella se van a registrar el login, password y rol de los usuarios necesarios para el uso del ERP de la empresa y la aplicación web. Para este proyecto los únicos roles disponibles van a ser el rol de administrador y el rol de comercial. Dependiendo de su rol van a tener restringido el acceso a ciertas páginas de la aplicación web. Su Primary Key (PK) es *CODIGO*.
- **Clientes:** tabla que muestra los datos de los clientes que tiene registrados la empresa. Su Primary Key (PK) es *CODIGO*. Se recogen datos como su teléfono, dirección, razón social, límite de crédito, forma de pago, etc. Además tiene como Foreign Key (FK) el código del comercial que lleva a ese cliente.
- **Comerciales:** tabla que recoge los datos de los comerciales que tiene la empresa. Su PK es *CODIGO*. En ella se recogen los datos básicos de DNI, nombre, teléfono, email y

- 
- dirección. Además tiene la PK que lo relaciona con un usuario, para poder tener acceso a la aplicación.
- **Reservas:** tabla que muestra todos los pedidos realizados por los distintos clientes de la empresa. Su PK es *CODIGO, ALMACEN*. En ella se recogen datos como la fecha de la reserva, el descuento, etc. Además dispone de dos FK que relaciona las reservas con los clientes y los comerciales a través de su código.
  - **Reservas\_lin:** tabla que recoge las líneas de las reservas. Es una entidad débil debido a que depende de las PK de la entidad Reservas. Su PK va a ser *CODIGO, ALMACEN, NUMERO\_INTERNO*. Dispone del código de artículo como FK para conocer que artículo se reserva. Se incluye la posibilidad de registrar la cantidad, precio, descuento, largo, ancho, espesor, comentarios y si está bloqueado en stock.
  - **Reservas\_lineas\_series:** tabla que recoge las distintas series que puede tener un mismo artículo que se registra en una línea de reserva. Se trata de una entidad débil de Reservas\_lin, por lo que su PK es *CODIGO, ALMACEN, NUMERO\_INTERNO, NUMERO\_SERIE*. Se añade un número de serie que distingue a los productos que tienen el mismo mineral, espesor, formato, acabado, colección y nombre de familia.
  - **Albaranes\_ventas:** tabla que recoge los albaranes creados por la empresa. Su PK es *CODIGO, ALMACEN*. Se obtienen datos típicos de un albarán de venta, como la fecha del albarán, el descuento, etc. Todo albarán va a tener un cliente y un comercial asociado (que son FK). También puede estar asociado a una factura o a un presupuesto (FK).
  - **Albaranes\_ventas\_lin:** tabla en la que se recogen las líneas de los albaranes de venta. Se trata de una entidad débil de Albaranes\_venta, por lo que su PK es *CODIGO, ALMACEN, NUMERO\_INTERNO*. Cada registro contiene el código de un artículo (FK), junto la cantidad, precio, descuento, largo, ancho, etc. Además puede estar asociada a una línea de reserva, que se mostraría con la siguiente FK *NUMERO\_RESERVA, NUMERO\_INTERNO*.
  - **Artículos:** tabla que recoge los datos de todos los artículos de la empresa. Su PK es *CODIGO*. Entre los datos que describen el producto destacan las FK, de las que dispone cada artículo, que son los códigos del mineral, espesor, formato, acabado, colección y nombre de familia. Esta clasificación tan particular es debido a que facilitará la navegación entre artículos que se utilizará en la aplicación web.
  - **Facturas\_ventas:** tabla con los datos de facturación de la empresa. Su PK son *CODIGO, ALMACEN*. En ella se guarda el cliente al que se le hace la factura (FK) y la forma de pago (FK).

- **Facturas\_ventas\_vctos:** tabla en la que se muestran los vencimientos que tienen las facturas de venta. Es una entidad débil que depende de facturas\_venta y su PK es *CODIGO, ALMACEN, ID*. En ella se guardan datos como el importe, la fecha de pago y la forma de pago, siendo esta última una PK.
- **Formas\_pago:** tabla que recoge los distintos modos de pago. Su PK es *CODIGO* que va acompañado de una descripción de la forma de pago.
- **Formas\_pago\_lin:** tabla con los distintos campos necesarios para cada forma de pago distinta. Es una entidad débil de formas de pago y su PK es *CODIGO, DIA*. En ella se guardan el número de días en los que se puede realizar el pago y el porcentaje que corresponde a esos días.
- **Presupuestos:** tabla en la que se muestran los distintos presupuestos que ha generado la empresa. Estos presupuestos son guardados para posibles albaranes de venta. Su PK es *CODIGO, ALMACEN*. Asimismo dispone de las FK que la relacionan con un cliente, un comercial y una forma de pago.
- **Presupuestos\_lin:** tabla que muestra los datos de las líneas de cada presupuesto generado. Es una entidad débil y su PK es *CODIGO, ALMACEN, NUMERO\_LINEA*. En ella se detallan datos como precio, cantidad, descuento y artículo que es una FK.
- **Carritos:** tabla que se crea cada vez que un comercial añade artículos que desea un cliente al carrito. Su PK es *CODIGO*. Guarda entre sus datos las FK del comercial y del cliente, además de registrar la fecha y observaciones.
- **Carritos\_lineas:** tabla que muestra los datos de las líneas de cada carrito. Es una entidad débil que depende de carrito y su PK es *CODIGO, NUMERO\_LINEA*. En ella va a guardarse el código del artículo PK con la cantidad, precio, unidad y si está bloqueado en stock.
- **Carritos\_lineas\_series:** tabla que muestra las distintas series que puede tener un artículo en una línea del carrito. Es una entidad débil de carritos\_lineas y su PK es *CODIGO, NUMERO\_LINEA, NUMERO\_SERIE*. Además se registran datos de su largo, ancho, espesor.
- **Rutas:** tabla con las distintas rutas que van asociadas a cada cliente. Su PK es *CODIGO*. Tiene como FK el código de cliente y también dispone de un campo para comentarios sobre cada ruta.
- **Espesores:** tabla con los distintos espesores que puede adquirir un artículo. Su PK es *CODIGO* y además se incluye una descripción y comentarios.

- 
- **Acabados:** tabla con los distintos acabados que puede tener un artículo. Su PK es *CODIGO* y además se incluye una descripción y comentarios.
  - **Colecciones:** tabla con las distintas colecciones que puede tener un artículo. Su PK es *CODIGO* y además se incluye una descripción y comentarios.
  - **Nombres\_familias:** tabla con los distintos nombres de familia por los que se puede dividir un artículo. Su PK es *CODIGO* y además se incluye una descripción y comentarios.
  - **Minerales:** tabla con los distintos minerales que puede tener un artículo. Su PK es *CODIGO* y además se incluye una descripción y comentarios.
  - **Formatos:** tabla con los distintos formatos que puede tener un artículo. Su PK es *CODIGO* y además se incluye una descripción y comentarios.

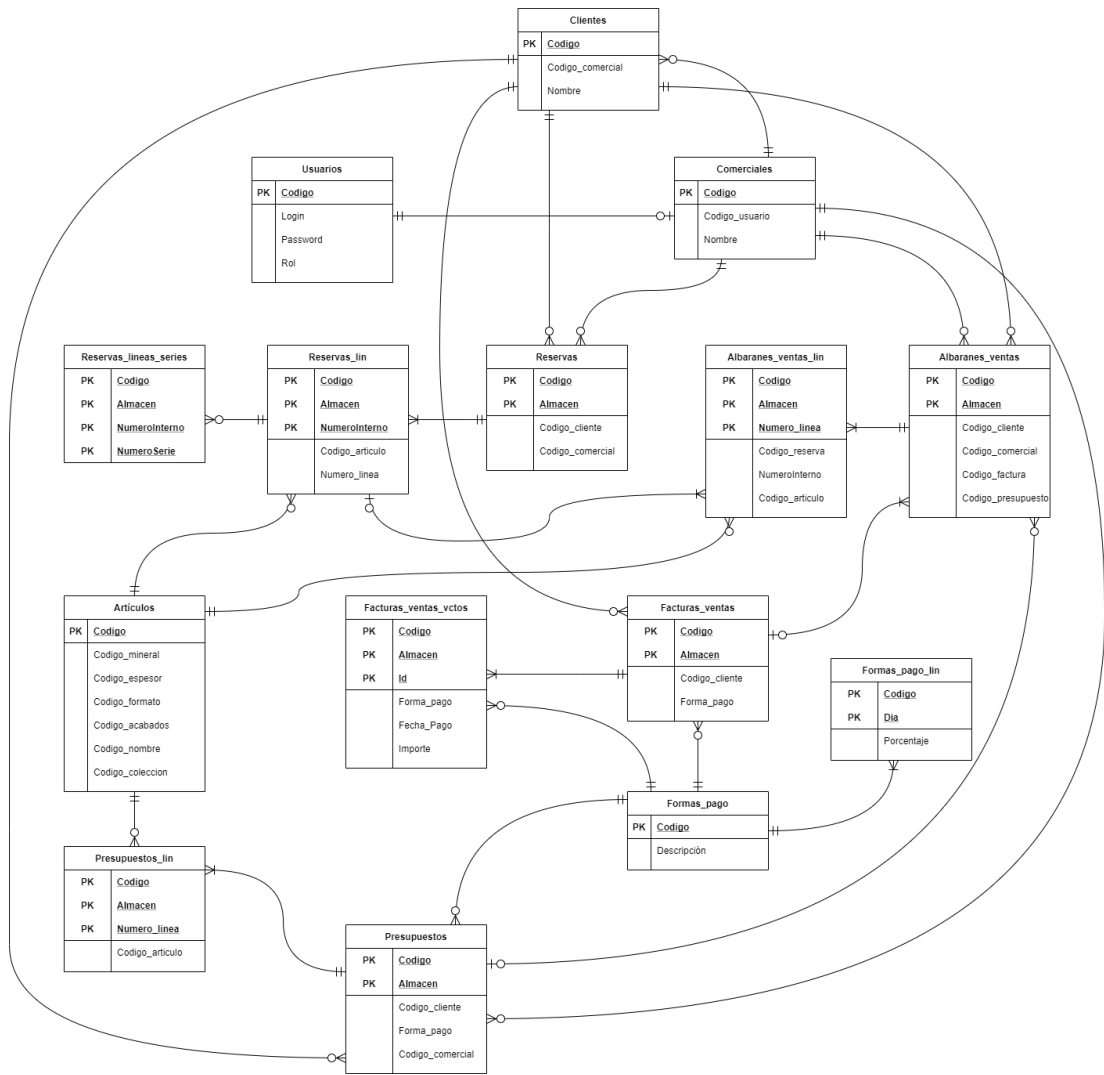


Figura 7.1: Diagrama E/R de la parte de facturación y reservas.

