

Manager Societies

Desarrollo Web y Análisis de Mercado de un ERP para PYMES

TRABAJO FIN DE GRADO

Apellidos y nombre: Barrio Martín. Javier

D.N.I.: 53852636E

Correo electrónico: j.barrio.2016@alumnos.urjs.es

Directora: Gema Gutiérrez Peña

Grado en Ingeniería de Software

Grupo: Móstoles

Curso: 2021/2022 – convocatoria: noviembre

Resumen:

Se conoce el término de emprendimiento como la actitud y aptitud que toma un individuo para iniciar un nuevo proyecto a través de ideas y de oportunidades. Este término, tiene su origen junto a la primera aparición registrada del ser humano caracterizado en asumir riesgos con el objetivo de generar oportunidades de crecimiento. Como analogía usaremos la huida del ser humano de la seguridad de las cuevas buscando un mejor hogar.

La propia definición de emprendimiento hace referencia a la existencia de riesgos, incrementándose en el caso de los emprendedores españoles, conocidos en la actualidad como PYMES o pequeños autónomos.

Los problemas de financiación, el adelanto del IVA, el pago de una cuota fija y la liquidez son algunos de los mayores problemas enfrentados por estos [1].

Estos contratiempos ocasionan un nuevo tipo de incertidumbre que suelen ir acompañados de problemas de índole personal, estrés y una sensación de riesgo continua. El origen del riesgo suele venir originado en un problema de gestión empresarial inicial debido en su gran mayoría a la falta de formación previa de nuestros autónomos en cuanto a lo que finanzas y marketing se refiere. Una forma de solventarlo, o por lo menos reducirlo, viene por la gestión de los recursos de la empresa de forma eficiente y ordenada.

La solución a dicho problema se puede encontrar en varias soluciones eficientes, pero fuera de mercado para la mayoría de los emprendedores del sector que venimos a tratar. Los autónomos y pequeñas PYMES no disponen de un capital que les permita hacerse con aplicaciones de gestión de recursos.

Por ello, se plantea diseñar una aplicación eficiente y que permita a este tipo de cliente gestionar de manera global su empresa. La idea de este proyecto nace mientras ayudaba en un comercio familiar, analizando como las paredes estaban llenas de librerías, y estés de archivadores repletos de facturas, albaranes, nóminas... Todo ello disperso en documentos de texto o en formato pdf impresos.

Mi brillante solución fue iniciar una búsqueda de un programa o aplicación que facilitará la vida de mis congéneres a un bajo precio o por lo menos, uno que fuera asequible. Mi experiencia fue darme de bruces con programas antiguos, incompletos, poco atractivos o, en el mejor de los casos, si cumplía los requisitos mínimos a un precio demasiado elevado.

Por ello, decidí que mi trabajo de fin de grado debía ser una aplicación que buscara solucionar dicho problema, tanto para mis familiares como para todo aquel que disponga de la necesidad de tener una aplicación para gestionar su empresa.

En mi opinión, considero que cualquier autónomo o PYME que la viera, sentiría cierto interés en analizar en detalle.

ÍNDICE

1. Introducción:	5
1.1 Descripción del Problema:	5
1.1.1. Definición de un Enterprise Resource Planning:	5
1.1.2. Ventajas del Uso de un ERP:	5
1.1.3. Desventajas del Uso de un ERP:	6
1.1.4. ERP como Aplicación Web:	6
1.1.5. Implantación de un ERP:	7
1.2 Objetivos:	7
1.2.1. Objetivo del Trabajo:	7
2. Estado del Arte:	8
2.1. Definición de Análisis de Mercado:	8
2.2. Identificación del Mercado:	8
2.3. Análisis PEST:	9
2.4. Consumidores Potenciales:	10
2.5. Estudio de la Competencia:	11
3. Metodología:	17
3.1. Kanban:	17
3.2. Scrum:	18
3.2.1. Sprint 1:	19
3.2.2. Sprint 2:	20
3.2.3. Sprint 3:	20
3.2.4. Sprint 4:	21
3.2.5. Sprint 5:	22

3.3.	Trello:	23
3.4.	Git-Hub:	26
3.5.	README:	27
3.6.	Control de Versiones:	27
4.	Tecnologías y Herramientas Utilizadas:	27
4.1.	Git:	27
4.2.	UML:	27
4.3.	Modelo Vista Controlador:	28
4.4.	HTML:	29
4.5.	CSS:	29
4.6.	Bootstrap:	30
4.7.	JavaScript:	30
4.8.	JQuery:	31
4.9.	Maven:	31
4.10.	Java:	31
4.11.	Spring Boot:	32
4.12.	Thymeleaf:	32
4.13.	MySQL:	32
4.14.	Jpa Repository:	32
5.	Diseño y Desarrollo:	33
5.1.	Diseño:	33
5.1.1.	Diagrama de Componentes:	33
5.1.2.	Maquetación:	33

5.2. Desarrollo Web:	34
5.2.1. Migración a Aplicación Web:	34
5.2.2. Aplicación:	36
5.2.3. Controlador de Aplicación:	36
5.2.4. Inicio de Sesión:	36
5.2.5. Información del Usuario:	40
5.2.6. Creación de Usuario:	43
5.2.7. Agenda:	44
5.2.8. Calendario:	50
5.2.9. Inventario:	53
5.2.10. Tareas:	59
5.2.11. Migración a Base de Datos:	63
5.2.12. Dashboard:	65
5.2.13. Facturación:	68
5.2.14. Finanzas:	75
6. Conclusiones y Trabajos Futuros:	78
7. Bibliografía:	79
8. Anexos:	81

1. Introducción:

1.1 Descripción del Problema:

El primer punto del proyecto recoge el concepto de *Enterprise Resource Planning*, se parte de la definición del concepto para tratar sus ventajas, desventajas, su versión web y su implantación.

1.1.1. Definición de un Enterprise Resource Planning:

En este apartado se va a definir ERP y analizar las necesidades y problemas de este. El término ERP o *Enterprise Resource Planning* [2] hace referencia a un sistema de planificación de los recursos de una empresa. Un ERP bien diseñado e implementado debe hacerse cargo de las operaciones internas de una empresa.

Entendemos como operaciones internas aquellos negocios asociados con las operaciones de producción, aspectos de distribución y producción de bienes o servicios. Un ERP debe ser un sistema especializado que permita la unificación y organización de todas las áreas, es decir, que permita la trazabilidad de todos los procesos, y, por tanto, cumple uno de los objetivos más codiciados del siglo XXI: planificar la optimización de los recursos.

El término planificación de recursos empresariales proviene de la unión de la planificación de recursos de manufactura, en inglés *Manufacturing Resource Planning*, o MRPII, y de la planificación de requerimientos de material, en inglés *Material Requirements Planning*, o MRP.

De forma típica un ERP gestiona únicamente producción, logística, inventario, distribución y contabilidad, todo ello de forma modular. Sin embargo, un buen software ERP podría intervenir en el control de muchas más actividades de negocios.

Análogamente podemos decir que un ERP es la trastienda de una empresa. Los clientes acuden a la tienda, compran el producto y se van, lo que no visualizan es el proceso que ha llevado a la materia prima a convertirse en ese producto final. De forma similar, un ERP contiene y controla toda la información de la empresa en una especie de sombra, solo visible para los usuarios que se deseen que accedan a ella.

1.1.2. Ventajas del Uso de un ERP:

En el mercado existen muchas soluciones software cerradas para gestionar ciertas funciones empresariales, como por ejemplo el uso de un gestor de clientes, en inglés *Customer Relationship Management* o CRM, el uso de *Business Intelligence* o incluso paquetes de oficina como Office. Por ello, muchos empresarios eligen estas funciones por su coste más reducidos y porque creen que resuelven sus necesidades.

Sin embargo, un diseño en función a las necesidades integrado puede proporcionar resultados más eficientes y sobre todo una mayor optimización del negocio en cuestión. Para los empresarios conservadores la implantación de un ERP supone un gran conflicto en sus negocios debido a la gran inversión inicial que esto supone, pese a esto, según un estudio [3] de Robit Kenge en 2020 se verificó un aumento del 56% de la productividad de las empresas con un ERP integrado.

Un ERP bien diseñado nos proporciona un sinfín de procesos automatizados, así como la posibilidad de disponer de la totalidad de la información de la empresa en una misma plataforma modular. Se debe considerar el uso de *Business Intelligence* como mecanismo de realización de informes sobre el estado actual de la empresa tomando la información del ERP como datos de entrada.

Por otro lado, los ERP modernos contemplan la integración de un CRM para la gestión de clientes y sobre todo un control de calidad, que nos asegura que no existen problemas no identificados o solventados en los productos finales a entregar.

1.1.3. Desventajas del Uso de un ERP:

Como se ha mencionado con anterioridad el mayor inconveniente de la implantación de un ERP es su coste, esto engloba desde el coste de personalización específico para cada empresa como la posible compra de hardware e infraestructuras necesarias para el correcto funcionamiento de este.

En el mundo empresarial los ERP poseen la visión externa de ser sistemas muy rígidos, se cree que son difícilmente adaptables a un flujo específico de un grupo de trabajadores y del proceso de negocios de algunas compañías. Se debe realizar un análisis sobre si dicho flujo es realmente incompatible por su especificación; o por las malas prácticas heredadas tanto de trabajadores como de los directivos de una determinada empresa.

Por el contrario, un sistema ERP puede ser complejo para empleados no formados. Al coste inicial del desarrollo e integración se le suma el coste del entrenamiento de operarios, sufriendo el sistema cuellos de botella si alguno de los departamentos que utilizan el sistema no lo utilizan de forma eficiente.

1.1.4. ERP como Aplicación Web:

Una posible solución para evitar el gasto infraestructural y la compra de equipamiento es la implantación de un ERP en la nube, es decir, un ERP ejecutado en una aplicación web. En los últimos años la implantación de un ERP Web [4] ha ganado una gran popularidad, el coste de esta se reduce al pago para el acceso a un servidor y el coste de la aplicación, servicio y mantenimiento, ya que, se da por hecho la presencia de ordenadores con conexión a internet en las empresas actuales.

Según un estudio [5] realizado a 832 empresas en 2012 por el grupo Aberdeen el 34% de ellas poseía un ERP obsoleto de entre 7 y 15 años de antigüedad. Según se analizó, el principal motivo de que sucediera tal obsolescencia se debía al coste de actualización. El uso de un ERP web nos permite actualizar de forma automática nuestro producto sin coste adicional otorgando a los usuarios el software más reciente en todo momento.

Por otro lado, un servidor web es accesible desde cualquier lugar geográfico y temporal, siempre que dispongamos de un dispositivo con conexión a internet. Esto genera beneficios, pero también la existencia de un riesgo de caída del servidor o de una parada necesaria para el mantenimiento por parte del proveedor.

1.1.5. Implantación de un ERP:

El ERP se debe implantar en la empresa demandante, la implantación de un ERP se divide realizando, por un lado, un análisis previo donde se definirá el proyecto. Y por otro, los requisitos que debe cumplir nuestro software; finalmente, el diseño, donde se desarrollarán las funcionalidades principales y se realizará la migración del software antiguo si lo hubiese.

Tras el análisis y el diseño procedemos a la implementación, instalando el software y su puesta en marcha, quedando finalmente la post implementación donde ajustaremos los detalles y mejoraremos el sistema.

Uno de los graves problemas que se originan en el proceso de implantación de un ERP es la búsqueda por parte de la empresa demandante de reducir costes eligiendo la opción de bajo coste o eliminar la empresa integradora. La integración es imprescindible puesto que se debe entender la problemática de cada sector y definir secuencias de implementación sobre dicha problemática.

1.2 Objetivos:

La introducción del proyecto finaliza con una breve explicación de los objetivos de este.

1.2.1. Objetivo del Trabajo:

El objetivo del proyecto es el diseño e implementación de un ERP completo, exitoso y funcional enfocado a PYMES, pequeños y medianos autónomos. Con este objetivo se va a realizar un análisis del entorno para realizar un estudio de mercado y comprobar así la viabilidad del desarrollo de la aplicación.

Por un lado, se evaluará su posible éxito en el mercado y sector elegidos, esto se recoge en un estudio que valorará si el producto es atractivo para el cliente objetivo, el método elegido para esta comprobación será un análisis de mercado.

A lo largo del análisis de mercado nos enfocamos en la implementación de un ERP alojado en una web, no en su vertiente de escritorio.

2. Estado del Arte:

El estado del arte agrupa el análisis de mercado, esta agrupación incluye una breve definición de dicho análisis, la identificación del mercado objetivo, un análisis PEST completo, la búsqueda de los consumidores potenciales y, por último, el estudio de las ofertas que la competencia puede ofrecer a los clientes objetivo.

2.1. Definición de Análisis de Mercado:

Metafóricamente diremos que si la aplicación es nuestra casa soñada el análisis de mercado debe cumplir con el rol de los cimientos, es decir, un buen análisis de mercado es la base del éxito económico de cualquier negocio.

El análisis de mercado nos va a permitir saber si nuestra idea es realizable y sobre todo si es rentable, esto nos permite identificar rápidamente los gastos de la realización del proyecto y sobre todo evaluar si los beneficios son mayores a los gastos, y lo más importante, suficientes como para comenzar dicha aventura.

Definimos un análisis de mercado como un examen sistemático de un mercado y de sus condiciones. Debemos incluir la situación competitiva, nuestro público objetivo, las necesidades de dicho público, y las tendencias y/o acontecimientos que pueden influir en el mercado futuro.

Dentro de un análisis de mercado nos enfrentamos a cuatro campos diferenciados: Identificación del sector o mercado, análisis PEST, acrónimo de Político, Económico, Social y Tecnológico; consumidores potenciales y, por último, el estudio de la competencia. Dentro del análisis de la competencia usaremos un mapa de posicionamiento aproximado.

2.2. Identificación del Mercado:

Cuando se habla de identificar el mercado realmente se hace referencia a definición más exacta posible de nuestro mercado objetivo. La definición debe incluir la delimitación de mercados adyacentes que no van a formar parte de nuestro análisis inicial.

Para ello se realiza una segmentación del mercado, donde se dividirá en cuatro niveles bien diferenciados: Región, precio, productos y sectores de clientes.

A nivel geográfico el límite inicial es el territorio español, ya que, el diseño e implementación de la aplicación idiomas diferentes al castellano supone un sobrecoste, además, para utilizar la aplicación en otros países habría que tener en cuenta legislaciones extranjeras.

Dentro del factor de la edad debemos realizar una aplicación atractiva para un posible público joven por encima de los 20 años, pero con un diseño simple y sencillo para la población adulta. El enfoque elegido toma como molde un modelo que cumpla con un sector intergeneracional.

Estudiado el plan de apoyo a los autónomos de la comunidad de Madrid publicado en 2006, se obtiene que la edad media de los autónomos a fecha de la realización del estudio es de 44 años. [6] Por este motivo, se delimita una franja de edad entre los veinte y los cincuenta años. En paralelo, el ministerio de trabajo realizó un estudio en 2020 que obtuvo que el 73,62 % de los autónomos del país superaban los 40 años. [7]

Ante dichos resultados se plantea la siguiente reflexión, la población del llamado *Baby Boom* puede ser reacia y no poseer de un gran manejo de la tecnología, en contra posición, la *Generación X*, 1969-1980, están teniendo una gran resiliencia a la tecnología según un artículo del periódico *Two Worlds* [8]. El último sector, los *Millennials* tienen un gran manejo sobre ella.

Dentro del estudio realizado por el ministerio de trabajo se refleja que el 64,7 % de los autónomos en 2020 eran de género masculino, por lo que habrá que realizar un diseño neutro enfocado a dicho mercado.

Conociendo la primera información del cliente potencial se segmenta por sector laboral, se ofrece un ERP a una empresa de venta de servicios y productos, por lo que cualquier empresa que ofrezca algún tipo de estos servicios sería un cliente objetivo siendo óptimos los que realicen ambos.

Se ha analizado el precio medio del coste de una implantación de un ERP modular estandarizado [9], este, ronda los doce mil euros de inversión inicial. Para una empresa con procesos sencillos oscila entre doce mil y veinticinco mil euros, para empresas pequeñas como las que hacemos referencia podríamos encontrar soluciones más baratas, pero no por menos de seis mil euros de entrada. A esto, le debemos sumar el hardware necesario si no se dispone de él, una licencia anual, más el mantenimiento y servicio técnico.

Cabe mencionar el coste adicional de formaciones de personal y un servicio de consultoría externa. En secciones posteriores se visualizarán ejemplos para estos precios.

2.3. Análisis PEST:

Se ha valorado la elaboración de una análisis PESTEL, pero tras una valoración previa, se resumirá en un PEST, las letras E y L hacen referencia a restricciones medioambientales y legislaciones. En cuanto a las restricciones medioambientales no se visualiza relación con una aplicación web, las legislaciones se ubicarán dentro de la letra P, que referencia lo político.

Como **factores políticos** se toma el aumento impositivo que los distintos partidos políticos han tomado como estrategia para paliar la crisis económica, esto se refleja en el gasto fijo de autónomos y empresarios, que, en su eterna búsqueda de soluciones que reduzcan sus gastos u optimicen sus ingresos pueden ver en una aplicación ERP una solución atractiva.

Se visualizan diferentes **factores económicos** relevantes a tener en cuenta, en un primer lugar, una situación inflacionaria provocada por la inestabilidad geopolítica, de forma general, una inflación viene provocada por el aumento de la demanda de productos, esto a priori es algo positivo para un empresario, por contraposición nos encontramos con una inflación en el precio del coste de materias primas, por lo que aumentarían los gastos de creación de estos productos.

Unido a la subida del coste de materias primas se suele producir un aumento de crispación dentro de los empleados que, debido a la subida del precio de nivel de vida, provoca el deseo y demanda de subida del salario suponiendo un nuevo coste añadido. Esta situación se zanja con que el empresario en cuestión aumente el precio de sus servicios y productos para compensar la subida de los costes, indirectamente esto produce que la inflación siga aumentando.

Si aislamos la **situación geopolítica** mala actual, la tendencia insta a la optimización total en búsqueda de la obtención del máximo beneficio reduciendo los costes al mínimo, por ello, es tendencia la informatización y automatización de distintos procesos, uno de los pasos que aparecen es la instalación de un sistema de gestión empresarial en las empresas.

Aunque a priori puede parecer mal momento un periodo de crisis para la inversión de capital en una tecnología de uso interno, es al contrario uno de los momentos en los que se buscan más soluciones, si la empresa está en un buen momento económico se suele obviar o intentar no hacer grandes cambios, ahora bien, es cuando la economía aprieta cuando se intenta revertir dicha situación.

A **nivel social**, el ser humano ha desarrollado una necesidad en que la tecnología este diariamente reduciendo su trabajo, o por lo menos, ayudándole a que este sea más sencillo y llevadero. Por ello, encontramos tantas aplicaciones en el día a día, ya sea desde la realización de un listado para la compra de productos, o aquellos softwares internos que no podemos visualizar, pero aparecen en nuestros electrodomésticos, vehículos, etc. Por qué no, llevar eso no solo al ámbito personal, si no al ámbito profesional en nuestra propia empresa.

Pese a que la situación geopolítica ha provocado escasez de hardware o de elementos físicos, parece que podría ser un momento clave para invertir en software, algo que a priori es más fácil de mantener y menos costoso que incluso almacenar y gastar en material de oficina, como puede ser papel, toners, etc.

2.4. Consumidores Potenciales:

Como se ha visto durante el análisis de mercado los consumidores potenciales van a ser aquellos pequeños y medianos autónomos, además de PYMES con pocos trabajadores. Por otro lado, se hará hincapié en el colectivo en torno a los cuarenta y cinco años con empresas de servicios y venta de productos.

Dentro de este rango se agrupan diferentes tipos de clientes potenciales, habrá una parte de ellos que ya poseen algún sistema de gestión, que a su vez se dividen en tres subtipos, los que no están nada contentos con el sistema que poseen, los que poseen uno que no les entusiasma y valorarían un cambio. Y, finalmente, los que están encantados con su sistema.

De forma lógica donde se ha de incidir es en aquellos que tengan dudas, puesto que los que están hartos de su sistema se interesarán con cierta facilidad y celeridad, y en contraposición, los que aprecian su sistema se mantendrán recelosos a cambiarlo.

Adicionalmente nos interesan aquellas empresas que proporcionen muchos servicios y vendan, de media, una gran cantidad de productos, por lo que estarán más interesados en una herramienta que gestione todos esos movimientos.

Paradójicamente, obtienen gran relevancia aquellos autónomos y directores de PYMES que cuenten con menos de treinta y cuatro años, ya que, según un estudio de Rafael Schwarz [9], el 34 % de los usuarios entre dieciochos y treinta y cuatro años confía más en las herramientas tecnológicas, en la cercanía y la influencia en contraposición a una publicidad generalizada.

2.5. Estudio de la Competencia:

Leyendo el ranking realizado por tecnologías de la información conocemos la posición de los diez mejores ERPs del mercado para el año 2015 [10]. Encabezando la lista se encuentra SAP, con más de seis mil novecientos clientes para su suite ERP.

El análisis de estas aplicaciones ha sido verdaderamente complicado debido a las dificultades presentadas para encontrar una versión gratuita de prueba, ya que, en la mayoría de ellas no existía, y en otras debías contactar con el departamento comercial de dicha aplicación para la compra.

SAP se basa en un ERP Cloud similar al planteado, en su propia web SAP otorga información sobre sus cinco módulos [11]. El primer módulo trata sobre la planificación y análisis financiero, el segundo sobre el cierre contable, el tercero sobre la tesorería, cuarto sobre ahorro, facturación e ingresos, y por último, poseen un módulo de riesgos y ciberseguridad. Una licencia de IMPULSAP, licencia básica de no más de cinco usuarios con el software preconfigurado y entrenamiento en el uso y funcionamiento alcanza los treinta y cinco mil dólares [12], además del apoyo de asesor durante las dos semanas de implantación.

En el caso de necesitar hasta doce usuarios el precio asciende a cincuenta mil dólares, esto añade la capacidad de disponer de licencias SAP Business One y un equipo de asesores. En el caso de necesitar más de doce usuarios el precio superará estos cincuenta mil dólares con licencias completas y análisis de la empresa.

Tabla de estimación de precio de SAP Business One

IMPULSAP	PROYECTO SAP BUSINESS ONE COMPLETO	PROYECTO SAP BUSINESS ONE COMPLETO
Funcionalidades básicas	Proyecto personalizado	Para más de 12 usuarios
Ideal para pequeñas empresas con procesos y flujos sencillos.	Ideal para empresas en crecimiento.	Ideal para empresas en crecimiento con más de 12 usuarios.
\$35,000 USD	\$50,000 USD	+*50,000 USD
Licencias Starter: Software Preconfigurado y Entrenamiento.	Licencias Completas, Análisis, Configuración y Capacitación.	Licencias Completas, Análisis, Configuración y Capacitación.
Implementación por un Ingeniero en Sistemas	Implementación por un Equipo de profesionales.	Implementación por un Equipo de profesionales.
Implementación en 2 semanas	Implementación en 8-12 semanas	Implementación en 12-16 semanas
Financiamiento hasta 12 meses	Financiamiento hasta 12, 34, 36 y hasta 48 meses	Financiamiento hasta 12, 34, 36 y hasta 48 meses
Hasta 5 usuarios	De 6 a 12 usuarios	Ilimitado
No requiere consultoría	Consultoría en tus oficinas	Consultoría en tus oficinas
Entrenamiento en Avance	Capacitación en tus oficinas	Capacitación en tus oficinas
<input checked="" type="checkbox"/> Incluido	<input checked="" type="checkbox"/> Incluido	<input checked="" type="checkbox"/> Incluido
Habla con un asesor >	Habla con un asesor >	Habla con un asesor >

Ilustración 1 - Tabla Estimación SAP

Visualmente, el software SAP Cloud posee un selector de navegación superior donde aparece el logo de SAP y enlaces a cada módulo, además de una sección de búsqueda, alertas, tareas y el perfil del usuario. En cuanto al Dashboard aloja demasiada información, de forma poco clara e intuitiva, además, la paleta de colores utilizada dificulta su uso, es demasiado monótono.

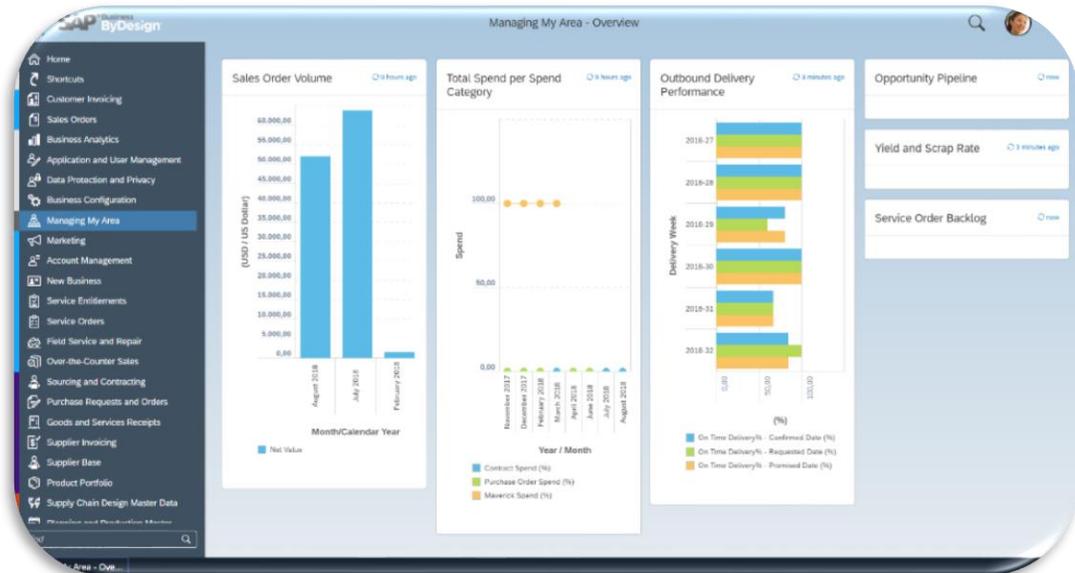


Ilustración 2 - Dashboard SAP

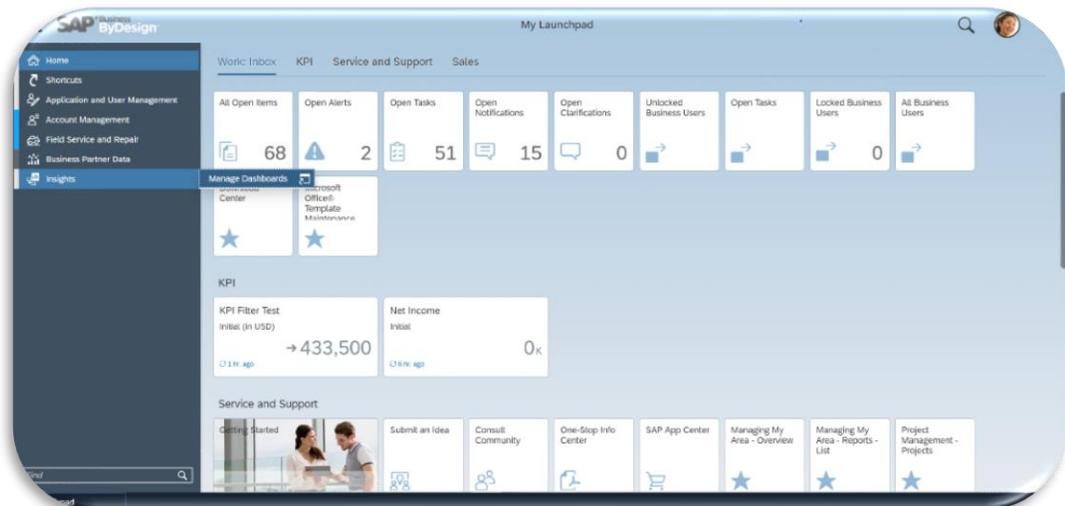


Ilustración 3 - Dashboard SAP

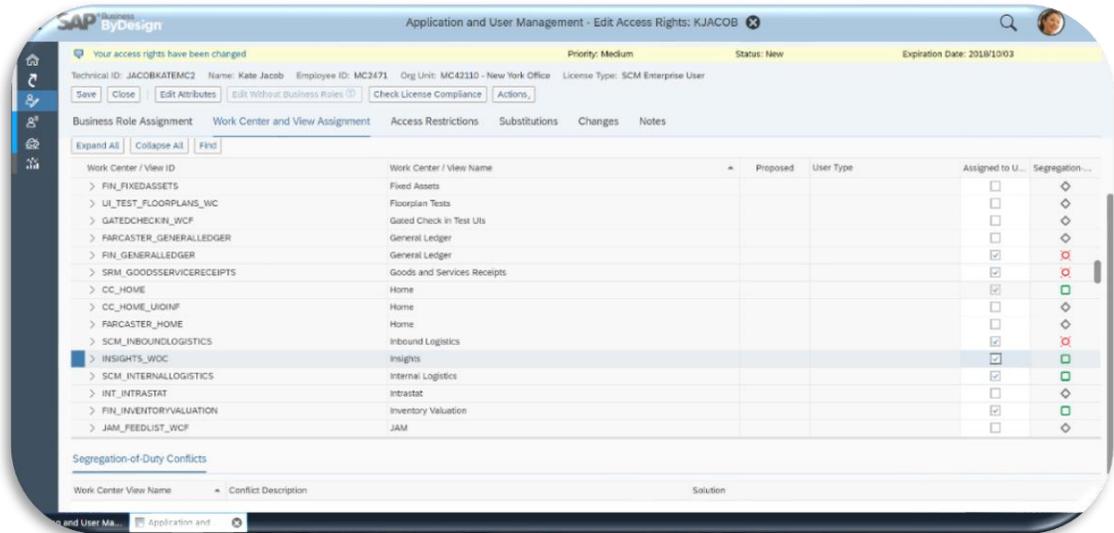


Ilustración 4 - Módulo de SAP

En el tercer lugar tenemos Oracle ERP Cloud R13 [13], un ERP ubicado en la nube, presenta tres planes diferentes, un módulo básico financiero con un precio de ciento setenta y cinco euros mensuales por usuario; para añadir el módulo avanzado de finanzas se añadirán ochenta euros mensuales por usuario. Para obtener todas las funcionalidades se añaden ciento cincuenta euros mensuales más alcanzando los trescientos cincuenta euros mensuales por usuario al mes o tres mil seiscientos sesenta euros al año por usuario.