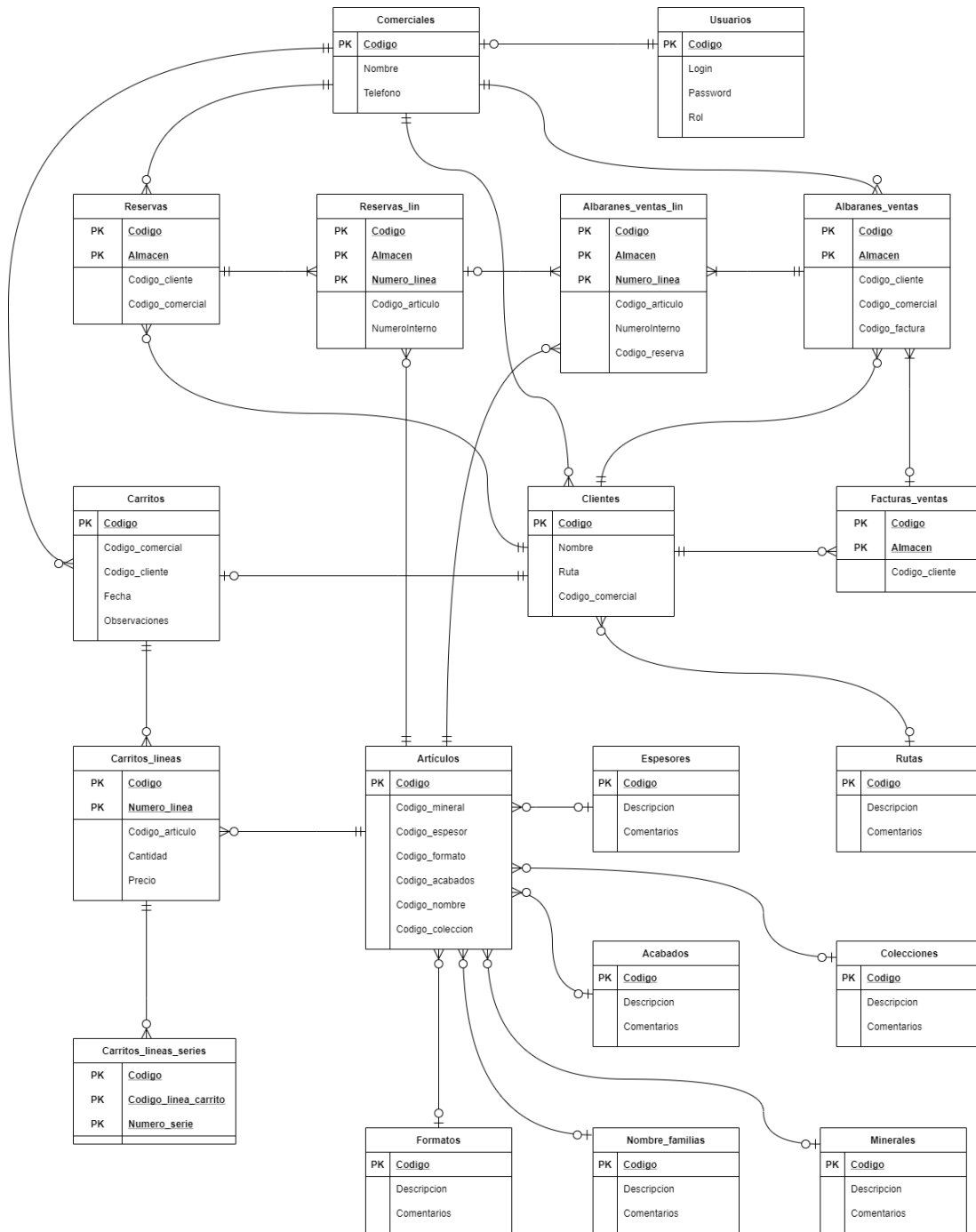


Figura 7.2: Diagrama E/R de artículos.



## Capítulo 8

# Diseño

---

En esta fase, se busca una solución conceptual que satisfaga los requisitos obtenidos en la fase de análisis. Para ello, se van a ir mostrando diferentes diagramas que permitan visualizar mejor las clases que se involucran en el sistema. Todos los diagramas cumplen con el patrón MVC, por lo que su diseño es similar, cambiando únicamente las funciones de cada clase.

Básicamente, en el diseño de la aplicación, se puede apreciar que un usuario (un administrador o un comercial) interactúa con la aplicación a través de la vista, que esta le envía una petición al controlador. Dependiendo del tipo de petición que se trate, el controlador le envía unos datos que desea consultar, insertar, modificar o eliminar de la base de datos. Este tipo de operaciones las realiza el modelo, que una vez las finaliza le devuelve la respuesta deseada al usuario pasando antes por el controlador y la vista.

El diseño de la base de datos de la aplicación se ha mostrado en la sección 7 (página 31).

### 8.1 Primera iteración

Las funcionalidades que se añaden en la primera iteración son: la de autenticación de los usuarios y la visualización de los productos de la empresa. Ambas pueden ser realizadas tanto por comerciales como por administradores. Para ello se ha realizado el diseño a través de los siguientes diagramas de clases (8.1 y 8.2 de la página 39).

### 8.2 Segunda iteración

En esta segunda iteración se añaden las funcionalidades de visualización de todos los clientes, sus detalles y la visualización de la lista de ventas. Ambas pueden ser realizadas por cualquier tipo de usuario. Al comercial se le mostrarán únicamente los datos de sus clientes y sus ventas, mientras que los administradores pueden consultar todos los datos. A continuación se muestran los diagramas de clases para cumplir con todos estos requisitos (8.3 y 8.4 de la 40).

### 8.3 Tercera iteración

En la tercera iteración, se presentan las funcionalidades relacionadas con la trazabilidad de un pedido. Estas son: añadir productos al carrito, modificar o eliminar el carrito de un cliente, confirmar esa compra y mostrar la lista de pedidos. La elección de los productos que se pueden añadir al carrito se hace a través del catálogo. Por tanto, para esa funcionalidad se debe modificar la vista del listado de productos, y dar la opción de añadir al carrito de compra. En el diagrama de clase 8.5 de la página 41 se muestra el diseño que se debe seguir para completar esta iteración.

### 8.4 Cuarta iteración

Para finalizar, en la última iteración, se añaden los requisitos necesarios para visualizar un cuadro de mando mediante la definición de KPIs. Para ello se han diseñado los diagramas 8.6 y 8.7 de las páginas 42 y 43, donde se pueden apreciar diferentes clases. En ellas se muestran las funciones necesarias para cumplir con el patrón MVC y completar todas las funcionalidades que faltan de la aplicación.

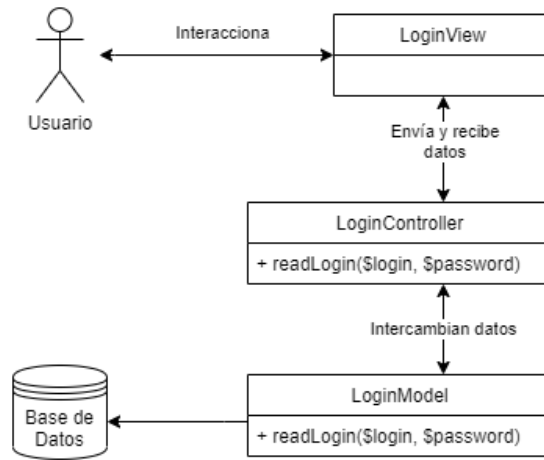


Figura 8.1: Diagrama de clases del Login.

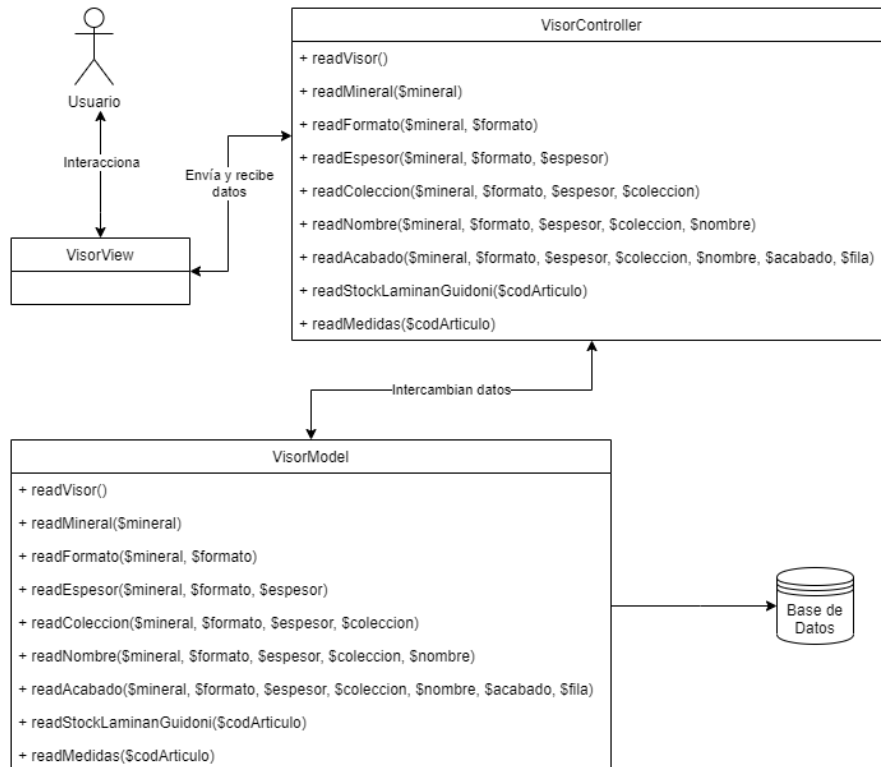


Figura 8.2: Diagrama de clases de Visor.

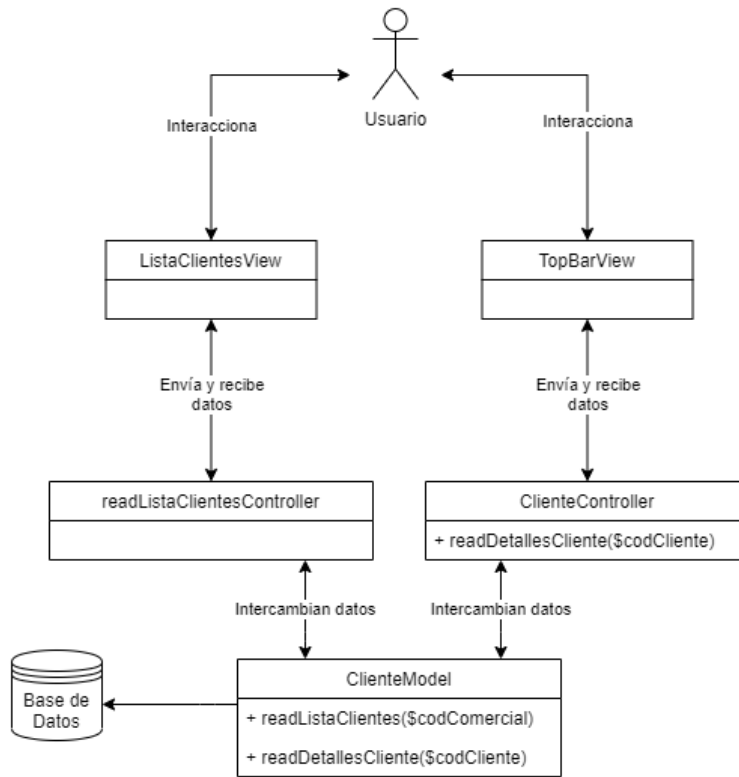


Figura 8.3: Diagrama de clases de Clientes.