El tiempo para implantarse se establece entre los seis y los nueve meses. Se ofrece una prueba gratuita, pero en el caso de desear la versión completa se deben abonar más de cincuenta mil euros más el número de usuarios deseados.

En cuanto al diseño, la distribución es similar al ERP de SAP, con una estética poco formal y un diseño que debería ser más cuidado. Los elementos están apilados en un mismo punto, de forma que no se aprovecha el espacio disponible. Posee funcionalidades variadas y complejas que permiten una buena gestión empresarial, dejando en segundo plano el diseño, que facilitaría la usabilidad de usuario dentro de la aplicación.

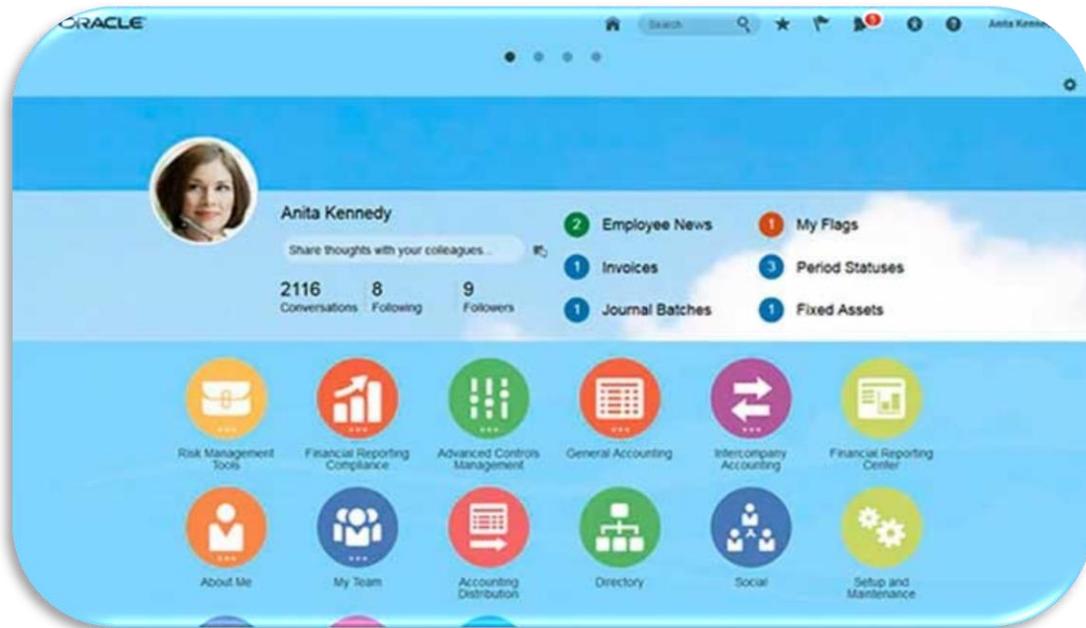


Ilustración 5 - Dashboard Oracle ERP

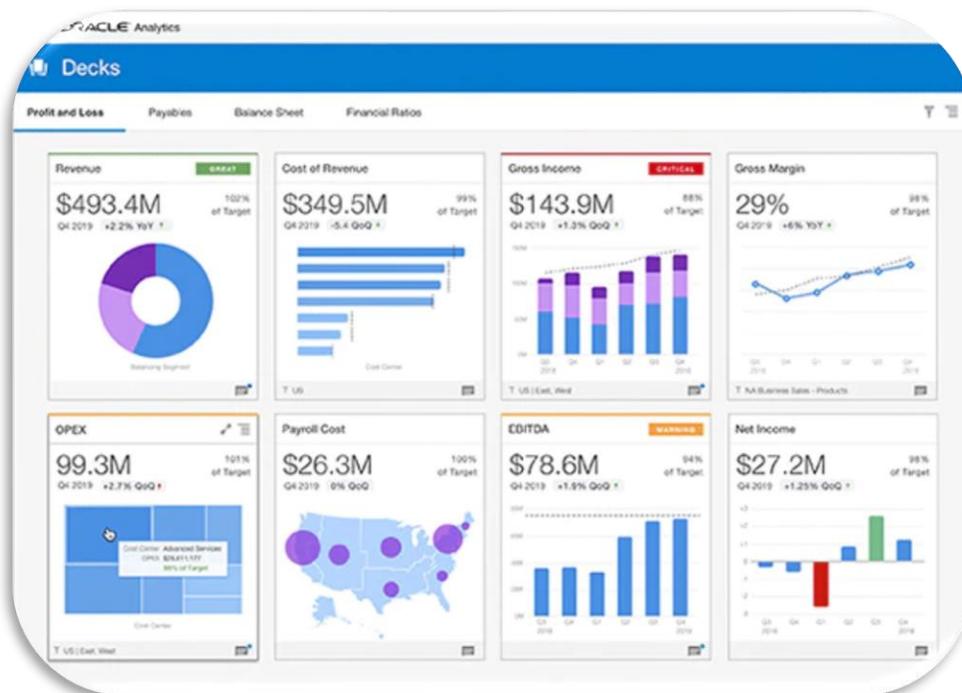


Ilustración 6 - Review Transaction Oracle ERP

En quinta posición encontramos el ERP de Intuit [14], con tres ediciones disponibles, ninguna en la nube, la opción más barata cuesta mil setecientos cuarenta dólares anuales, con acceso para treinta usuarios, pero sin control de inventario. La opción intermedia con valor de dos mil ciento cuarenta dólares por año permite treinta usuarios y control de inventario. Por último, tenemos una opción de cuatro mil doscientos dólares anuales que añade un CRM y control de nóminas para cuarenta usuarios.

Nos encontramos con una aplicación de escritorio, el diseño recuerda a aplicaciones diseñadas en versiones antiguas de Windows, como las anteriores es una aplicación modular. La versión premium cumple las funcionalidades esperadas, aunque las dos más económicas se pueden quedar un poco cortas en funcionalidad. El diseño no es atractivo, ni permite una usabilidad fácil para un autónomo medio.

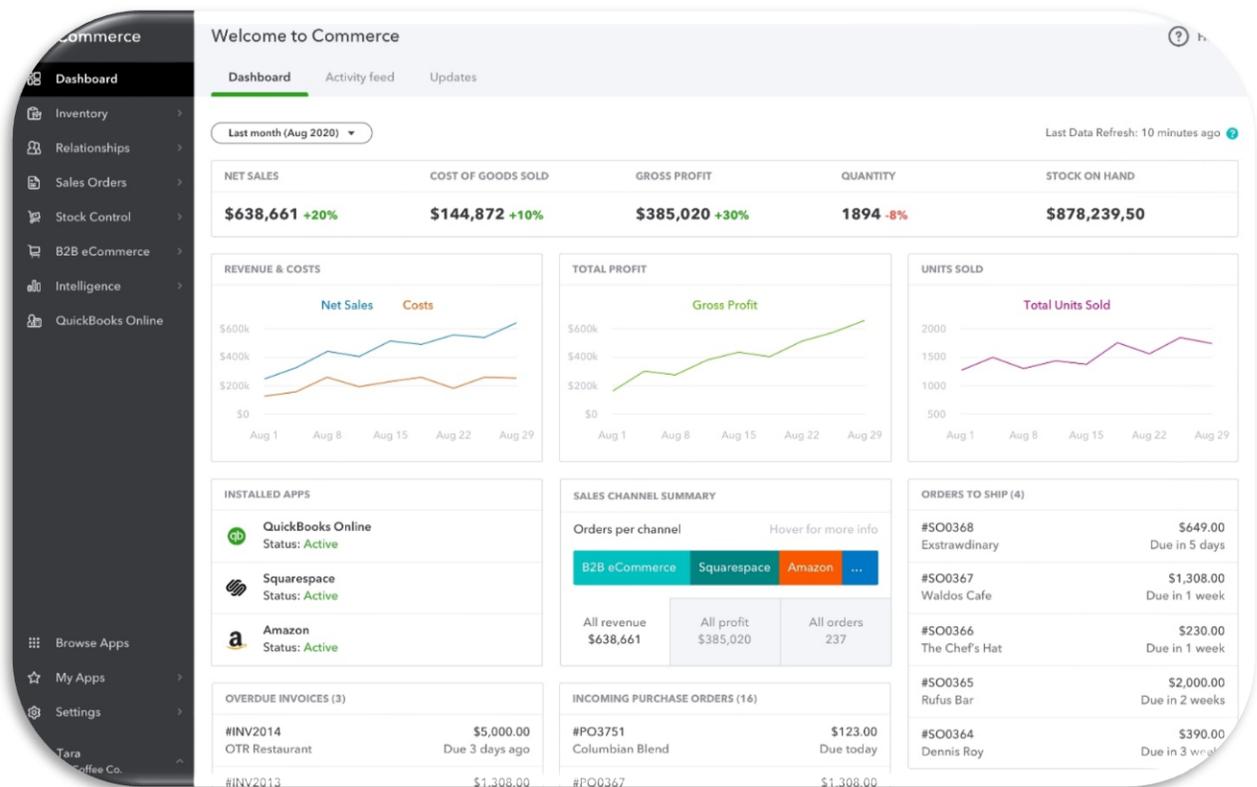


Ilustración 7 - Intuit Dashboard

Se concluye que las tres opciones estudiadas son excesivamente caras para implementarse en autónomos o empresas emergentes, además, no poseen un diseño atractivo, ni sencillo para clientes promedio.

Por ello, el proyecto se centra en una herramienta que cumpla todas las necesidades que se pueda encontrar un autónomo promedio en la gestión diaria de su empresa. Además, debe ser atractiva visualmente para facilitar y favorecer un ambiente de trabajo ideal. Por otro lado, debe tener un diseño sencillo y claro.

Se ha agrupado la información recalcada en un mapa de posicionamiento, en él podemos ver de forma visual un gráfico entre el precio y la calidad de cada ERP estudiado.

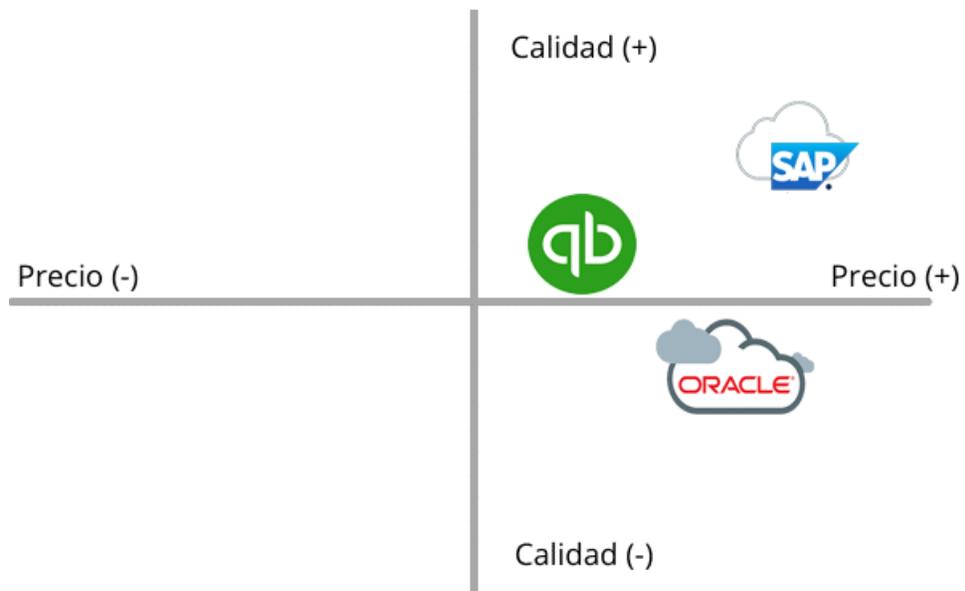


Ilustración 8 - Mapa de Posicionamiento

Como se observa en el mapa de posicionamiento las tres aplicaciones estarían por encima del precio medio, y pese a que dos de ellas están por encima de la medida de calidad, no tienen un ratio calidad precio adecuado.

Solo SAP podría estar en un precio calidad aceptable, pero es un software enfocado a grandes empresas que aparte de un precio desorbitado, es demasiado técnico para el público que se esperaba como cliente potencial.

3. Metodología:

A lo largo del tercer punto se explicará la metodología utilizada durante el proyecto.

3.1. Kanban:

Kanban [15] es un método de gestión del flujo de trabajo para definir, gestionar y mejorar los sistemas de trabajo. Su origen proviene de la industria de fabricación, pero se ha llegado a convertir en uno de los métodos de trabajo ágiles más famoso y utilizado por equipos de desarrollo de software.

El término Kanban proviene del japonés, y su significado es *tablero visual*, se viene utilizando desde la década de los años cincuenta. A principios del siglo XXI surge, además, el método Kanban, en su origen se resumía en un tablero con tres columnas básicas: solicitado, en curso y hecho; sirviendo como un depósito de información en tiempo real, permitiendo identificar fácilmente cuellos de botella, si los hubiese.

A la hora de hablar de los principios de la metodología Kanban debemos explicar dos tipos y seis prácticas. Los tipos son *Principios de la Gestión del Cambio* y *Principios de prestación de servicios*.

Dentro de ambos tipos nos encontramos con tres buenas prácticas, en el caso del primero nos encontramos con *Comienza con lo que haces ahora*, *Acepta el cambio incremental y evolutivo* y *Fomenta los actos de liderazgo a todos los niveles*.

En cuanto al otro tipo aparece *El centro sobre las necesidades y expectativas del cliente*, *Gestión del trabajo, no de los trabajadores* y *Revisión periódica de la red de los servicios*.

Aunque Kanban es una metodología muy amplia no se abordará cada principio y práctica en detalle, puesto que ha sido una herramienta y no el tema principal del proyecto, para finalizar se exponen seis ventajas que presenta Kanban frente a otras metodologías.

Kanban nos da mayor visibilidad del flujo alineando los objetivos con la ejecución de estos, además mejora la velocidad de entrega, la previsibilidad, la gestión de dependencias y la satisfacción de los clientes.

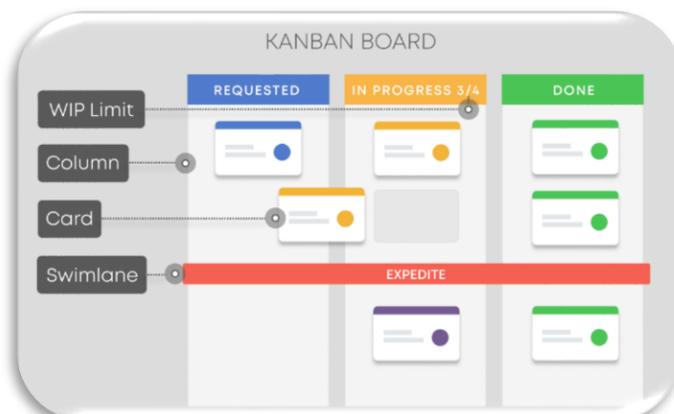


Ilustración 9 - Ejemplo Básico de Tablero

3.2. Scrum:

Aunque Scrum [16] es un marco que permite un trabajo colectivo entre equipos, se ha considerado heredar ciertas características para el desarrollo del proyecto. Pese a que no hay un equipo como tal puesto que la realización del proyecto será llevada a cabo por una sola persona, está, sí que ocupará ciertos roles característicos del marco de trabajo.

Además, no solo ocupará los roles si no sus funciones y se aprovechará de la metodología Scrum a lo largo del desarrollo de este. Cabe destacar que una de las reglas identitarias de Scrum es que este marco debe ser una base adoptándose este a las características del equipo que lo use.

La idea es utilizar las bases de Scrum y adaptarlo al desarrollo de un proyecto multi disciplinar en la que el único integrante ira asumiendo los diferentes roles de Scrum, así como realizando sus funciones a lo largo del tiempo y entendiendo cada rol siendo consciente de sus límites en cada momento.

Se va a mantener la idea del uso de Sprint, con una pequeña modificación. No va a existir un tiempo fijo real, aunque si un tiempo fijo estimado. Los Sprints serán de dos semanas de trabajo, aunque no siempre se pueda cumplir dicho valor por no disponer del tiempo necesario.

A su vez, se realizarán las planificaciones de Sprints, dejando claro el trabajo a realizar en dicho Sprint y el incremento de valor sobre el proyecto final, aunque se va a hacer un mayor énfasis en la parte de implementación que en la parte de documentación.

Se mantiene la idea del Backlog de producto, pese a perder el concepto de Dailys y revisiones, manteniendo únicamente las retrospectivas para revisar el trabajo realizado durante el sprint y anotar posibles modificaciones de cara al siguiente Sprint.

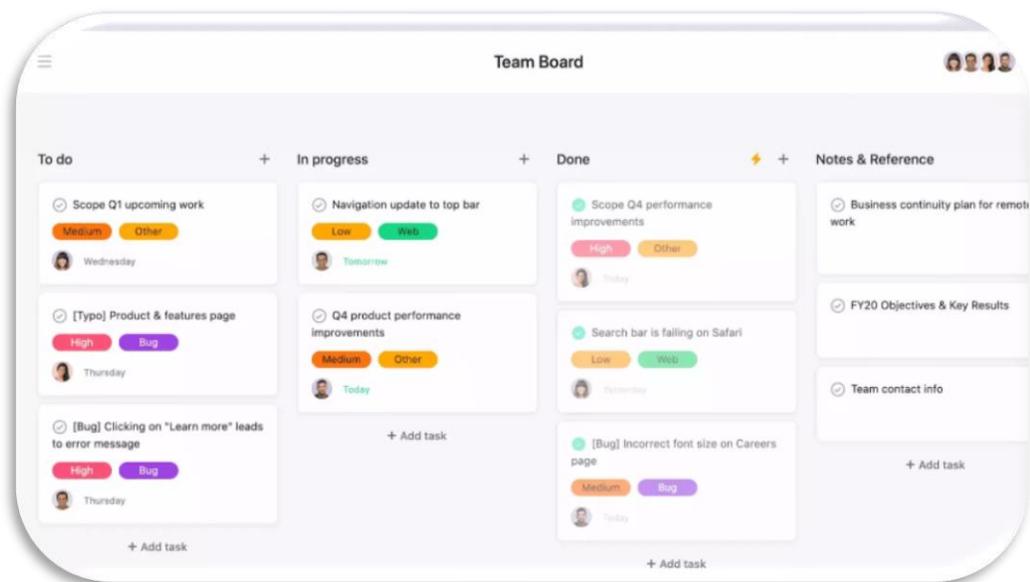


Ilustración 10 - Tablero Scrum

El realizador del proyecto iniciará el Sprint con el rol de Product Owner, teniendo claro cuál es el objetivo de negocio para dicho Sprint, una vez está claro, procederá a elaborar las propias historias de usuario seleccionadas para el equipo de desarrollo. Durante la fase de documentación no se realizará estimación de puntos de historia o tiempo porque la realización es secuencial y equitativamente prioritaria. Las estimaciones llegarán en la fase de desarrollo e implementación del ERP.

Una vez que las historias de usuario están definidas, se cambia el rol a desarrollador para llevarlas a cabo, es cierto, que en la caracterización del marco de trabajo Scrum para este proyecto pierde fuerza la figura del Scrum Máster, puesto que pierde cierto sentido.

3.2.1.Sprint 1:

Durante el primer Sprint se completaron tres funcionalidades. Las dos primeras funcionalidades se referencian dentro de N-DOC01 – Introducción. Estas funcionalidades serán la descripción del problema y los objetivos. A su vez, la primera se desglosaba en cinco historias de usuario, todas ellas documentales, la definición de un ERP, las ventajas y desventajas de este, el ERP como aplicación web y su implantación.

En cuanto a la segunda funcionalidad se desglosaba en una historia de usuario basada en la definición de los objetivos del trabajo. La tercera funcionalidad completada durante este Sprint pertenece a la referencia de N-DOC02 – Estado del Arte.

Esta funcionalidad agrupa cinco historias de usuario, la definición de análisis de mercado, la identificación de este, un análisis PEST, búsqueda de consumidores potenciales y estudio de la competencia.

Es decir, en el Sprint de dos semanas de trabajo se realizaron once historias de usuario. Tras la finalización de este se realizó un sprint Review y sprint retrospectivo con la tutora para analizar el trabajo realizado durante el mismo, se encontró cierto nivel de deuda “técnica”, catalogado así, pese a ser trabajo documental.

Esta deuda técnica se añade al siguiente sprint como carga de trabajo.



Ilustración 11 - Columna de Reviewed al Finalizar Sprint 1

3.2.2.Sprint 2:

El segundo Sprint recogió la deuda técnica del primero resolviendo los errores documentales analizados en la retrospectiva. Lo que ocupó gran parte del Sprint y provocó que solo se completará una funcionalidad. Esta está referenciada dentro de N-IMP01 – Diseño y Desarrollo.

Esta funcionalidad presenta dos historias de usuario, la realización del diagrama de componentes y la maquetación de los templates de la aplicación mediante HTML, CSS y JavaScript, dicha maquetación se verá afectada mínimamente en un futuro.



Ilustración 12 - Product Backlog del Sprint 2

Aunque en un primer momento se planteó documentar durante este segundo Sprint de manera completa todo el trabajo de diseño, se decidió esperar e ir documentando el diseño realizado a la vez que se expliquen las funcionalidades implementadas en cada sección de la aplicación.

3.2.3.Sprint 3:

El Sprint 3 es meramente documental y no se ve recogido dentro del versionado de Git, ya que sirvió para documentar todas las metodologías y herramientas utilizadas hasta este momento en los dos primeros Sprints.

En el Sprint se completaron dos funcionalidades, la primera referenciada en N-DOC03 – Metodología, esta, agrupaba siete historias de usuario completadas, aunque realmente hay una que se irá modificando sprint a sprint, que no es otra que la parte del documento actual donde se explican los Sprint, la daremos por finalizada a nivel Trello.

En cuanto a la segunda funcionalidad, referenciada en N-DOC04 – Tecnologías y Herramientas Utilizadas, se completaron siete historias de usuario, pero no se da como finalizada puesto que más adelante desarrollará nuevas historias de usuario.

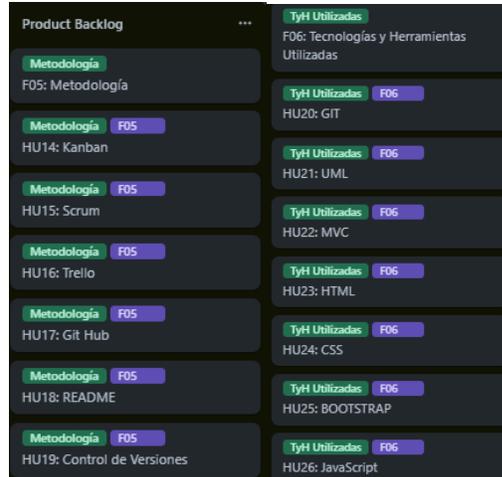


Ilustración 13 - Sprint Backlog del Sprint 3

3.2.4.Sprint 4:

En este Sprint coge importancia la figura de Product Owner ya que se empiezan a desglosar las diversas funcionalidades requeridas dentro de la aplicación. Es el Sprint más ambicioso de los cuatro primeros con diecisiete historias de usuario.

Dentro de la referencia N-DOC04 – Tecnologías y Herramientas Utilizadas del anterior Sprint, aparecen cinco historias de usuario nuevas, lo que hace que la funcionalidad F06 agrupe doce historias de usuario, que hacen referencia a tecnologías utilizadas dentro del propio Sprint.

A su vez, dentro de la referencia N-IMP01 – Diseño y Desarrollo, comenzamos una nueva funcionalidad, la F07: Desarrollo que va a agrupar todas las historias de usuario que tengan relación con implementación de código, en el caso de este Sprint agrupa doce historias de usuarios, que implementan la parte back de la aplicación, de estas doce, solo se implementaron nueve, por lo que el Sprint cinco comenzará con tres historias de usuario como deuda, Dashboard, Facturación y Finanzas.

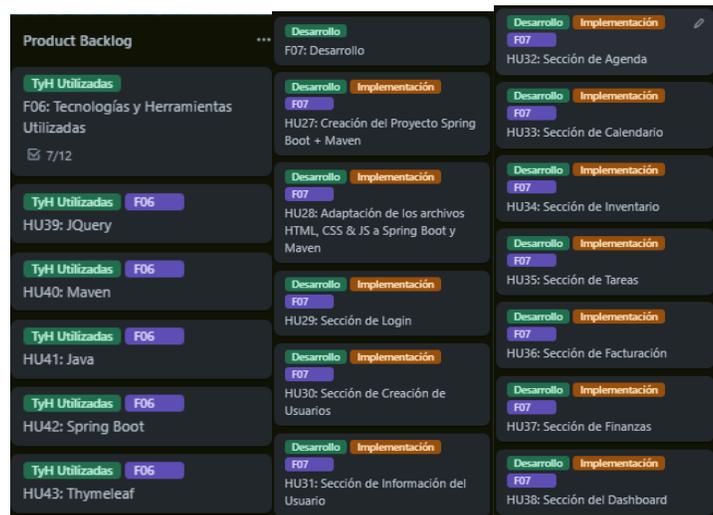


Ilustración 14 – Sprint Backlog del Sprint 4

3.2.5.Sprint 5:

El Sprint cinco es el último del proyecto, agrupa once historias de usuario de distintas funcionalidades diferentes. Dentro de la referencia N-DOC04 – Tecnologías y Herramientas Utilizadas del anterior Sprint, aparecen dos historias de usuario nuevas, lo que hace que la funcionalidad F06 agrupe catorce historias de usuario, que hacen referencia a tecnologías utilizadas dentro del propio Sprint.

A su vez, dentro de la referencia N-IMP01 – Diseño y Desarrollo, continuamos con la funcionalidad F07: Desarrollo con las tres historias de usuario que dejamos como deuda en el anterior. Esta funcionalidad ha agrupado quince historias de usuario, que implementan la parte back de la aplicación.

Además, han aparecido tres funcionalidades nuevas que hacen referencia a la documentación de conclusiones y trabajos futuros, bibliografía y anexos, cada una de ellas con su historia de usuario correspondiente.

El proyecto ha conestado de diez funcionalidades, desglosadas en cincuenta y una historias de usuario.



Ilustración 15 - Done del Sprint 5

3.3.Trello:

Trello [17] es un software de gestión en línea, enfocado al trabajo de la metodología Kanban explicada en el [punto 3.1.](#) y utilizada durante la elaboración de este proyecto. Trello proporciona una sencilla interfaz y un simple manejo.

Trello provee a los usuarios de un sistema de trabajo colaborativo, respecto a esta característica ha servido de gestión entre alumno y tutor para comprobar el estado y flujo del proyecto en todo momento. Para este proyecto se ha aprovechado su sistema de etiquetado creando tarjetas funcionales y descriptivas bajo un modelo explicado más adelante en este apartado.

Como se ha mencionado se ha utilizado el método Kanban apoyado en la herramienta Trello, que nos facilita la construcción de un tablero como se ha explicado en el apartado anterior. Identificamos siete columnas muy específicas, la primera, denominada Sprint Planning almacena íntegramente las ideas principales del proyecto.

La segunda columna se denomina Sprint Backlog, se ha utilizado para desglosar y detallar las tarjetas previsualizadas en el Sprint Planning. La tercera columna es nuestro Product Backlog, en él, desarrollamos las historias de usuario fragmentadas y definidas a partir de nuestro Sprint Backlog.