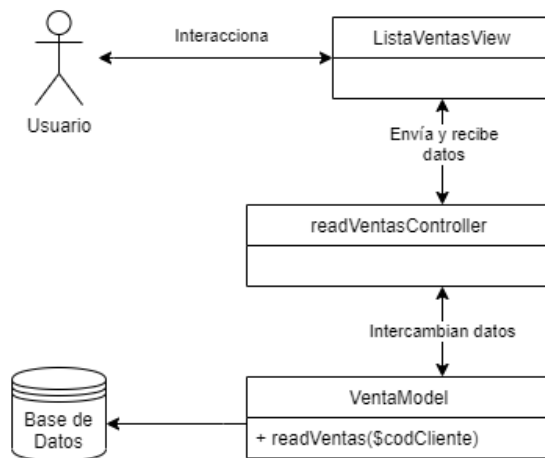


Figura 8.4: Diagrama de clases de Ventas.

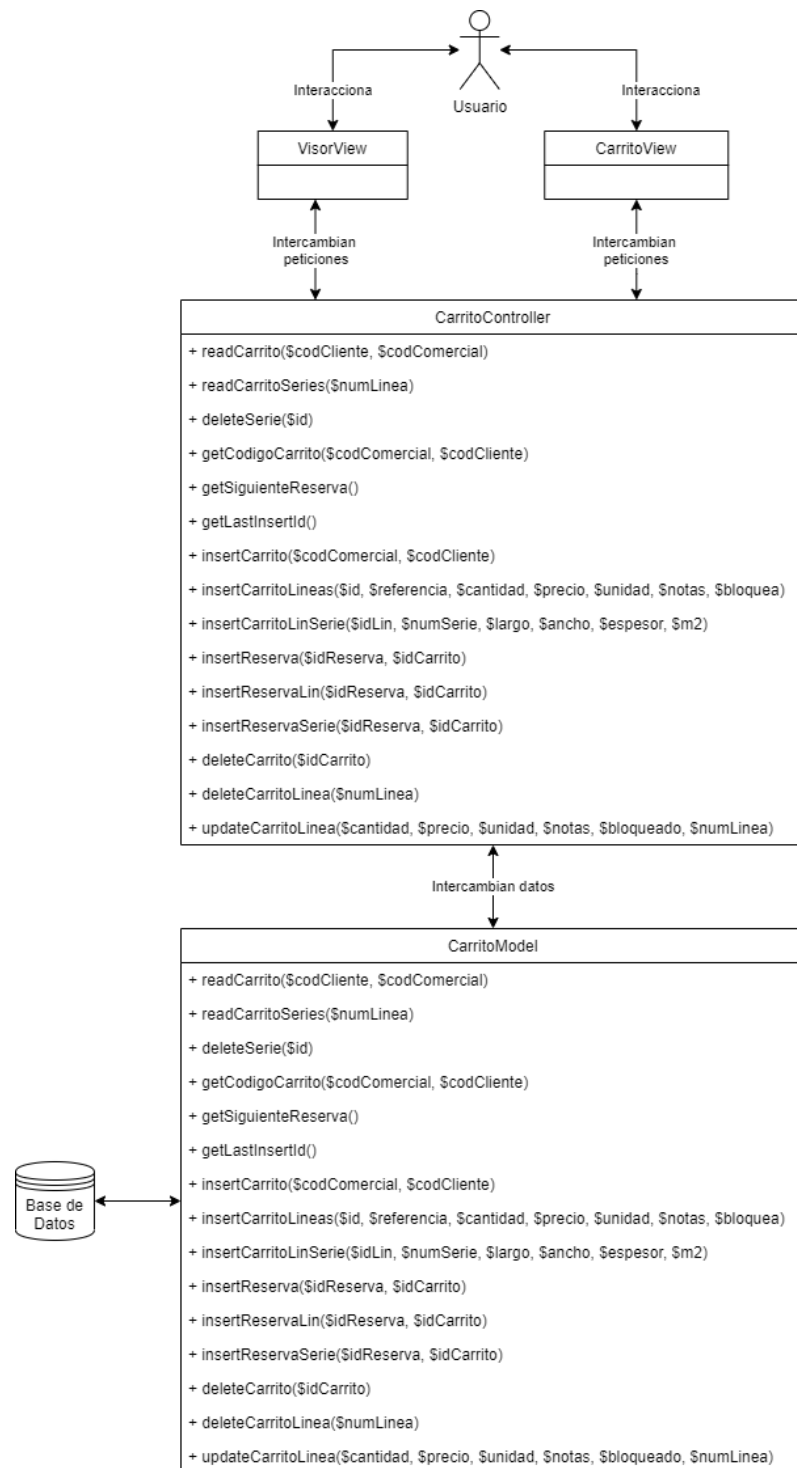


Figura 8.5: Diagrama de clases de Carrito.

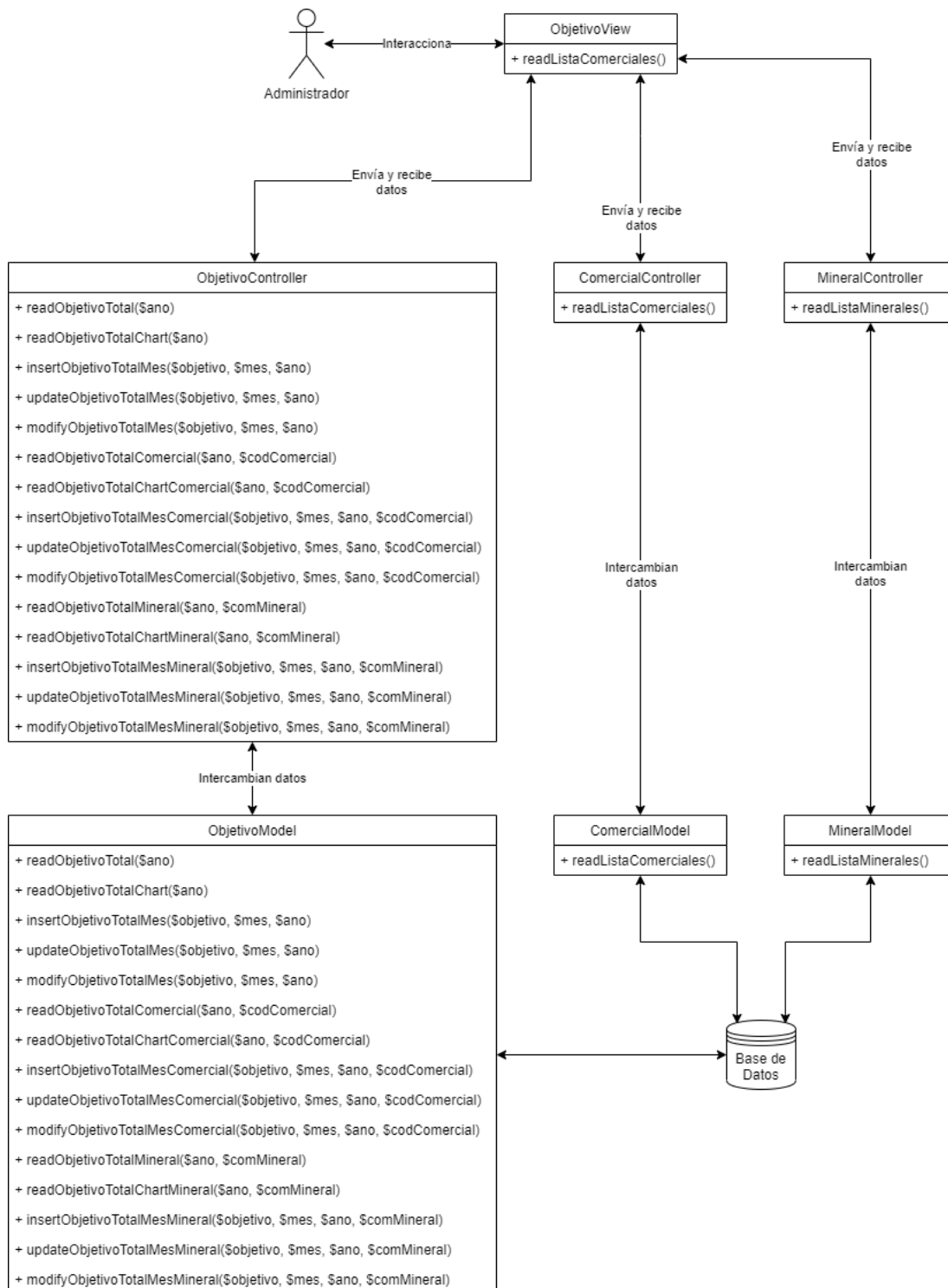


Figura 8.6: Diagrama de clases de Definición KPIs.