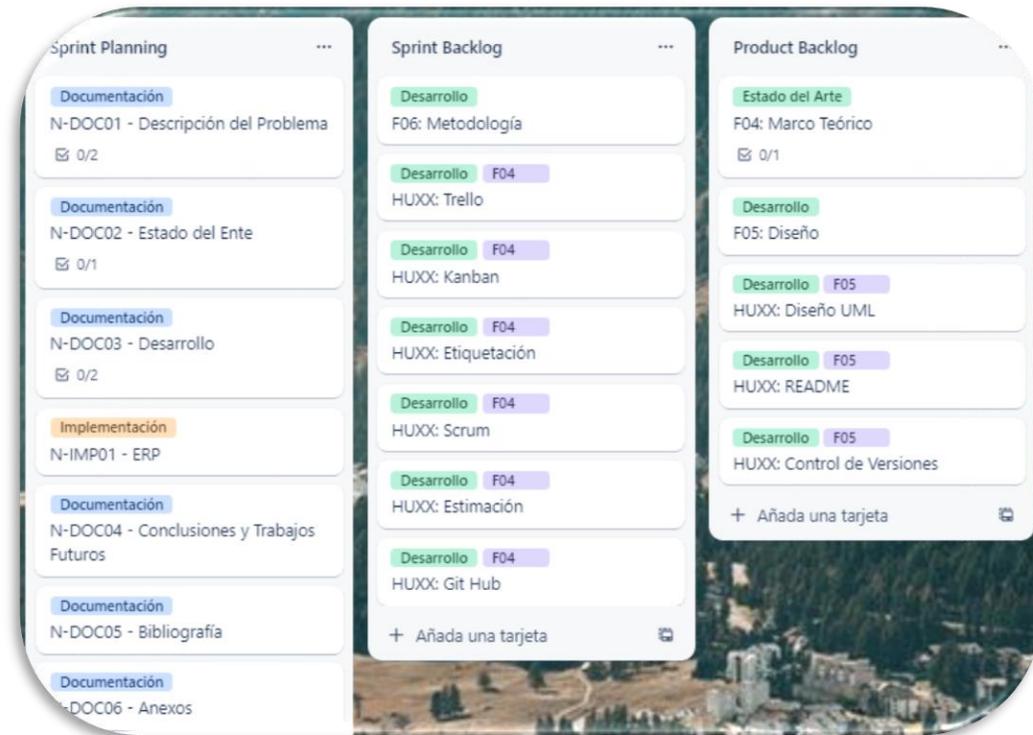


Ilustración 16 - Columnas del Tablero Trello

A continuación, nos encontramos con tres columnas denominadas Doing, Done y Reviewed. En todas ellas podemos encontrar las historias de usuario con las que estamos trabajando en un momento determinado. Específicamente dentro de la columna Done, podemos encontrar tanto historias de usuario, como funcionalidades del Sprint Backlog e incluso los campos generales del Sprint Planning. La columna Done agrupa todas las tarjetas cuyo flujo está a punto de acabar y solo falta que sean revisadas.

Finalmente, la columna Reviewed hace referencia no solo a un trabajo finalizado, sino, además, revisado exitosamente. Cabe destacar la existencia de una columna extra donde se podrán colocar versiones de la memoria del proyecto en diferentes momentos temporales para su revisión.

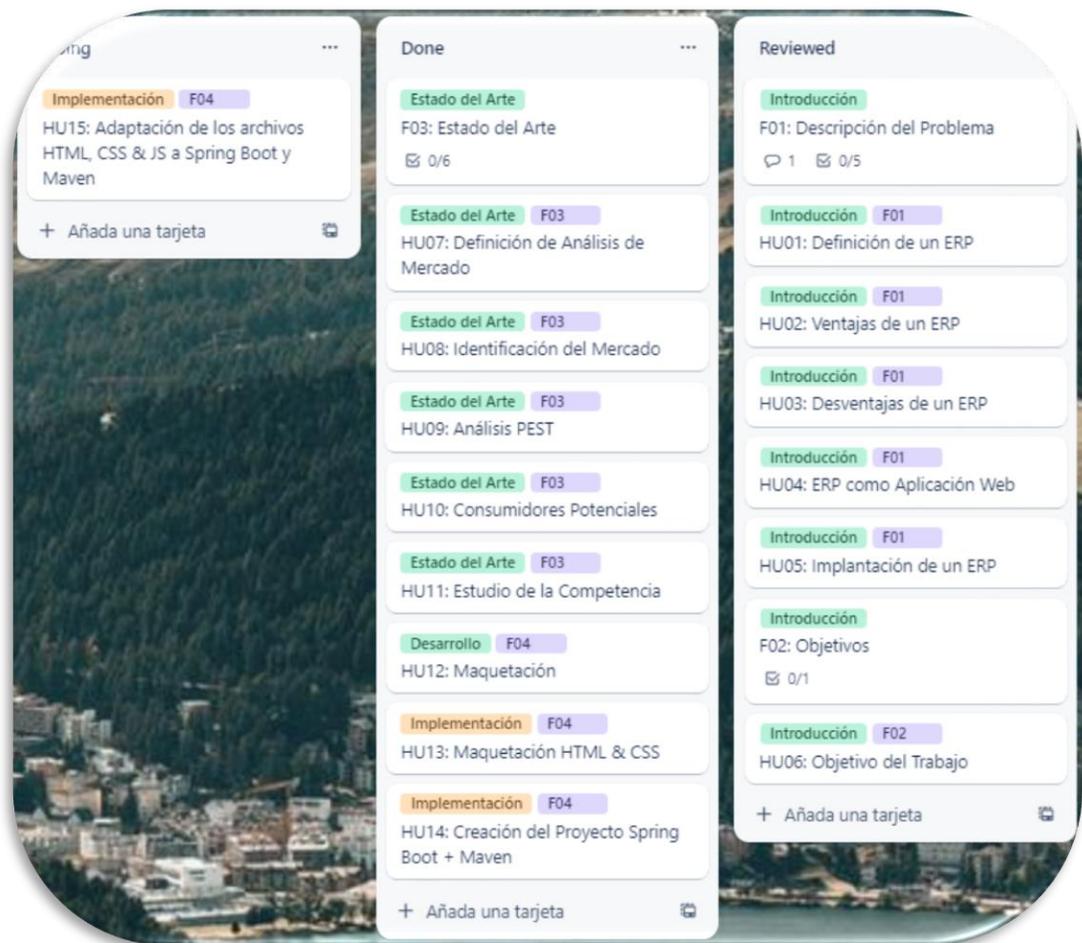


Ilustración 17 - Columnas Doing-Done-Reviewed

Dentro de la creación de las diferentes tarjetas del tablero Trello se ha implementado una tabla de etiquetas para identificar rápidamente la referencia de esta tarjeta. Dentro de la columna de Sprint Planning podemos encontrar dos tipos de etiqueta, una azul para referenciar un trabajo de documentación, con el ID *Documentación* y una etiqueta naranja para referenciar un trabajo de implementación con el ID *Implementación*.

Además, todas las tarjetas con la etiqueta azul del Sprint Planning comenzarán con el título *N-DOCX-Nombre*, en el caso de la tarjeta naranja comenzarán con *N-IMPOX-Nombre*.

Cada columna posee una etiquetación diferente, para las tarjetas del Sprint Backlog poseemos un solo tipo de etiquetación de color verde con un ID, este es exactamente igual al nombre de la tarjeta del Sprint Planning al que hacen referencia. Las tarjetas del Sprint Backlog se nombran como *FXX: Nombre*.

En el caso del Product Backlog donde creamos las diferentes historias de usuario visualizamos dos etiquetas diferentes, la primera heredada del Sprint Backlog, la de color verde referenciando a la tarjeta del Sprint Planning y una nueva etiqueta morada cuyo ID es el número identificador de las tarjetas del Sprint Backlog con la forma *HUXX*. El nombre de estas tarjetas será *HUXX: Nombre*.

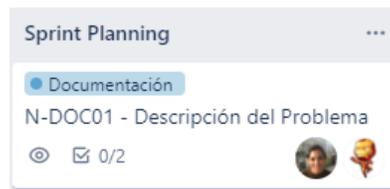


Ilustración 18 - Tarjeta del Sprint Planning

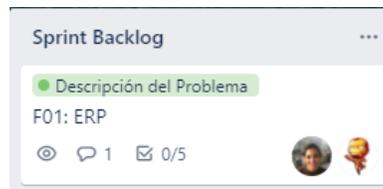


Ilustración 19 - Tarjeta del Sprint Backlog

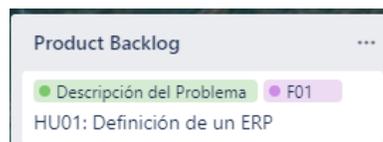


Ilustración 20 - Tarjeta del Product Backlog

3.4. Git-Hub:

En el proyecto se va a utilizar Git-Flow, dentro de este flujo de trabajo tendremos cinco ramas diferenciadas.

La primera rama es master, siendo la rama principal contiene el contenido en producción y debe ser estable, la rama Development es una rama clonada de master con el objetivo de integrar todas las nuevas funcionalidades, tras la integración y la revisión de errores para estabilizar la rama, está se lleva a master.

La rama Features se origina desde Development, para cada nueva funcionalidad deberemos crear una rama Feature nueva con un nombre identificativo, una vez que la funcionalidad está terminada y probada, se combina con Development.

Finalmente, la rama HotFix se utiliza para solucionar errores encontrados en producción, se clona de master y una vez que se corrige el error urgentemente se devuelve la versión al master.

Ahora que sabemos lo que es Git, definimos Git-Hub [18] como la plataforma de desarrollo colaborativo que aloja proyectos que utilizan el control de versiones Git.

Para la creación del proyecto vamos a utilizar la aplicación Git Bash que no es más que una línea de comandos con el intérprete de Git. Creamos un repositorio en Git, se llamará Manager Societies igual que el proyecto en cuestión.

Vamos a crear nuestro archivo Readme donde ira toda la documentación del proyecto, los comandos utilizados para la creación de este documento como para la inicialización del proyecto en Git son los siguientes:

```
echo "# ManagerSocieties" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/JaviBarrio6/ManagerSocieties.git
git push -u origin main
```

Ilustración 21: Creación Proyecto

Una vez creado nuestro proyecto en la rama master, crearemos la rama Development y Feature para empezar a trabajar en él.

3.5. README:

Un archivo README es una forma de documentar software mediante un archivo de texto plano. Está ligado con GitHub ya que si aparece en un repositorio git este es convertido en lenguaje HTML y presentado en la página principal.

Nuestro README contendrá información de interés para el desarrollo como el control de versiones, el autor, la licencia y otra información adicional.

3.6. Control de Versiones:

El control de versiones que se va a utilizar a lo largo de todo el proyecto se basa en el uso de tres números separados por tres puntos, para su correcto entendimiento los representaremos mediante las letras X-Y-Z.

Por lo que nuestra versión tendrá la forma X.Y.Z, siendo cada letra un valor numérico con un significado, el valor X hace referencia a versiones principales de software, en nuestro caso, tendremos una por cada sprint englobando todas las funcionalidades diseñadas y desarrolladas.

El valor Y serán versiones menores, una para cada funcionalidad añadida, mientras que el valor Z se reserva a la corrección de errores o revisión del código.

4. Tecnologías y Herramientas Utilizadas:

A lo largo del cuarto punto se va a realizar un estudio minorizado desde un punto de vista teórico de cada uno de los lenguajes y herramientas utilizadas en la elaboración de nuestro proyecto.

4.1. Git:

Git [16] es un software de control de versiones que busca la eficiencia, confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estás almacenan un gran número de archivos de código fuente. Git nos permite una gran libertad en la forma de trabajar en torno a un proyecto. Sin embargo, es necesario acordar un flujo de trabajo como un acuerdo entre miembros.

4.2. UML:

UML o lenguaje unificado de modelo [17] es un lenguaje de modelado de sistemas respaldado por OMG. Su propósito es capacitar un lenguaje gráfico que nos permita visualizar, especificar, construir y documentar un sistema.

UML proporciona un estándar por el cual se describe un plano del sistema que se determina como modelo, este modelo puede incluir procesos, funciones, aspectos concretos, esquemas de bases de datos, etc.

Dentro de UML podemos dividir los tipos de diagramas en dos subgrupos, los estructurales y los de comportamiento. Mientras que los diagramas estructurales muestran cómo se estructura estáticamente el sistema y sus partes en varios niveles de abstracción los diagramas de comportamiento muestran el comportamiento dinámico del sistema.

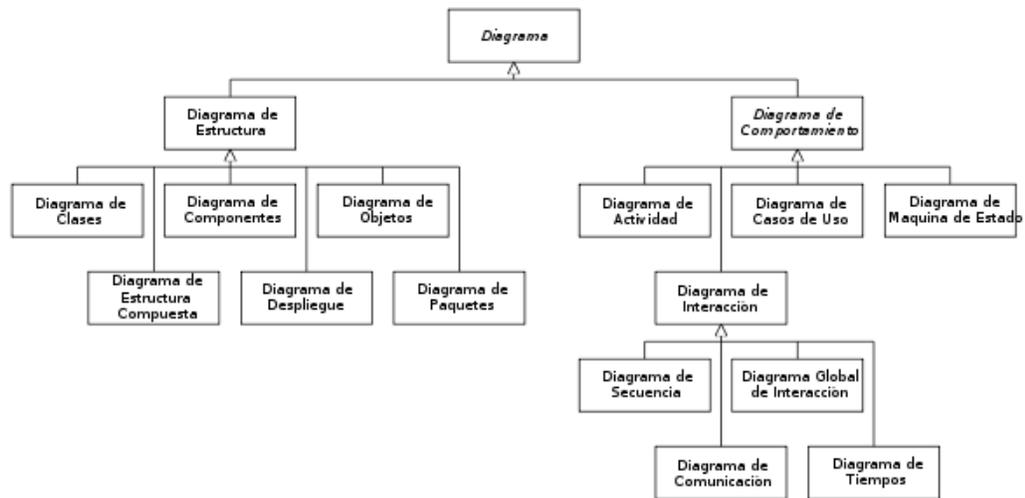


Ilustración 22 - Tipos de Diagramas UML

4.3. Modelo Vista Controlador:

MVC [18] es un estilo de arquitectura de software basado en la separación de los datos, interfaz de usuario y lógica de control de una aplicación en tres componentes distintos: Modelo, Vista y Controlador.