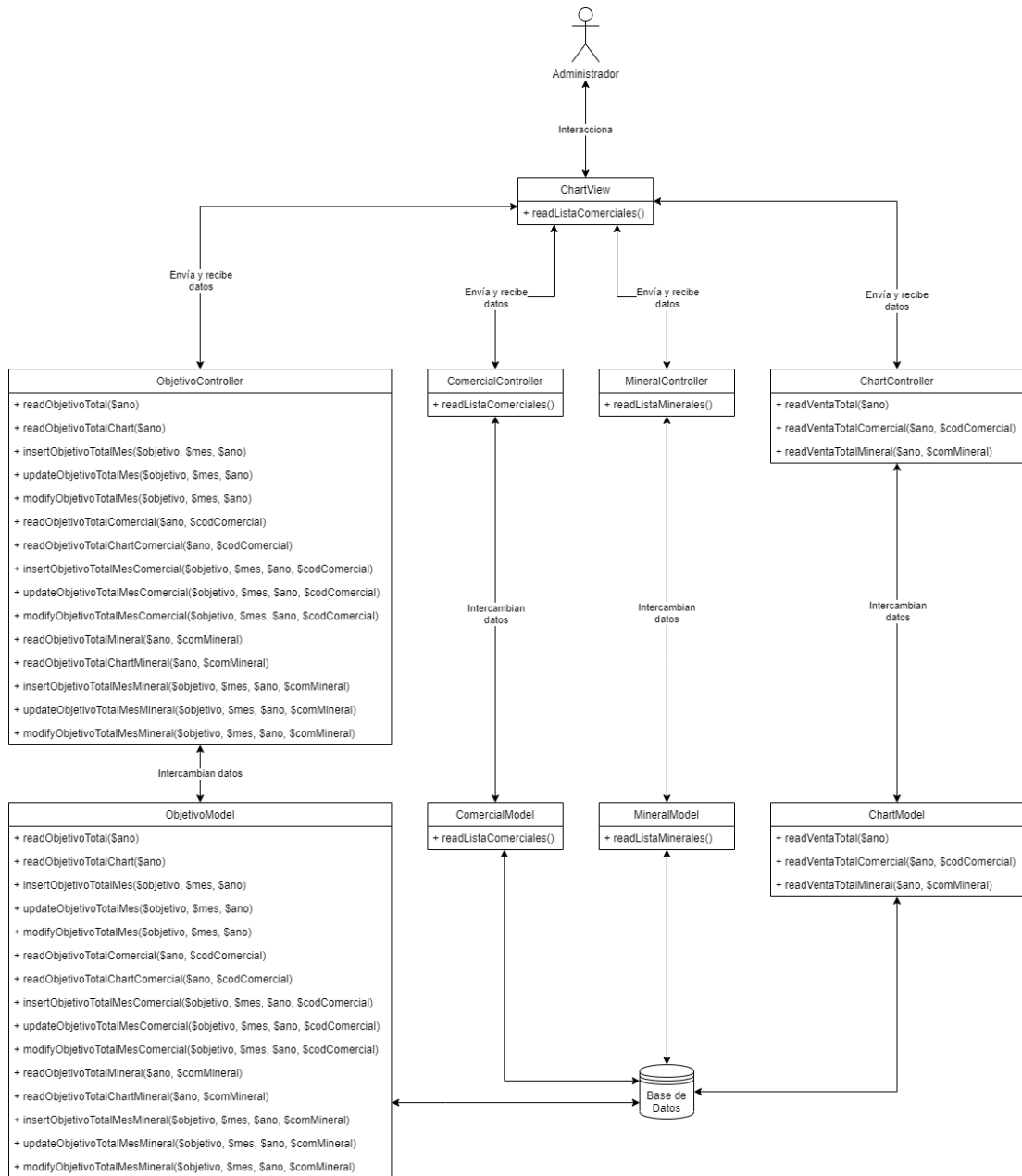


Figura 8.7: Diagrama de clases del Cuadro de Mando.





# Implementación

---

Tras analizar (6) y diseñar (8) la solución web para la empresa, es necesario enfrentarse a la etapa de implementación del programa. Esta fase es la construcción definitiva donde se elaboran, adaptan y añaden los elementos previamente contemplados.

El modelo diseñado en la fase anterior (8) es la guía para comenzar a programar las funciones que debe cumplir la aplicación. Se respeta el patrón MVC, explicado en la sección 4.2.2 de la página 17, para elaborar el software.

A continuación se van a documentar todas las implementaciones que se han realizado por cada iteración.

### 9.1 Primera iteración

En la primera iteración, se han elaborado los archivos PHP que se muestran en la figura 9.1 de la página 47 para poder obtener el primer entregable de la aplicación web.

Una de las funciones básicas, de cualquier aplicación, es tener una autenticación de usuarios. Por este motivo, se comienza a construir la aplicación atendiendo a esta funcionalidad. No se añade el registro de usuarios, otra funcionalidad básica, debido a que la empresa da de alta a sus empleados a través del ERP personalizado del que ya disponen.

En segundo lugar, procedemos a implementar el visor de los productos de la empresa, como hemos acordado en la fase de análisis en las reuniones con el cliente. Para la visualización del catálogo se ha optado por poder seleccionar la categoría de cada producto (tipo de mineral, formato, espesor, colección, familia y acabado) para llegar a obtener un producto final.

A continuación se explican los archivos generados en esta primera iteración:

- **Model:**

- **DB\_connection.php:** contiene la conexión con la base de datos de la aplicación. Este archivo va a ser incluido en todas las vistas de la aplicación, debido a que es

imprescindible tener conexión con la base de datos para poder realizar cualquier operación.

- **Usuario.php**: define la clase Usuario con sus campos, que van a ser necesarios para loguearse en la aplicación.
- **UsuarioModel.php**: archivo que contiene la función para leer la información de un usuario.
- **ElementosVisor.php**: define la clase ElementosVisor con los campos que van a ser manejados en los distintos productos en el visor.
- **VisorModel.php**: contiene todas las funciones necesarias para consultar los datos, a través de la base de datos, relacionados con el visionado de los productos.

- **Controller:**

- **Comprueba\_login.php**: verifica si un usuario se loguea con éxito después de poner el login y password.
- **Comprueba\_seguridad.php**: comprueba si un usuario está autenticado. Va a ser utilizado por todas las vistas, ya que es necesario estar logueado para poder realizar cualquier operación en la aplicación web. Redirecciona a la vista del login si no está autenticado.
- **Logout.php**: archivo que cierra la sesión del usuario que está logueado.
- **VisorController.php**: contiene todas las funciones que necesita la vista del catálogo de productos. En ellas se llaman a las creadas por el modelo.
- **LoginController.php** archivo con todas las funciones necesarias para la comunicación entre el modelo y la vista de la autenticación de los usuarios.

- **Views:**

- **TopBar.php**: archivo con el html necesario para generar la vista de la barra superior de la aplicación. Muestra el nombre del usuario que está conectado en la parte derecha. También va a mostrar la información de un cliente en cuanto se añada su funcionalidad.
- **Sidebar.php**: genera la vista de la barra izquierda de la aplicación. En ella se muestran los distintos enlaces para navegar por toda la aplicación y el logo de la empresa. En esta primera iteración sólo va a contener el enlace a los productos de la empresa.
- **Footer**: crea la barra inferior de la aplicación en la que se muestra el Copyright de la empresa.

- **CommonScripts.php**: archivo con diferentes scripts que se utilizan en la mayoría de vistas, y que por comodidad se juntan todos en un mismo archivo.
- **Visor.php**: genera el visor de productos de la empresa. Muestra las distintas clasificaciones que puede obtener un producto para que la visualización sea más cómoda. Empieza mostrando los distintos tipos de minerales de los que se dispone. Una vez seleccionado uno, muestra el tipo de formato que puede tener con ese mineral. Va repitiendo esta acción con los distintos espesores, colecciones, nombres de familia y acabados, hasta que se obtiene el producto final. En esta primera iteración, sirve simplemente para mostrar el producto, pero en futuras iteraciones se podrá añadir ese producto a un pedido.
- **Login.php**: vista con el formulario para identificarse en la aplicación. Se piden los campos de login y password que van a ser comprobados por su controlador correspondiente. Avisa si algún campo no es correcto.
- **LogOut.php**: archivo que muestra una ventana emergente para asegurarse que el usuario desea cerrar la sesión.

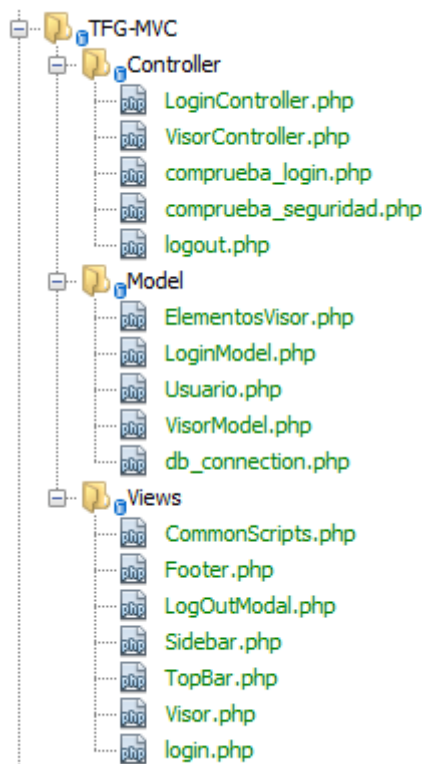


Figura 9.1: Captura de los archivos generados en la primera iteración.

Una vez que se genera toda la implementación para la primera iteración, la aplicación va a obtener su primera versión. Con ella se puede autenticar un usuario de la empresa (un administrador o un comercial) y navegar por las distintas categorías que tiene un producto. A continuación se muestran las dos vistas principales de esta iteración (figuras 9.5 y 9.6 en la página 54).

## 9.2 Segunda iteración

Para esta segunda iteración, se ha implementado una vista con el listado de todos los clientes de la empresa. Si el usuario de la aplicación es un comercial, entonces se mostrarán únicamente los clientes que lleva a su cargo.

También se ha creado otra vista con la lista de todas las ventas que se han realizado a un cliente seleccionado, además de mostrar información clave para los comerciales.

Para generar todas estas funcionalidades, se han añadido distintos archivos PHP y se han modificado algunos de los ya creados en la iteración anterior. En la figura 9.2 de la página 49 se muestran todos esos archivos que van a ser explicados a continuación.

- **Model:**

- **Cliente.php:** define la clase Cliente con los campos necesarios.
- **ClienteModel.php:** contiene las funciones necesarias para extraer los datos de los clientes de la base de datos.
- **VentaModel.php:** obtiene los datos de las ventas registrados en la base de datos.

- **Controller:**

- **ClienteController.php:** contiene la conexión entre la vista y el modelo para conseguir los detalles de un cliente.
- **ReadListaClientesController.php:** intercambia los datos de lista de clientes que se desea mostrar en la vista. Lo hace llamando a una función de la capa modelo.
- **ReadVentasController.php** crea una lista de las ventas de un cliente concreto en formato JSON, para que sea llamado por la vista correspondiente. Consigue esos datos a través del modelo.

- **Views:**

- **ListaClientes.php:** genera una vista con la lista de clientes solicitados al controlador. Esta lista va a variar según el usuario que esté utilizando la aplicación.

- **ListaVentas.php:** mediante el html necesario, genera el listado de ventas de un cliente en concreto.
- **TopBar:** se ha modificado este archivo para que muestre, en la barra superior de la aplicación, los datos de un cliente seleccionado. Con esto se consigue visualizar siempre los datos de un cliente en todas las pantallas del programa.
- **Sidebar.php:** archivo modificado para mostrar los enlaces a la lista de clientes y de ventas.

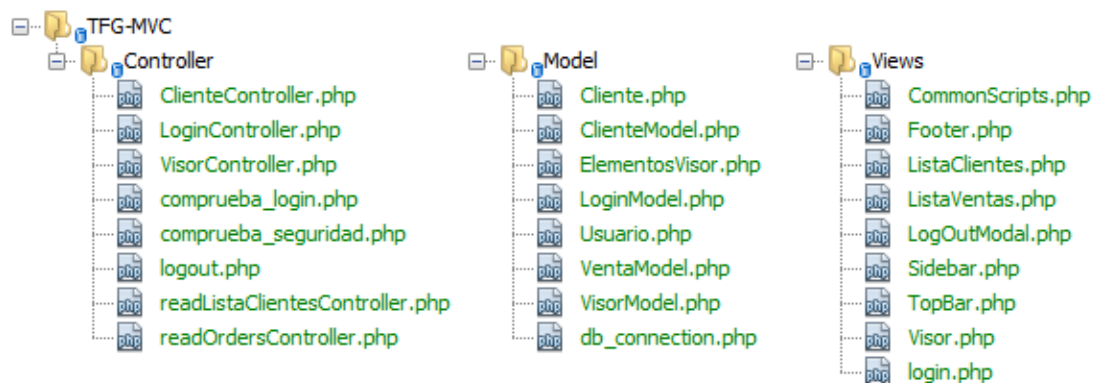


Figura 9.2: Captura de los archivos generados hasta la segunda iteración.

Al finalizar esta segunda iteración, el usuario dispone de la posibilidad de saber la información de los clientes que lleva a su cargo y sus ventas. Mas adelante, en la página 55, se muestran las vistas que incluyen estas funcionalidades.

### 9.3 Tercera iteración

Con todos los clientes disponibles y la posibilidad de poder seleccionar a uno en particular, lo que queda para terminar con la gestión comercial, es añadir pedidos a un cliente.

Para eso se ha creado la opción de añadir productos a un carrito, con la posibilidad de modificar o eliminar su contenido. Una vez decidido los productos que quiere pedir un cliente, lo único que falta es poder confirmar ese pedido, función que realizamos en esta iteración. Por último, también se ha implementado el listado de la situación de los pedidos de todos los clientes. Dependiendo del rol del usuario le aparecerán unos, otros o todos.

Todas estas funcionalidades se han llevado a cabo generando archivos PHP, que se muestran en la figura 9.3 de la página 51. Se explica su contenido seguidamente.

- **Model:**

- **Carrito.php:** define la clase Carrito con la información de cada línea del carrito.
- **CarritoSerie.php:** define la clase CarritoSerie con los datos de la serie de un producto que se añade a carrito.
- **CarritoModel.php:** contiene las funciones necesarias para consultar, insertar, modificar y eliminar los datos del carrito de un cliente de la base de datos.
- **Pedido.php:** define la clase Pedido con los datos de los pedidos de un cliente.
- **PedidoModel.php:** consigue los datos de los pedidos a través de la base de datos.
- **Ruta.php:** define la clase Ruta con los datos de una ruta.
- **RutaModel.php:** obtiene la lista de las distintas rutas de la empresa.

- **Controller:**

- **CarritoController.php:** contiene la conexión entre la vista y el modelo para gestionar los datos de un carrito.
- **PedidoController.php:** archivo con la función necesaria para intercambiar los datos de los pedidos entre la vista y el modelo.
- **RutaController.php:** archivo con la función necesaria para intercambiar los datos de las rutas existentes en la empresa.

- **Views:**

- **addProduct.php:** graba un producto en el carrito tras ser añadido a través del visor de productos.
- **updateCarrito.php:** modifica los datos de las líneas de un carrito.
- **Borrarserie.php:** elimina la serie de un producto de la línea de un carrito.
- **Carrito.php:** crea la vista con los datos que se añadieron al carrito de un cliente.
- **ConfirmPedido.php:** genera un pedido con los datos que estaban en el carrito de un cliente.
- **ListaPedidos.php:** vista de la situación de los pedidos de todos los clientes que están a cargo del usuario de la aplicación. Si se trata de un administrador, muestra la situación de todos los clientes de la empresa.
- **Visor.php:** archivo modificado para poder añadir un producto en el carrito de compra. Se muestra un formulario cuando se llega a la selección de un producto final.

- **Sidebar.php**: se ha modificado para que la barra izquierda muestre los enlaces al carrito y a la situación de pedidos.

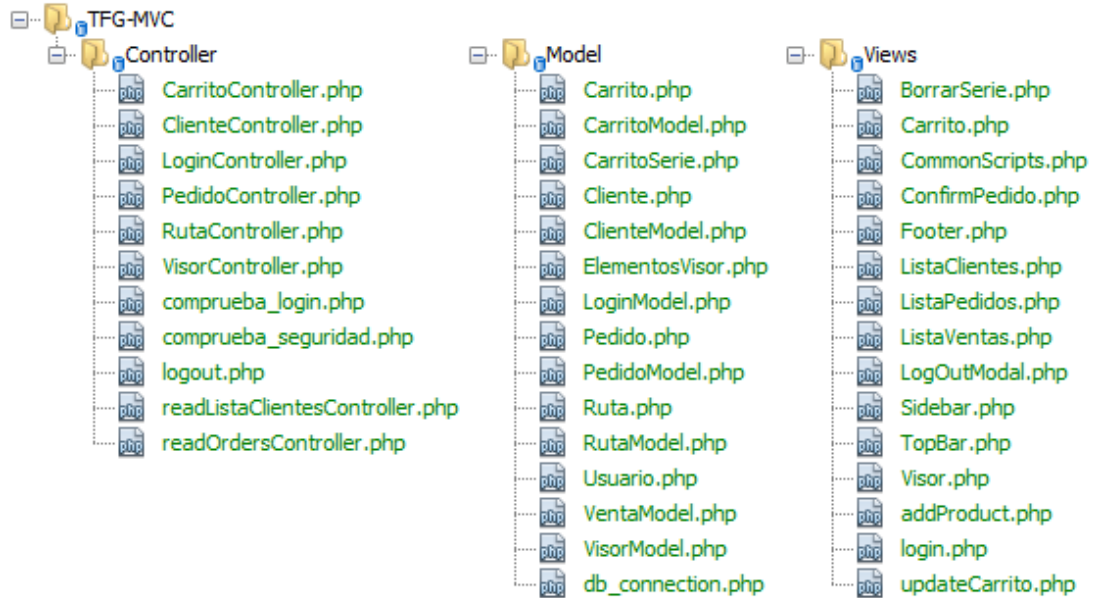


Figura 9.3: Captura de los archivos generados hasta la tercera iteración.

Con esta iteración terminada, la aplicación ya dispone de todas las funcionalidades necesarias para la gestión comercial. Por tanto, un comercial dispone de una trazabilidad completa de sus ventas, sabiendo en todo momento la información que necesita desde cualquier lugar. En las páginas 56 y 57 se presentan las vistas que incluyen las novedades implementadas en esta tercera iteración.

## 9.4 Cuarta iteración

Los últimos requisitos que faltan por implementar son los relacionados con la gestión administrativa de la empresa. En la fase de análisis, tras las reuniones con el cliente y jefe de proyecto, se ha determinado que la aplicación debe incluir un apartado con un cuadro de mando para controlar dicha gestión.

En esta última iteración se han completado esos requisitos, creando en un apartado la definición de KPIs y en otro la representación gráfica de la actividad económica de la empresa.

Para ello, se han añadido distintos archivos PHP para completar la aplicación, quedando esta como se muestra en la figura 9.4 de la página 53. A continuación se detallan las novedades que presentan estos archivos.

- **Model:**

- **Comercial.php:** define la clase Comercial con la información de un comercial.
- **ComercialModel.php:** obtiene la lista de comerciales de la empresa.
- **Mineral.php:** define la clase Mineral con los campos propios de un mineral.
- **MineralModel.php:** contiene la lista de minerales de la empresa.
- **ChartModel.php:** consigue los datos de las ventas totales de la empresa según la categoría deseada.
- **ObjetivoModel.php:** contiene las funciones necesarias para gestionar la definición de KPIs.

- **Controller:**

- **ComercialController.php:** contiene la función precisa para intercambiar la lista de comerciales entre la vista y el modelo.
- **MineralController.php:** archivo con la función necesaria para intercambiar la lista de minerales entre la vista y el modelo.
- **ChartController.php:** archivo con la función necesaria para intercambiar los datos de las ventas totales de la empresa entre la vista y el modelo.
- **ObjetivoController.php:** contiene la conexión entre la vista y el modelo para gestionar los datos de la definición de KPIs de la empresa.
- **Comprueba\_rol.php:** comprueba el rol del usuario conectado. Va a ser utilizado para las vistas en la que se necesita restringir el acceso a los comerciales.

- **Views:**

- **confirmObjetivos.php:** graba los objetivos marcados para la definición de KPIs tras ser añadidos a través de la vista.
- **Objetivos.php:** vista con la definición de KPIs para las distintas categorías.
- **Chart.php:** vista con los dashboard que permiten ver la evolución de la actividad de la empresa.
- **Error\_rol.php:** vista que muestra la advertencia de que esa página no es accesible para ese tipo de usuario.
- **Sidebar.php:** se ha modificado para que la barra izquierda muestre los enlaces a la definición de KPIs y al cuadro de mando cuando hay un administrador utilizando la aplicación.



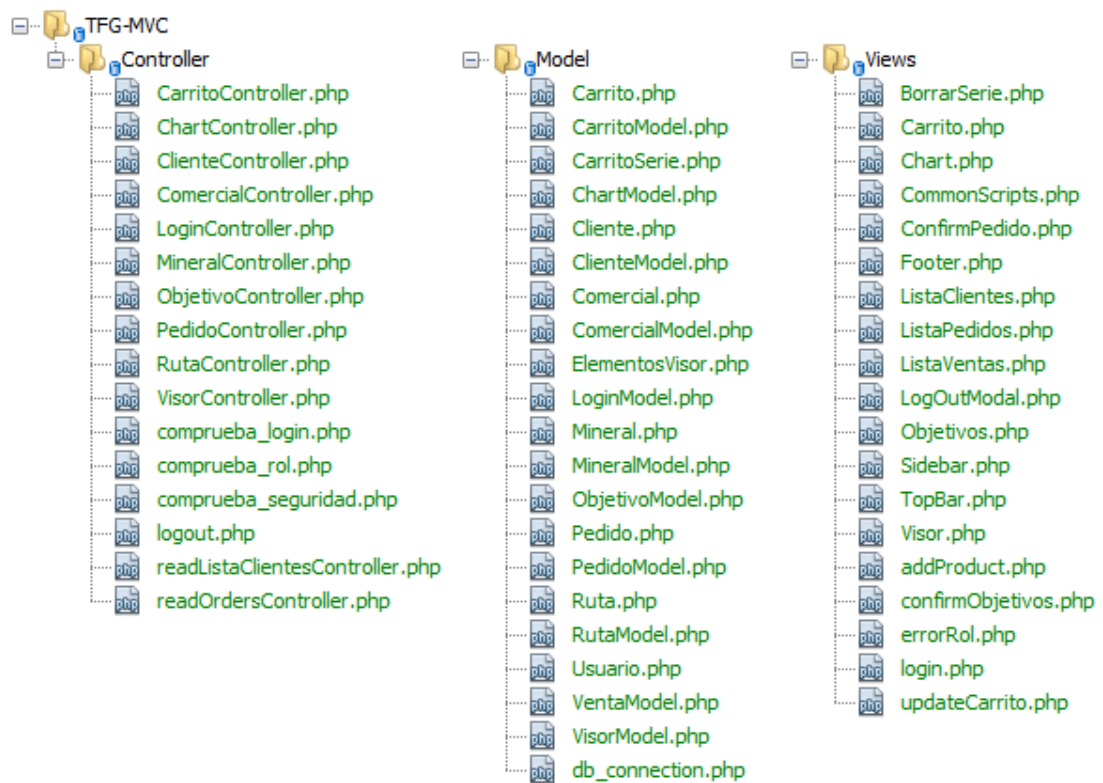


Figura 9.4: Captura de los archivos generados hasta la cuarta iteración.