El Modelo contiene una representación de los datos manejados por el sistema, la lógica de negocio y sus mecanismos de persistencia. El modelo se responsabiliza de acceder a la capa de almacenamiento de datos. Se busca que el modelo sea independiente del sistema de almacenamiento.

El modelo define las reglas de negocio y funcionalidad del sistema, además, lleva un registro de las vistas y controladores del sistema. Por otro lado, si el modelo es activo, notificará a las vistas de los cambios sufridos en alguno de sus datos.

La vista o interfaz de usuario contiene la información que se envía al cliente y los mecanismos que interactúan con este. La vista es la responsable de recibir los datos emitidos por el modelo y mostrárselos al usuario. La vista suele dar un servicio de actualización de los datos para que se invoque desde el controlador o modelo.

El controlador es el intermediario entre el modelo y la vista, gestiona el flujo de información entre ellos y transforma los datos a las necesidades de cada uno. El controlador es activado mediante los eventos de entrada, y activa sus reglas de gestión de eventos, dado X evento efectúa la acción Y.

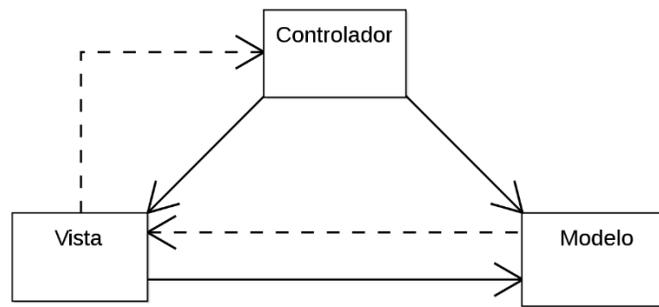


Ilustración 23 - Modelo Vista Controlador

4.4. HTML:

HTML [19] es un lenguaje de marcado de hipertexto, sus siglas en inglés son *HyperText Markup Language*. Cuando hablamos de HTML, referenciamos un lenguaje de marcado para la elaboración de páginas web. HTML es un estándar que permite conectar la elaboración de páginas web, definir una estructura básica y un código que nos permite definir el contenido de una página web, dicho contenido va desde texto, hasta vídeos.

El funcionamiento del lenguaje HTML se basa en el añadido de elemento a una página web mediante una referencia a su ubicación real, y no a una incrustación o copia de dicho contenido, siendo el navegador web el que interpreta dicho código y une todos los elementos para visualizar el contenido final.

Como HTML nos permite indicar la estructuración de los elementos de nuestro documento mediante etiquetas, esto nos ofrece una gran adaptabilidad, estructura lógica y fácil interpretación tanto por los diseñadores como por las máquinas que lo interpretan. En HTML escribimos mediante etiquetas, estas, no son más que unos identificadores encerrados entre corchetes angulares, HTML solo posee elementos. Estos a su vez, se dividen en sus atributos o su contenido. Un ejemplo básico sería `<nombreElemento atributo = "valor"> Contenido </nombreElemento>`.

4.5. CSS:

CSS [20] es un lenguaje de diseño gráfico que nos permite definir y crear la presentación deseada utilizando como base un documento escrito en un lenguaje de marcado. Sus siglas hacen referencia a *Cascading Style Sheets*, o en español, hojas de estilo en cascada.

Esto se debe a que una de sus características es que las propiedades aplicadas a los elementos exteriores son heredadas por los interiores, de ahí, el termino cascada. CSS es muy utilizado para

establecer el diseño visual de archivos HTML y XHTML; aunque este lenguaje puede ser aplicado a cualquier documento del tipo XML.

El principal motivo por el que se diseñó CSS era para marcar una clara separación entre el contenido del documento XML y la forma de representar el contenido de este. Gracias a esta separación se consigue mejorar la accesibilidad de ambos documentos, además, proveemos una mayor flexibilidad y control en la especificación de las características de los elementos del documento.

Por otro lado, CSS nos permite que varios documentos puedan poseer un mismo estilo mediante la misma hoja de estilos .css.

Existen frameworks CSS que nos permiten la simplificación y estandarización del diseño de páginas web, estos no son más que bibliotecas CSS. Algunos de los más famosos son *Bootstrap*, *Blueprint* o *Materialize*. Para utilizarlos utilizamos la etiqueta HTML `<link>` que realiza una referencia externa a dicha biblioteca.

4.6. Bootstrap:

Bootstrap [21] es una biblioteca multiplataforma o un conjunto de herramientas de código abierto para el diseño de aplicaciones web. Su función es la de incorporar en una plantilla de diseño todos los elementos HTML necesarios con su diseño gráfico CSS para la elaboración de dicha aplicación web.

Además, cuenta con extensiones de JavaScript adicionales. Aunque existen otros frameworks web similares a Bootstrap que generan una plantilla Full-Stack, este framework solo elabora plantillas Front-End.

Bootstrap es un framework modular que consiste en una serie de hojas de estilo LESS, estas implementan todos los componentes de la herramienta. Este framework permite a los desarrolladores adaptar el archivo `Bootstrap.less` para seleccionar los componentes que desean añadir a su proyecto.

Los componentes de JavaScript de Bootstrap se basan en la librería JQuery de la propia JavaScript. Esta provee elementos adicionales para la interfaz de usuario como pueden ser diálogos, botones, tooltips, etc. Además, incluyen la posibilidad de usar funciones de auto completado para inputs, entre otras características.

4.7. JavaScript:

JavaScript, o JS [22] como se le conoce de forma abreviada, es un lenguaje de programación interpretado, es un dialecto del estándar ECMAScript. A JS se le define como un lenguaje orientado a objetos, basado en prototipos, imperativo, dinámico y con un tipado débil.

Su principal función es la de añadir mejoras en la interfaz de usuarios y proporcionar dinamismo en las páginas web estáticas. Aunque se diseñó con una sintaxis similar a C++ o Java, tienen semánticas y sobre todo propósitos muy diferentes, existiendo un nombre común con motivo comercial al pertenecer al mismo dueño.

Pese a que en su origen este lenguaje se centraba en su uso del lado de cliente lejos de accesos al servidor, la tendencia a cambiado gracias al uso de tecnologías como AJAX.

4.8. JQuery:

JQuery [23] es una biblioteca multiplataforma de JS, esta nos permite interactuar con archivos HTML, manipular el DOM, controlar eventos, desarrollar animaciones de una forma simplificada. La serie de funcionalidades basadas en JS logran grandes resultados en poco tiempo y espacio.

JQuery nos permite desarrollar aplicaciones AJAX, permitiendo además crear complementos adicionales por parte de los desarrolladores, así se pueden crear abstracciones interaccionales y animaciones de bajo nivel.

JQuery nos permite separar el código JS del archivo HTML, en vez de necesitar archivos o scripts que llamen a nuestras funciones JS poseemos unos eventos manipuladores de la DOM mediante llamadas JS.

4.9. Maven:

Maven [24] es una herramienta software que nos permite gestionar y construir proyectos Java. Posee un modelo de construcción simple basado en XML. Maven utilizar un POM para describir sus proyectos, dependencias, componentes externos y sobre todo, el orden en el que se van a construir los elementos.

Maven posee la filosofía de que se debe construir en base a la estandarización del principio de la convención sobre la configuración, cuyo fin es el uso de modelos existentes para producir software.

Uno de los mayores inconvenientes de Maven es la poca adaptabilidad de migración de proyectos ya creados a proyecto Maven. Pese a ello, permite la reutilización de proyectos, para proyectos nuevos futuros con las mismas características.

Maven genera ficheros .class a partir de las fuentes Java, una vez se ha realizado esto, ejecuta si existiesen las pruebas creadas en JUnit, parando el proceso si es que alguno de ellos fallase. En el caso de que ninguno falle o no los hubiese, genera el fichero .jar con los class compilados.

Finalmente, copia el archivo .jar a un directorio concreto para su reutilización y genera el copiado en un servidor remoto.

4.10. Java:

Java [25] es un lenguaje de programación derivado del lenguaje C y C++ poseyendo menos utilidades de bajo nivel. Las aplicaciones Java son compiladas a bytecode o clases Java. Java fue creado para usar el paradigma de la orientación a objetos.

A su vez, en el diseño del lenguaje se buscó que este fuese ejecutable en programas aptos en multitud de sistemas operativos, incluyendo por defecto la posibilidad de trabajar en red, permitiendo esto la ejecución de código en sistemas remotos con cierta seguridad y crear un lenguaje sencillo orientado a objetos.

Java EE nos proporciona una plataforma para construir aplicaciones que serán transaccionales, seguras, interoperables y distribuidas para crear páginas web, conectarse a bases de datos, mapear objetos mediante APIs, gestionar componentes, proporcionarnos una interfaz de usuario, gestionar transacciones y enviar o recibir mensajes

4.11. Spring Boot:

SpringBoot [26] es una tecnología para crear aplicaciones autocontenidas permitiendo al equipo centrarse en un desarrollo a medida de esta. Para ello, queda en un segundo plano la arquitectura, es decir, Spring Boot es el encargado de configurar las dependencias y el despliegue del servicio de aplicaciones.

Spring Boot utiliza un servidor de aplicaciones embebido y un gestor completo de dependencias, esto facilita la creación de aplicaciones independientes. Spring Boot se apoya en Maven con el uso de las dependencias necesarias.

4.12. Thymeleaf:

Thymeleaf [27] es una biblioteca Java que implementa un motor de plantillas para XML y HTML. Puede ser utilizado en modo web como en los entornos sin web. Es utilizado para trabajar en la capa MVC de aplicaciones web.

Thymeleaf proporciona un módulo para el uso de Spring MVC, su objetivo es permitir la creación de plantillas de una manera elegante y con el código bien formateado.

4.13. MySQL:

MySQL [32] es un sistema de gestión de bases de datos relacional. Esta considerada como una de las bases de dato de código abierto más populares del mundo. MySQL Workbench es un entorno integrado de MySQL, permite a los usuarios administrar gráficamente y diseñar las bases de datos.

4.14. Jpa Repository:

Spring Data JPA [33] proporciona soporte de repositorio para la API de persistencia de Jakarta facilitando el desarrollo de aplicaciones que necesitan acceder a fuentes de dato JPA.

5. Diseño y Desarrollo:

En el quinto punto se explicará el desarrollo del proyecto fragmentado de manera secuencial desde su comienzo hasta el punto final.

5.1. Diseño:

En este punto se planifica, diseña, implementa y mantiene la aplicación web. Engloba la unión del buen diseño con la jerarquía de contenidos.

5.1.1. Diagrama de Componentes:

Antes de comenzar el desarrollo del ERP se debe pensar el diseño funcional de este, es decir, que componentes va a tener y cómo van a estar presentados, aunque este primer diseño es posible que sufra cambios en el futuro hay que intentar crear una base sólida que nos permita hacernos una idea general.

Para ello, utilizaremos un diseño UML de componentes, sin entrar en detalles para analizar la extensión del proyecto, más adelante se realizarán diseños de clases de cada componente.

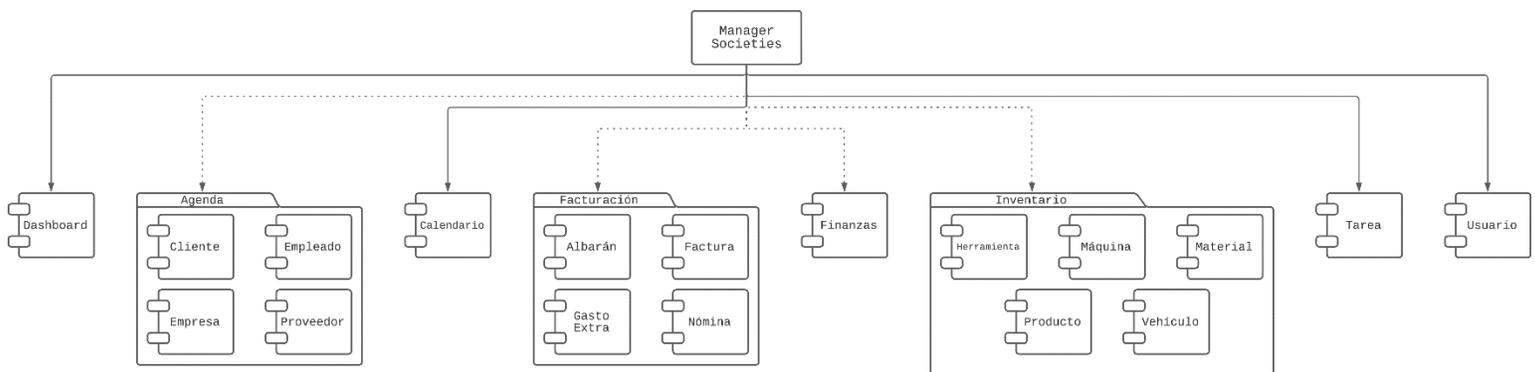


Ilustración 24: Diagrama UML de Componentes

Este diseño inicial ha sido utilizado para la elaboración de la maqueta de la aplicación, pese a que durante el diseño de esta se puedan realizar cambios en él por diferentes necesidades que se indicarán detalladas en puntos posteriores.

5.1.2. Maquetación:

A lo largo del desarrollo se va a explicar detalladamente cómo y por qué se ha utilizado cada tecnología o herramienta a lo largo del proceso de desarrollo. Para resolver nuestro problema se ha decidido realizar una aplicación web.

Una aplicación web no es más que una web que posee un servidor dinámico, el servidor sirve el contenido generado mediante código. Durante los primeros pasos implementaremos nuestra aplicación web usaremos HTML, CSS y JavaScript.

Esto provoca que nuestra aplicación web no sea más que un conjunto de recursos HTML, CSS y JavaScript que cargan en el navegador la información al acceder a la URL principal.

Para realizar la maquetación de la aplicación utilizaremos el lenguaje HTML, es el lenguaje utilizado para crear documentos en la web, siendo el navegador web el encargado de visualizar dichos documentos. Las peticiones http realizadas por un navegador web a un servidor suelen devolver documentos HTML.

El formato de documentos HTML está muy relacionado con los lenguajes CSS y JavaScript. CSS nos permite añadir estilos mientras que JavaScript añade interactividad en los clientes. La maquetación se apoyará en la librería Bootstrap para agilizar el diseño y maquetado.

Bootstrap, como se ha definido en el marco teórico, es un framework de software libre para el diseño del lado cliente de aplicaciones web. Para poder usar Bootstrap deberemos insertar ciertas etiquetas que realicen una serie de JQueryes. Para el proyecto se ha utilizado el Bootstrap gratuito *Nice Admin* [28].