Con esta iteración completada, la aplicación web ya dispone de todas las funcionalidades demandadas por el cliente. En esta última, se han añadido las relacionadas con la gestión administrativa de la empresa. En las páginas 57 y 58 se muestran las vistas con estas nuevas funciones.

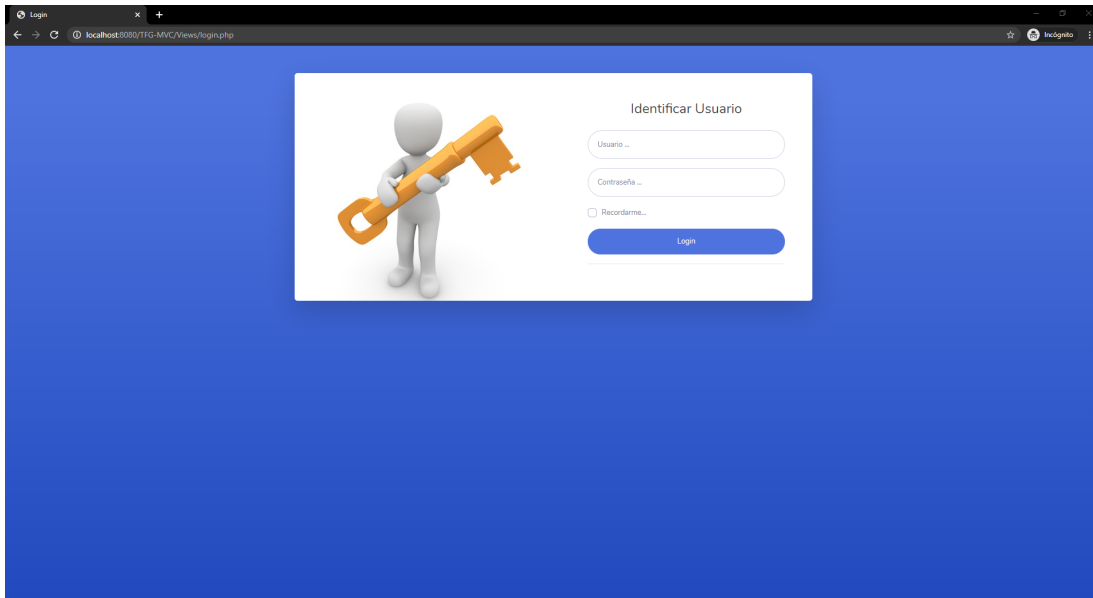


Figura 9.5: Vista de la autenticación de usuario.

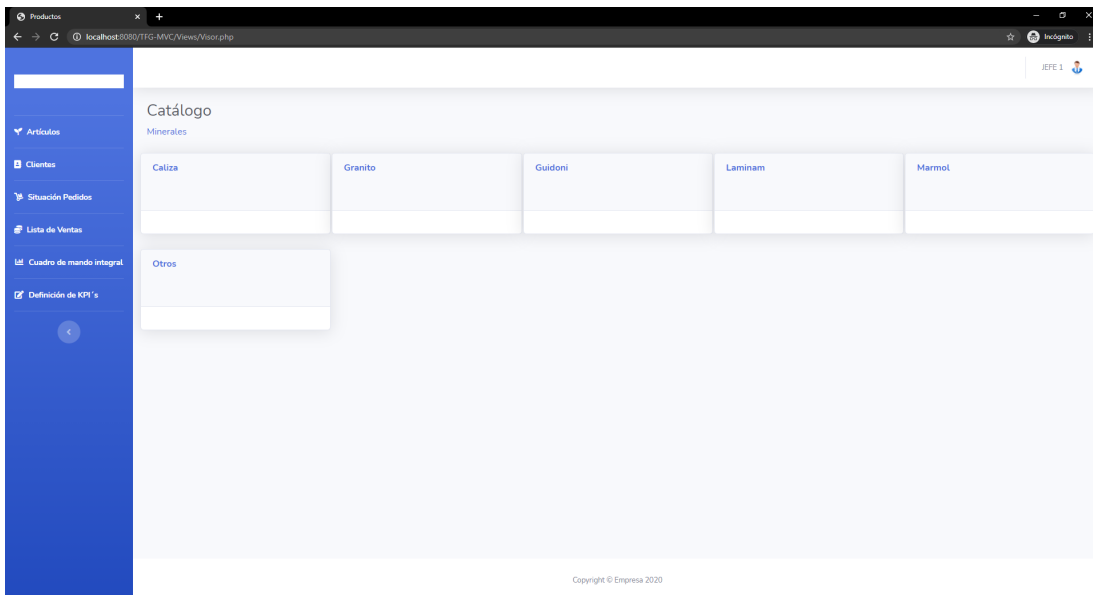


Figura 9.6: Vista del visor de productos de la empresa.

## CAPÍTULO 9. IMPLEMENTACIÓN

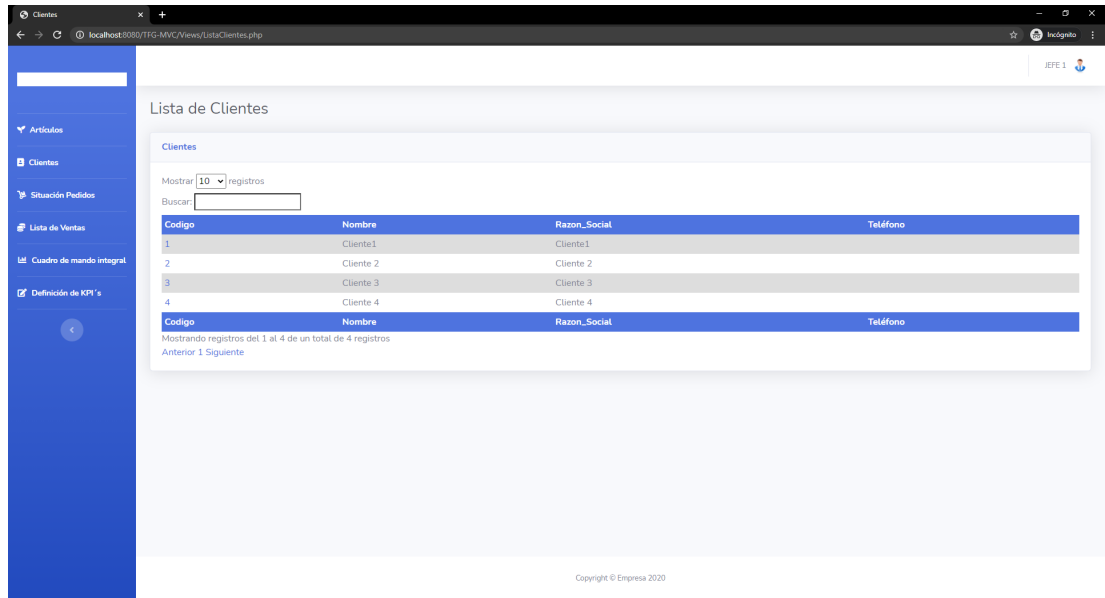


Figura 9.7: Vista de la lista de clientes.

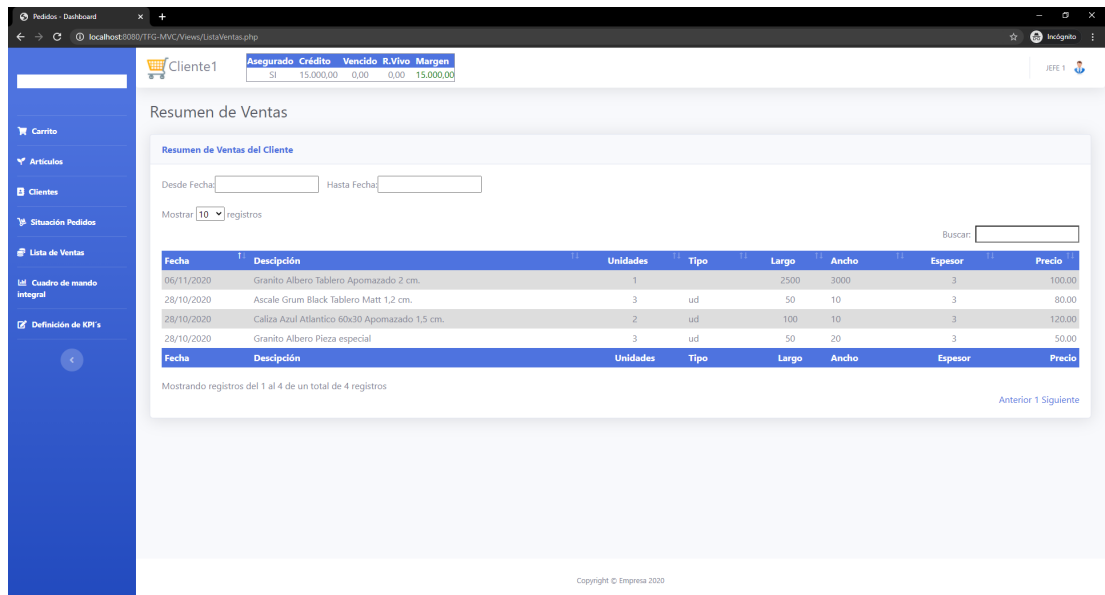


Figura 9.8: Vista de la lista de ventas de un cliente.

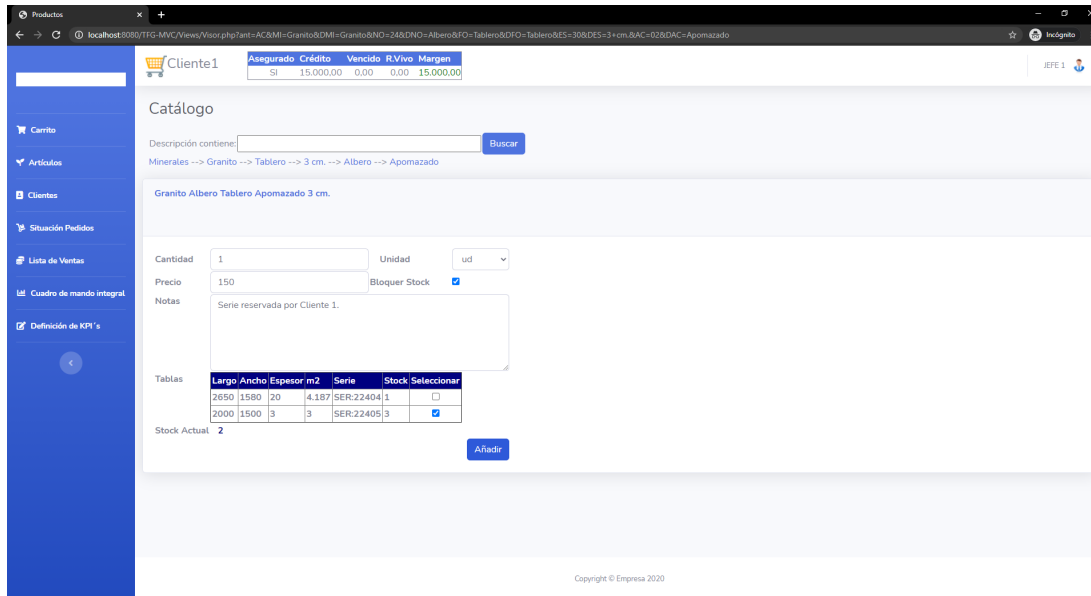


Figura 9.9: Vista donde se añaden los productos al carrito.

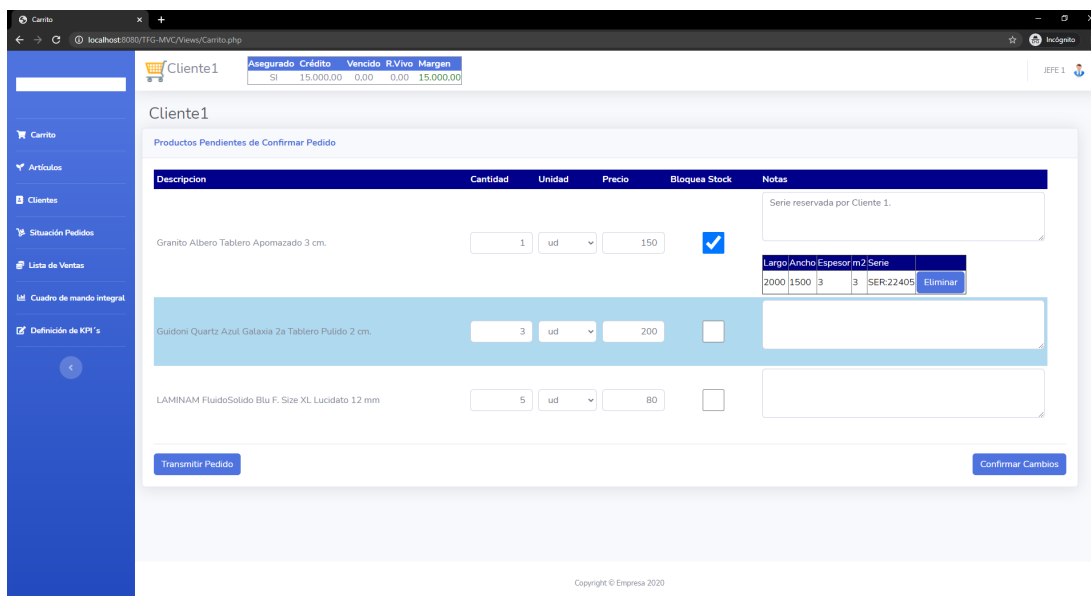
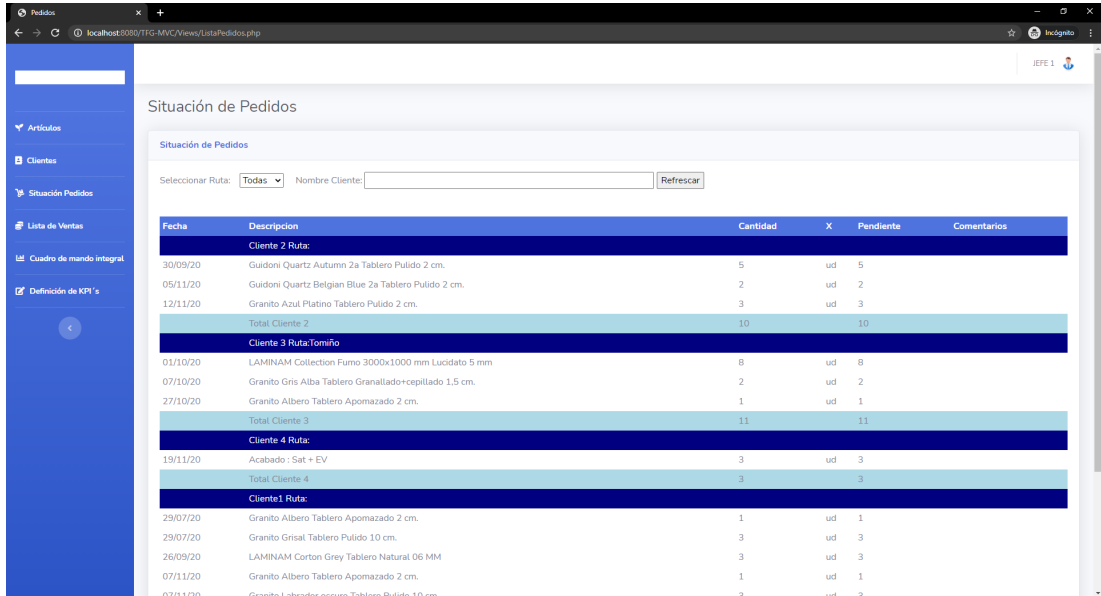


Figura 9.10: Vista del carrito de un cliente.

## CAPÍTULO 9. IMPLEMENTACIÓN

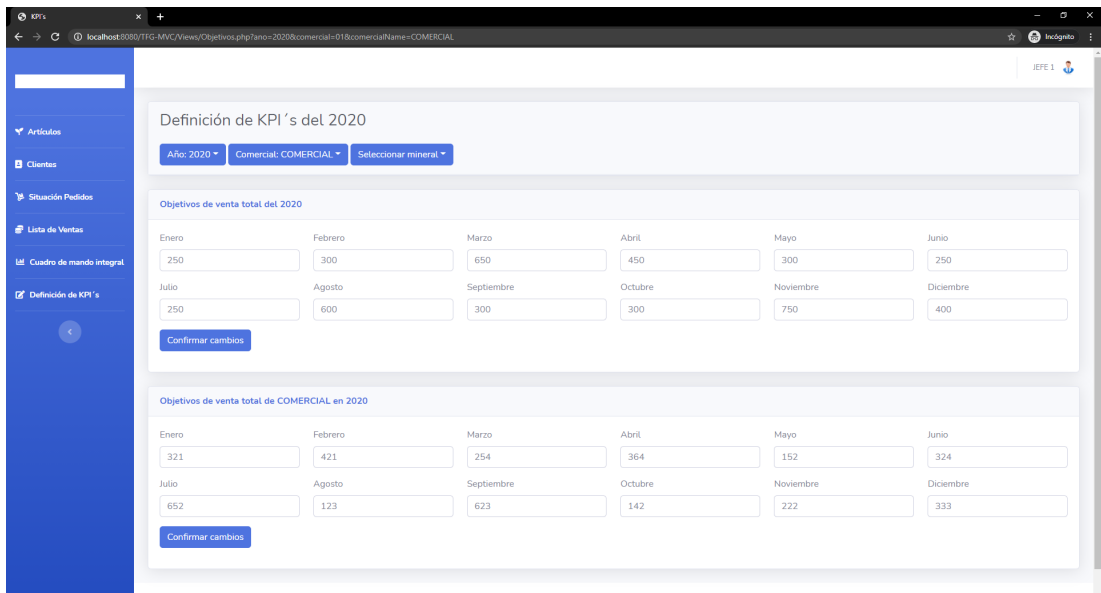


Situación de Pedidos

Selecciónar Ruta: **Todas** Nombre Cliente:  **Refrescar**

Fecha	Descripción	Cantidad	X	Pendiente	Comentarios
<b>Cuenta 2 Ruta:</b>					
30/09/20	Guidoni Quartz Autumn 2a Tablero Pulido 2 cm.	5	ud	5	
05/11/20	Guidoni Quartz Belgian Blue 2a Tablero Pulido 2 cm.	2	ud	2	
12/11/20	Granito Azul Platino Tablero Pulido 2 cm.	3	ud	3	
Total Cliente 2		10		10	
<b>Cuenta 3 Ruta:Tomño</b>					
01/10/20	LAMINAM Collection Fumo 3000x1000 mm Lucidato 5 mm	8	ud	8	
07/10/20	Granito Gris Alba Tablero Granallado+cepillado 1.5 cm.	2	ud	2	
27/10/20	Granito Albero Tablero Apomazado 2 cm.	1	ud	1	
Total Cliente 3		11		11	
<b>Cuenta 4 Ruta:</b>					
19/11/20	Acabado : Sat + EV	3	ud	3	
Total Cliente 4		3		3	
<b>Cuenta1 Ruta:</b>					
29/07/20	Granito Albero Tablero Apomazado 2 cm.	1	ud	1	
29/07/20	Granito Grisal Tablero Pulido 10 cm.	3	ud	3	
26/09/20	LAMINAM Corton Grey Tablero Natural 06 MM	3	ud	3	
07/11/20	Granito Albero Tablero Apomazado 2 cm.	1	ud	1	
07/11/20	Granito Labrador oscuro Tablero Pulido 10 cm.	3	ud	3	

Figura 9.11: Vista de la situación de los pedidos.



Definición de KPI's del 2020

Año: 2020 \* Comercial: COMERCIAL \* Seleccionar mineral \*

Objetivos de venta total del 2020

Enero	Febrero	Marzo	Abril	Mayo	Junio
<input type="text" value="250"/>	<input type="text" value="300"/>	<input type="text" value="650"/>	<input type="text" value="450"/>	<input type="text" value="300"/>	<input type="text" value="250"/>
Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
<input type="text" value="250"/>	<input type="text" value="600"/>	<input type="text" value="300"/>	<input type="text" value="300"/>	<input type="text" value="750"/>	<input type="text" value="400"/>

**Confirmar cambios**

Objetivos de venta total de COMERCIAL en 2020

Enero	Febrero	Marzo	Abril	Mayo	Junio
<input type="text" value="321"/>	<input type="text" value="421"/>	<input type="text" value="254"/>	<input type="text" value="364"/>	<input type="text" value="152"/>	<input type="text" value="324"/>
Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
<input type="text" value="652"/>	<input type="text" value="123"/>	<input type="text" value="623"/>	<input type="text" value="142"/>	<input type="text" value="222"/>	<input type="text" value="333"/>

**Confirmar cambios**

Figura 9.12: Vista donde se definen los KPIs.