Una vez elegido se ha modificado para que cumpliera con el diagrama de componentes inicial en UML. Aunque se podrían detallar dichos cambios, esperaremos a explicar el diseño junto al desarrollo funcional del mismo, ya que, el diseño estático debe ser modificado para mostrar los datos reales.

Como punto de partida se buscó el diseño de un logo que fuera atractivo a la par que visual, recordamos que tanto el nombre del proyecto como el de la aplicación es el mismo: *Manager Societies*, el término proviene del latín y su significado se traduce como gestor de sociedades. El logo diseñado es el que sigue:



Ilustración 25 - Logo de la Aplicación

5.2. Desarrollo Web:

En este punto se explica la construcción y mantenimiento de la aplicación web a nivel funcional. Se explica cómo se han estudiado las necesidades del cliente y como se ha construido el sistema para proponer una interfaz atractiva que incorpora las necesidades funcionales del usuario final.

5.2.1. Migración a Aplicación Web:

Una vez que existe un diseño elaborado, aunque este abierto a pequeños cambios, se comienza el proceso donde se transforman los archivos HTML, CSS y JavaScript a aplicaciones web, para ello se ha utilizado la tecnología Java.

Dicha migración de los archivos a aplicación web se ha realizado mediante la tecnología Java EE, apoyada en Spring. Concretamente se ha utilizado la librería Spring Boot que facilita el desarrollo de aplicaciones con Spring simplificando la configuración y acelerando el desarrollo de la aplicación.

Además, se necesita de un sistema de gestión de dependencias, por ello, se ha implementado el uso de Maven dentro del nuevo proyecto. Para ello, mediante el IDE Intel J Idea se crea dicho proyecto Maven.

El nuevo proyecto Maven debe tener una estructura marcada y definida, a simple vista visualizamos un archivo pom.xml, en él, se guarda la descripción del proyecto, así como el nombre, las dependencias, configuraciones, etc.

Una de las carpetas con mayor importancia dentro del nuevo proyecto creado es la carpeta src, está se divide en dos subcarpetas de especial interés, la carpeta main y la carpeta test, por el momento el centro de atención recae en la subcarpeta main dejando las pruebas de test para más adelante.

A su vez, main se divide en la subcarpeta java y la llamada resources, es en éstas donde se debe reorganizar todos los archivos generados hasta este momento, pero antes se debe añadir Spring MVC para la construcción de la aplicación web.

Spring MVC posee la arquitectura Model View Controller, donde se utilizan modelos de datos mediante objetos Java, plantillas generadas mediante HTML y un controlador que atiende las peticiones http que llegan del navegador.

Los controladores serán clases Java encargadas de atender cada petición web, para ello tendrán que manipular los datos que llegan mediante peticiones a bases de datos, obteniendo dichos datos para dar una vista mediante una plantilla HTML.

Para la implementación de las vistas de SpringMVC usaremos la tecnología Thymeleaf, por lo que debe añadir la librería al proyecto. Además, añadimos a nuestro archivo pom las dependencias tanto de Spring, como de Thymeleaf.

Thymeleaf [29] es un motor de plantillas Java desde el lado del servidor para entornos web, su objetivo principal es proporcionar plantillas elegantes al flujo de trabajo de desarrollo. Permite mostrar correctamente código HTML en los navegadores e integrar funcionalidades dinámicas a través de módulos de Spring Framework.

Se reubican las veinticuatro plantillas en su carpeta correspondiente, esta está dentro de la carpeta *resources* y es denominada *templates*. Además, se deben reubicar los archivos .css, .js y las imágenes necesarias para la aplicación, todo ello se ubica en la carpeta *static* dentro de *resources*. Por último, se ha creado un subdirectorio con esta memoria actualizada en formato pdf.

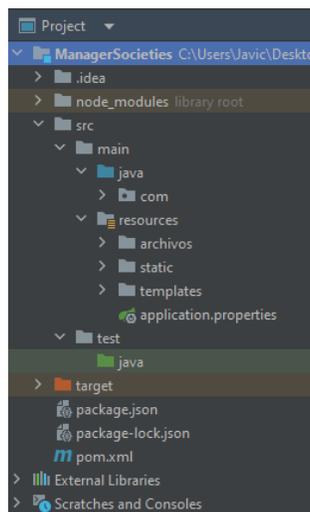


Ilustración 26 - Directorios del Proyecto Maven

Dentro de la carpeta *java*, y del subdirectorio *com* se encuentra la clase Java aplicación. Para su correcto funcionamiento se ha añadido la exclusión de auto configuración de Spring Boot.

A continuación, se genera el primer controlador de la aplicación, el *ApplicationController* que hará las veces de nexo entre el resto de los controladores de esta.

5.2.2. Aplicación:

Para el correcto uso de Maven y Spring Boot se ha generado una clase aplicación simple, en ella realmente solo hay un método ejecutable. Este llama a *SpringApplication.run* con la clase de la aplicación y sus argumentos.

Como modificación de interés se ha añadido a la notación *SpringBootApplication*, que auto configura la aplicación de forma que podamos escanear componentes, una exclusión en la que eliminamos la configuración de seguridad predeterminada para poder crear así un componente de errores y a través del archivo *application.properties* indicar el comportamiento de la aplicación en caso de obtener un error.

```
@SpringBootApplication (exclude = SecurityAutoConfiguration.class)
public class Application {
    @ Javibarrío6
    public static void main(String[] args) { SpringApplication.run(Application.class, args); }
}
```

Ilustración 27 - Clase Principal

5.2.3. Controlador de Aplicación:

El controlador de la aplicación es el punto de unión entre la clase Aplicación y el resto de los controladores de esta. Es el encargado de hacer las peticiones de mapeo de la aplicación solicitando a cada controlador que realice el correcto comportamiento sobre su modelo vista.

Cada vez que se hace una petición de mapeo, el controlador de aplicación la recibe, detecta si dicha petición existe y en ese caso, solicita al controlador determinado que realice los métodos necesarios de forma que se cumpla el modelo vista de la petición.

5.2.4. Inicio de Sesión:

La primera vista de la aplicación es la sección de Login, se ha elegido un diseño minimalista en el que nos encontramos con un mensaje de bienvenida y un par de inputs solicitando el usuario y la contraseña junto a un botón que solicita la petición de mapeo del inicio de sesión.

Cuando se realiza la petición de mapeo de inicio de sesión, esta, llega al controlador de la aplicación que deriva la verificación del usuario al controlador de usuarios. Antes de comprender las funciones de este debemos conocer las características de un usuario.



Ilustración 28 - Sección de Login

La clase Usuario se caracteriza por cuatro variables, una variable de texto contraseña, un booleano que determina si el usuario es administrador, una variable de texto que especifica la Url de la imagen de perfil del usuario y una variable Empleado con la información del usuario empleado. La clase Empleado de dicha variable será analizada en la sección de Agenda vista posteriormente.

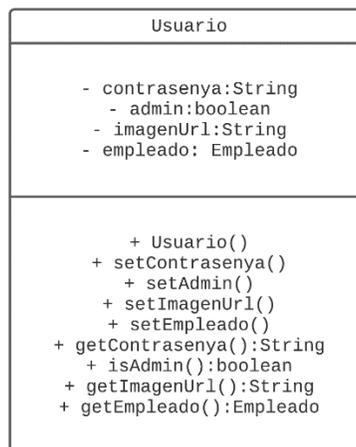


Ilustración 29 - Diagrama de la Clase Usuario

La clase Usuario posee sus getter y setter, nada más. A partir de la clase Usuario se genera una clase Usuarios que servirá de base de datos provisional hasta que la aplicación tenga la suya propia, en ella podemos visualizar una variable usuarios de tipo LinkedHashMap, se ha utilizado la combinación de la tabla Hash más la lista doblemente enlazada por su capacidad de ordenar y controlar los datos en función de su clave valor, siendo su clave el nombre de usuario y el valor la instancia usuario completa.

La clase Usuarios migrará a una clase Interface cuando la base de datos esté operativa, en esta clase nos encontramos la función `editarUsuario` que nos permite la modificación de este mediante la referencia única del mismo, encontrando al usuario por dicha referencia en nuestra lista y aprovechando el constructor que incluía está para crear una nueva instancia con los datos modificados. Además, encontramos una función para modificar únicamente la contraseña, ya que este valor no permitimos modificarlo en el método anterior, para ello se busca al usuario dentro de la lista por su clave específica y se modifica el valor mediante el set propio de la clase.

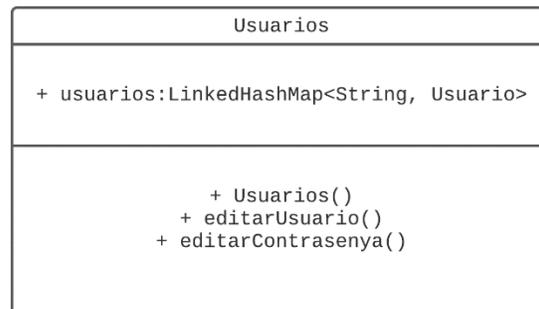


Ilustración 30 - Diagrama de Clase Usuarios

Una vez entendido el funcionamiento de los usuarios podemos visualizar el comportamiento del controlador de estos, este controlador posee siete métodos, pero en esta sección solo se explicarán dos de ellos, el primero llamado `loginPage` se utiliza para volver a la sección de inicio de sesión cuando las credenciales introducidas no son las correctas. El controlador carga la vista con el template de inicio de sesión y añade un objeto booleano falso para que mediante Thymeleaf mostremos el mensaje de que las credenciales son erróneas.



Ilustración 31 - Inicio de Sesión fallido

El siguiente método denominado `login` es el encargado de comprobar si el usuario introducido está contenido entre nuestros usuarios y si así es, se comprueba si la contraseña proporcionada se corresponde a la de este usuario. En el caso de ser correcta el controlador de usuario avisa a al controlador de la aplicación de que dicho inicio de sesión es correcto.

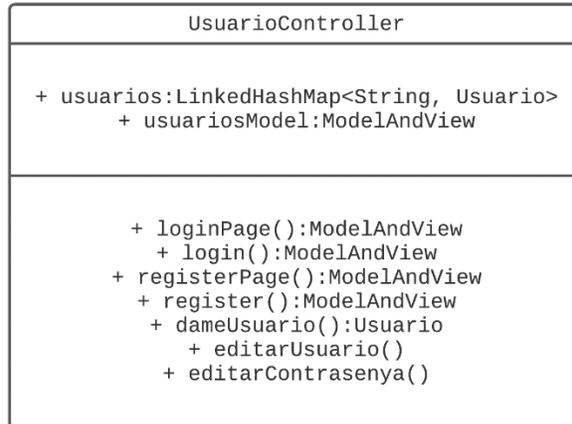


Ilustración 32 - Diagrama de Clase del Controlador de Usuarios

A partir de este momento, el controlador de aplicación activa su variable booleana de login a cierta y le solicita al controlador de usuarios que le proporcione en una variable Usuario toda la información del usuario que ha iniciado sesión, esta información será utilizada más adelante para mostrar los datos del usuario en los diferentes templates de la aplicación. A continuación, el controlador de la aplicación llama a su propio método index para mostrar el dashboard de la aplicación.

El comportamiento y funcionamiento del controlador de la aplicación se irá explicando fragmentado debido a su extensión y complejidad. De tal modo, en los diagramas únicamente aparecerán los métodos y variables que estén relacionados con esta fragmentación.

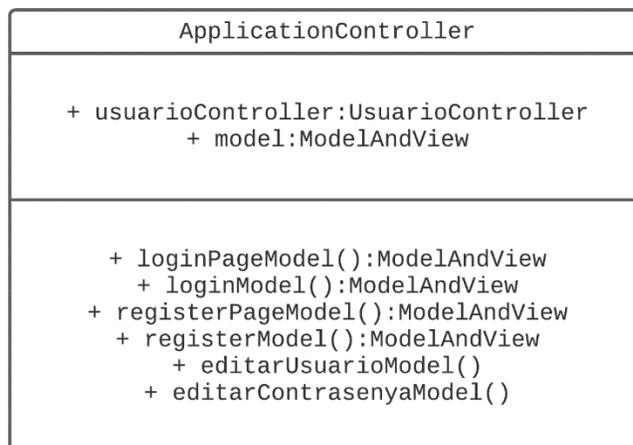


Ilustración 33 - Diagrama de Clase del Controlador de Aplicación

5.2.5. Información del Usuario:

Desde cualquier sección de la aplicación podemos acceder a la información del usuario desde un menú desplegable situado en la parte de navegación derecha. Esta tiene dos aspectos diferenciados en base a si el usuario es administrador o no.



Ilustración 34 - Desplegable Información del Usuario

En ambos casos, aparece la foto de perfil junto al nombre y apellidos del usuario. En el menú desplegable volverá a mostrarse el nombre junto a los apellidos, pero aparecerá debajo Administrador en el caso de serlo o empleado en caso contrario. Dentro de las opciones del menú la única diferencia es que los administradores pueden acceder a un campo empresa que se explicará en el futuro.

El resto de las opciones son mi perfil, ajustes de cuenta, cambiar la contraseña o cerrar la sesión, lo que nos permite volver a la sección de login. Si el usuario accede a su perfil se le mostrará una sección en la que podrá ver su foto, el logo de su empresa junto al CIF, su nombre, identificador de usuario, DNI, teléfono, correo, dirección y puesto en la empresa.