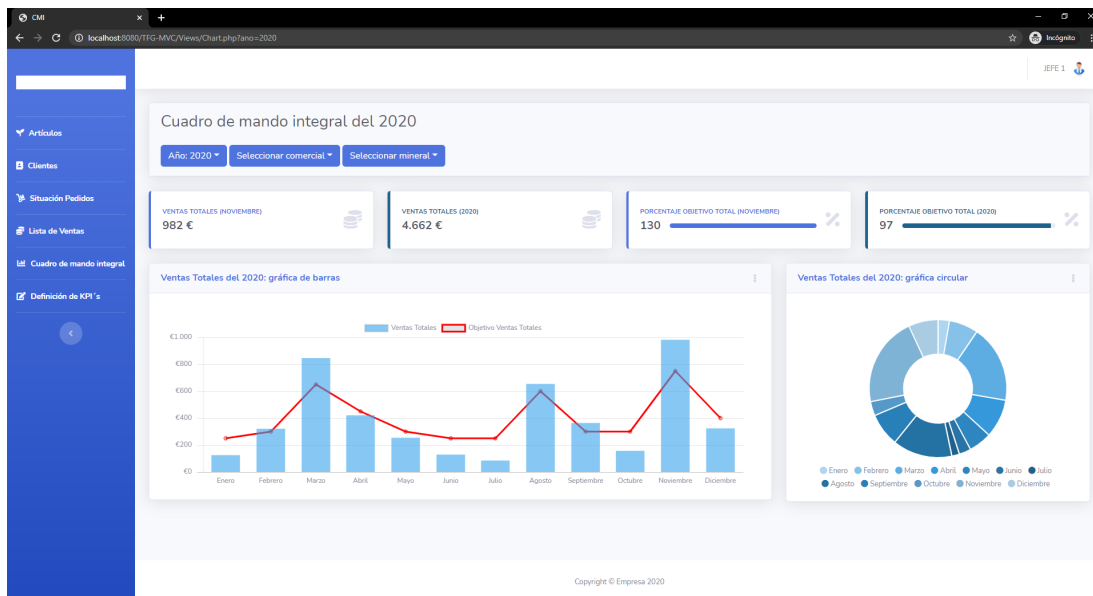


Figura 9.13: Vista del Cuadro de Mando.

## Capítulo 10

# Pruebas

---

En este capítulo se exponen las distintas pruebas que se han desarrollado en todo el proyecto. Al finalizar cada iteración, se han realizado una serie de pruebas para comprobar que el desarrollo realizado cumple los requisitos marcados. Sin el resultado positivo de estas pruebas no se puede continuar con la elaboración de la siguiente fase.

Las pruebas software sirven para descubrir la presencia de errores y ganar la máxima confianza de que se cumple con la especificación. Es importante averiguar los errores que pueda tener la aplicación lo antes posible, por eso se deben hacer pruebas en cada iteración. En este TFG, se han realizado pruebas unitarias, de integración, de sistema y de aceptación.

### 10.1 Pruebas unitarias

Se encargan de validar el correcto funcionamiento de un módulo de código. Asegurando que cada uno funcione correctamente por separado.

Se han realizado para todas las funcionalidades desarrolladas en cada iteración, comprobando que tienen el comportamiento deseado. Para eso, se han creado pruebas a todos los procesos. Examinando las posibles entradas y salidas para cada uno. Gracias a ello se han detectado algunos errores a la hora de insertar datos a la base de datos. Una vez resueltos estos errores se ha podido continuar con las siguientes pruebas.

### 10.2 Pruebas de integración

Las pruebas de integración, sirven para verificar que todos los subsistemas encajan correctamente y funcionan juntos. Se debe validar que la navegación de la información entre los módulos se realizan de la forma correcta.

Se comprueba que todas las llamadas entre los controladores y vistas funcionan correctamente, al igual que el intercambio de datos entre la capa modelo y controlador. Además,

se ha comprobado que la información con la que se trabaja en el ERP personalizado llega correctamente a la aplicación web. Debido a que este proyecto forma parte de un sistema mas completo.

### **10.3 Pruebas de sistema**

Son las que validan el correcto funcionamiento de todo el sistema. Se debe verificar que el desarrollo de la aplicación cumple con los requisitos acordados con el cliente.

Para la realización de estas pruebas se ha comprobado que un resultado esperado es devuelto por el sistema. Estas pruebas se basan en la metodología de pruebas de caja negra. Han sido realizadas junto con el jefe de proyecto, para cerciorarse de que todo el sistema funciona correctamente.

Son las últimas pruebas de desarrollo que se elaboran de la aplicación.

### **10.4 Pruebas de aceptación**

Se utilizan los casos de uso más relevantes para demostrar el funcionamiento de la versión a desplegar. Las realiza el propio usuario de la aplicación.

Para estas pruebas se ha implantado el software en la empresa, para que los comerciales y administradores puedan comprobar que se cumple con el funcionamiento esperado. Tras tres días de prueba, la aplicación ha tenido una buena aceptación por parte de los comerciales. Destacando sus procesos intuitivos y cómodos. Además, los administradores han destacado la representación gráfica de los datos de la empresa.



# Conclusiones

---

En este capítulo se realiza una reflexión sobre todo el proyecto. Detallando los distintos objetivos que se han cumplido en su desarrollo. También se comentan las lecciones aprendidas.

### 11.1 Conclusiones

Tras finalizar con todo el desarrollo del proyecto, se puede concluir que se han cumplido todos los requisitos fijados inicialmente. La aplicación web creada, obtiene la trazabilidad comercial y gestión administrativa que demandaba la empresa.

En el proyecto se obtiene la información completa de la situación económica de un cliente, sabiendo en todo momento los riesgos que se llevan a cabo al realizarle una venta. Es de vital importancia tener asegurado el cobro de todos los pedidos que se realizan, por eso es fundamental tener esta información. Además, se extrae toda la información de las ventas de un cliente, pudiendo así preparar una buena estrategia de venta.

Asimismo, el sistema permite generar los pedidos de un cliente tras navegar por el catálogo de productos. Hay artículos únicos, que tienen unas características que los diferencian de cualquier otro. Para estos, la aplicación web da la posibilidad de reservarlos para un cliente, estando prohibida su venta a cualquier otra persona. Igualmente, se permite saber la situación de todos los pedidos.

Toda la información relacionada con los clientes, pedidos y ventas se muestra de forma personalizada. Un comercial sólo podrá ver la información relacionada con sus clientes y sus ventas; mientras que un administrador dispone de toda la información que desee.

Finalmente, la aplicación web permite configurar los distintos KPIs, que van a estar acompañados de una representación gráfica de los datos para mayor facilidad de comprensión. Gracias a esta funcionalidad, los administradores van a poder realizar mejores estrategias de venta y toma de decisiones.

Con el diseño de la interfaz de la aplicación web se ha intentado conseguir un funciona-

miento fácil e intuitivo. Algo que ha sido respaldado por los usuarios de la aplicación en su periodo de prueba. Además de seguir un diseño adaptable, para garantizar su correcta visualización en los distintos dispositivos, como son las tabletas que utilizan los comerciales.

## 11.2 Lecciones aprendidas

Gracias a la realización de este TFG, he adquirido conocimientos sobre lo que supone realizar un programa para un entorno real. En el inicio del proyecto, me he encontrado con muchos conceptos que desconocía, tanto relacionados con las actividades de venta y facturación, como con el sector de la piedra.

También he conseguido aprender la importancia del trato con el cliente, sabiendo en primera persona como son las comunicaciones con ellos. Es extremadamente importante, saber bien qué es lo que quiere el cliente y cómo lo quiere desde el principio, para no tener que estar modificando las funcionalidades creadas.

En resumen, he aprendido a enfrentarme a una situación real, donde he cometido errores que me servirán para no cometerlos en el futuro.

## 11.3 Líneas futuras

Tras ver el resultado de la aplicación, y realizar pruebas durante unos días, el cliente ya piensa en distintas funcionalidades que se podrían añadir.

Una de ellas sería disponer de una galería gráfica de cada piedra, dando la opción de añadir distintas fotos en los productos del catálogo.

Otra posibilidad, sería diseñar una planificación de visitas de cada comercial. Sería interesante que el administrador tuviera una trazabilidad de las visitas de cada comercial. Registrando un historial de visitas y un planificador donde poder añadir citas en un calendario.

Estas dos ideas quedan pendientes para su estudio y posible incorporación futura.

# Lista de acrónimos

---

**DAO** Data Access Object. i, 18

**E/R** Entidad-Relación. 16, 31

**ERP** Enterprise Resource Planning. 1, 2, 27, 31

**FK** Foreign Key. 31–33

**KPI** Key Performance Indicator. 1, 2, 25, 28, 38, 51

**MVC** Model-View-Controller. i, 16, 17, 37, 38, 45

**PK** Primary Key. 31–34

**TFG** Trabajo de Fin de Grado. 1, 2, 21, 25, 31, 59, 62

**UML** Lenguaje Unificado de Modelado. 16

**VO** Value Object. i, 17



# Bibliografía

---

- [1] Importancia departamento comercial. (consultado: 02.11.2020). [En línea]. Disponible en: <https://www.euroinnova.edu.es/blog/gestion-comercial>
- [2] Definición de kpi. (consultado: 21.10.2020). [En línea]. Disponible en: <https://blog.es.logicalis.com/analytics/kpis-qué-son-para-qué-sirven-y-por-qué-y-cómo-utilizarlos>
- [3] Sage erp. (consultado: 14.10.2020). [En línea]. Disponible en: <https://www.sage.com/es-es/erp/>
- [4] Mycrosoft dynamics 365. (consultado: 14.10.2020). [En línea]. Disponible en: <https://dynamics.microsoft.com/es-es/what-is-dynamics365/>
- [5] Odoos. (consultado: 14.10.2020). [En línea]. Disponible en: [https://www.odoo.com/es\\_ES/](https://www.odoo.com/es_ES/)
- [6] Características windows 10. (consultado: 10.11.2020). [En línea]. Disponible en: <https://www.microsoft.com/es-es/windows/features>
- [7] Apache netbeans. (consultado: 10.11.2020). [En línea]. Disponible en: <https://netbeans.apache.org/>
- [8] Git. (consultado: 10.11.2020). [En línea]. Disponible en: <https://git-scm.com/>
- [9] Tortoisegit. (consultado: 10.11.2020). [En línea]. Disponible en: <https://tortoisegit.org/>
- [10] Mysql. (consultado: 10.11.2020). [En línea]. Disponible en: <https://www.mysql.com/>
- [11] Definición de metodología. (consultado: 05.11.2020). [En línea]. Disponible en: <https://obsbusiness.school/es/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software>
- [12] Definición de planificación de proyecto. (consultado: 08.11.2020). [En línea]. Disponible en: <https://www.unir.net/empresa/desarrollo-directivo/estrategia-de-negocio/planificacion-proyecto/>

- [13] Xix convenio colectivo del sector de empresas de ingeniería y oficinas de estudios técnicos, boe 2019. (consultado: 08.11.2020). [En línea]. Disponible en: [https://www.boe.es/diario\\_boe/txt.php?id=BOE-A-2019-14977](https://www.boe.es/diario_boe/txt.php?id=BOE-A-2019-14977)
- [14] Definición de requisito. (consultado: 08.11.2020). [En línea]. Disponible en: <https://www.significados.com/requisito/>