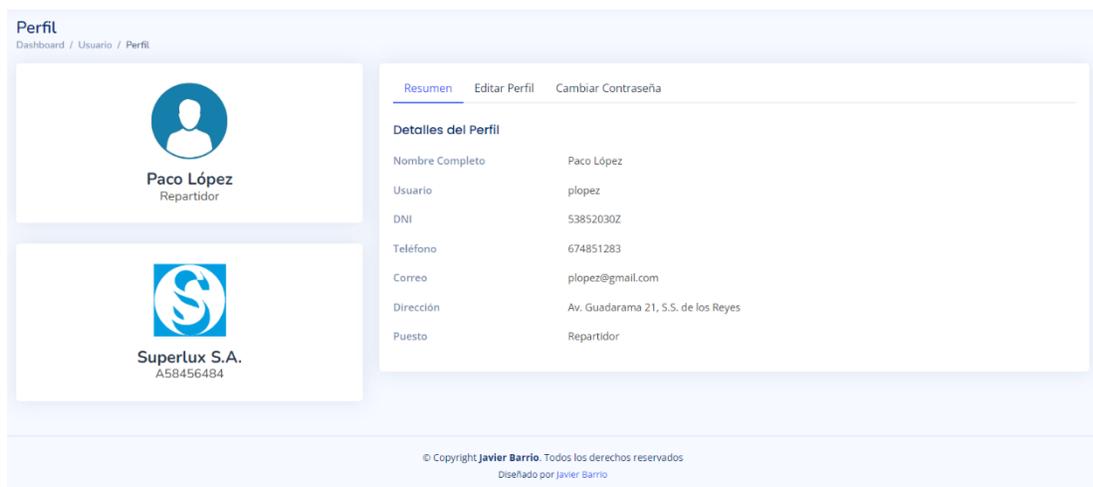
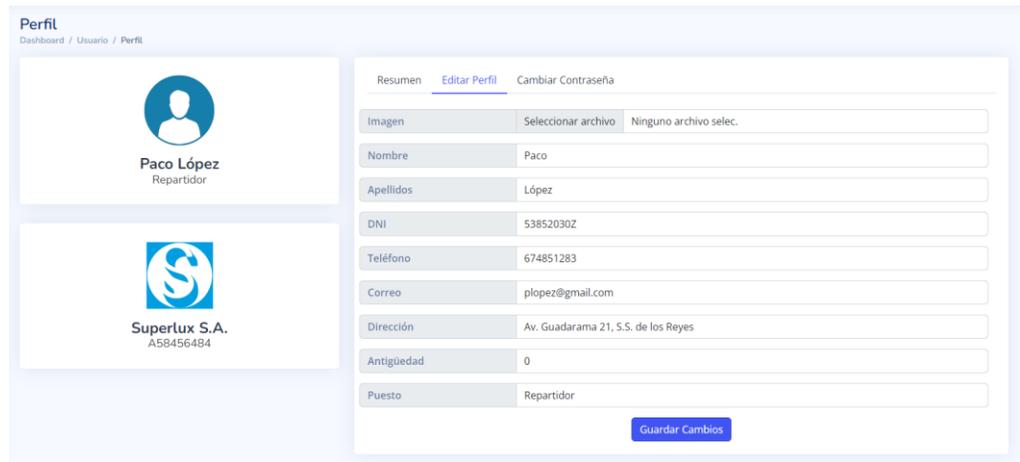


Ilustración 35 - Sección Mi Perfil

Si un usuario accede a los ajustes de cuenta se le mostrarán los datos mencionados anteriormente, pero con la posibilidad de ser modificados mediante una serie de inputs y un botón de guardar cambios. Para el correcto funcionamiento de esta sección el controlador de la aplicación recibe la petición de mapeo de la función `editarUsuarioModel`, vista en la ilustración 28, esta a su vez, solicita al controlador de usuario y al controlador de empleado que modifique los datos del empleado y del usuario de este. Por último, actualiza el valor del usuario activo en memoria.



Perfil
Dashboard / Usuario / Perfil

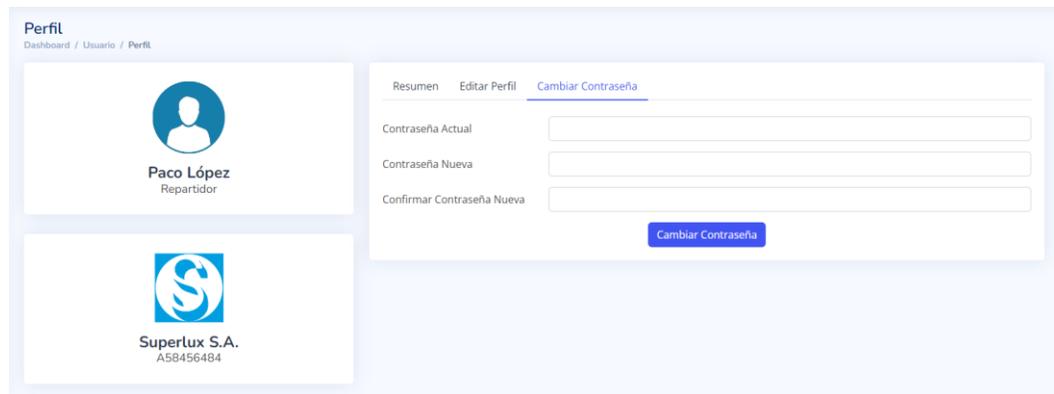
Resumen **Editar Perfil** Cambiar Contraseña

Imagen	Seleccionar archivo	Ninguno archivo selec.
Nombre	Paco	
Apellidos	López	
DNI	538520302	
Teléfono	674851283	
Correo	plopez@gmail.com	
Dirección	Av. Guadarama 21, S.S. de los Reyes	
Antigüedad	0	
Puesto	Repartidor	

Guardar Cambios

Ilustración 36 - Sección Ajustes de Cuenta

Cuando el usuario accede a cambio de contraseña se le mostrarán tres inputs en los que tendrá que indicar su contraseña actual, la nueva y confirmar esta última. Del mismo modo que en el caso anterior se solicita la petición de mapeo al controlador de la aplicación y es este el que solicita al controlador del usuario la edición de esta a través de la función `editarContraseñaModel`.



Perfil
Dashboard / Usuario / Perfil

Resumen Editar Perfil **Cambiar Contraseña**

Contraseña Actual	<input type="text"/>
Contraseña Nueva	<input type="text"/>
Confirmar Contraseña Nueva	<input type="text"/>

Cambiar Contraseña

Ilustración 37 - Sección Cambiar Contraseña

En el caso de que el usuario sea administrador posee una vista adicional, esta hace referencia a la información de la empresa. En ella se puede indicar y modificar el logo de la empresa, el nombre, CIF, teléfono, correo, dirección e IBAN.

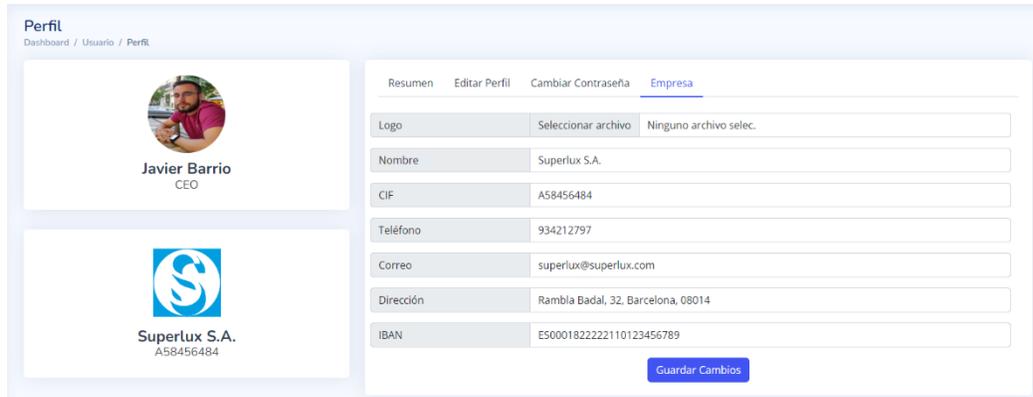


Ilustración 38 - Edición de la Empresa

Para la administración de esta información existe una clase Empresa muy simple con dichas variables, los métodos Get y Set correspondientes, el método de edición de la empresa y uno que nos comprueba si el CIF introducido es correcto.

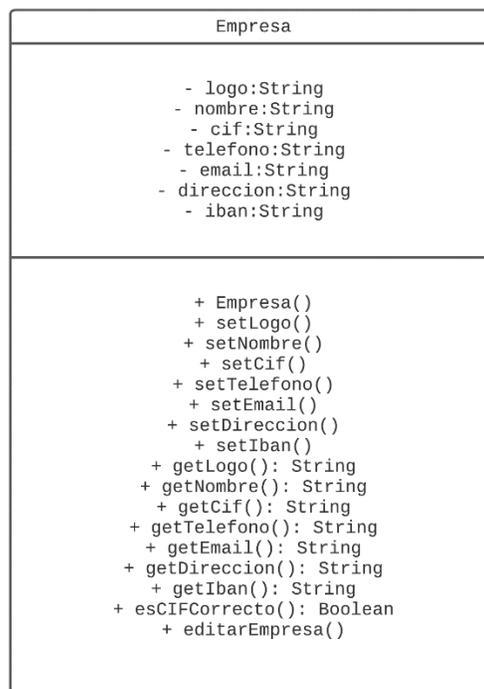


Ilustración 39 - Diagrama de Clase Empresa

A falta de la explicación del registro de nuevos usuarios que se hará en el siguiente apartado, podemos visualizar el diagrama de clases UML del conjunto de la clase Usuario, la clase Usuarios, el controlador de usuarios y su unión con el controlador de la aplicación.

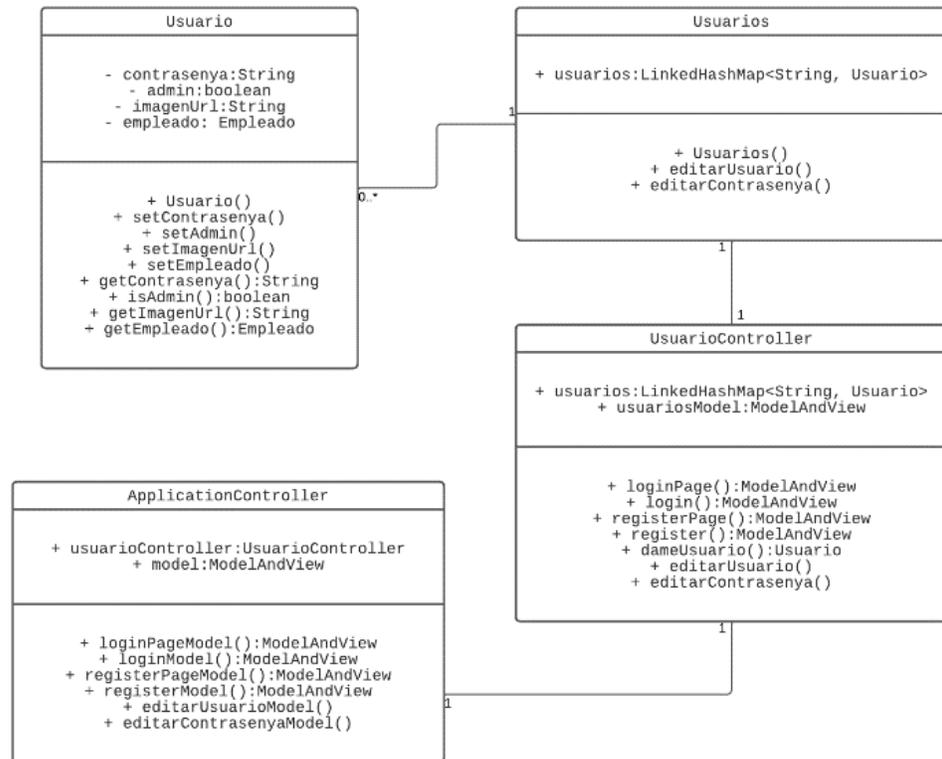


Ilustración 40 - Diagrama de Clases de Usuarios

5.2.6. Creación de Usuario:

Un usuario administrador posee la opción de crear usuarios a partir de los empleados registrados en la empresa. Para ello, en el menú de navegación izquierdo posee un botón a esta sección con el nombre Crear Usuario.

Manager Societies

- Dashboard
- Agenda
- Calendario
- Facturación
- Finanzas
- Inventario
- Tareas
- ADMINISTRADOR
- Crear Usuario

Ilustración 41 - Menú de Navegación rol Administrador

Dentro de esta sección aparece un input seleccionable con los empleados sin acceso a la aplicación, debajo dos inputs donde se indicará el usuario y la contraseña de estos, por último, mediante una casilla de selección se podrá indicar si posee el rol de administrador o no.



Ilustración 42 - Sección de Creación de Usuarios

Al seleccionar el botón de creación de usuario el controlador de la aplicación lanza una petición de mapeo a la función registerModel que solicita al controlador de usuarios el registro de un nuevo usuario. Este llama al constructor de la clase Usuario para luego añadirlo a la clase Usuarios utilizando su id como identificador. Una vez creado el usuario el controlador de la aplicación nos traslada al Dashboard.

5.2.7. Agenda:

El módulo de la agenda en la aplicación es exclusivo del rol de administrador y agrupa cuatro secciones, el apartado para clientes, empleados, empresas subcontratadas y proveedores. Aunque todos ellos poseen un diseño común, tienen sus pequeñas particularidades que los diferencian del resto.



Ilustración 43 - Navegación de la Agenda

Para el correcto funcionamiento de la agenda se ha generado una clase Persona con características comunes al resto de objetos del módulo. Una Persona se caracteriza con su nombre, apellidos, identificador, teléfono, correo y dirección, todos ellos variables de texto, y por último por un listado de documentos.

Dentro de las funciones de una Persona podemos encontrar los setter y getter, necesarios para la modificación y obtención de los datos desde otras clases Java. Pero, de interés la función esDni que determina si un identificador es un Dni.

Relacionada con la anterior la función esDNICorrecto determina si un identificador DNI posee las condiciones para que sea aceptado como un DNI existente, de igual manera, existe la función esCIFCorrecto que hace lo propio.

Por último, se ha diseñado una función imprescindible denominada generadorRef que dado un número y un String elabora una combinación de ambos en la que obtendremos una referencia única para cada instancia de nuestra aplicación.

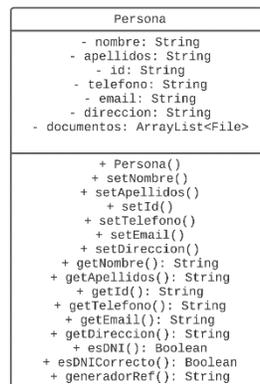


Ilustración 44 - Diagrama de Clase Persona

La primera sección de la Agenda es la sección de Clientes, para su diseño se ha elaborado el archivo CSS, en el que entraremos en detalle a continuación, la clase Cliente hereda las características de la clase Persona, pero además posee un generadorId numérico, variable que aparecerá en todas las clases, ya que, permite controlar el número exacto de instancias para generar la próxima referencia. Este será eliminado al crear la base de datos, ya que se consultará a ella cuál será dicho número.

Por otro lado, una constante de texto que denomina la referencia para los clientes *CLI*, una variable booleana para determinar si son clientes premium o no, y la propia referencia completa. La clase Cliente posee tres constructores disponibles, el primero en el que no se le pasa ningún valor, un segundo en el que se construye la instancia a partir de todas sus variables menos la referencia que será generada dentro del propio constructor, y, en último lugar, otro en el que la referencia es dada.

El resto de las funciones de la clase son getter y setter. A partir de la clase Cliente se genera una clase Clientes que servirá de base de datos provisional hasta que la aplicación tenga la suya propia, en ella podemos visualizar una variable clientes de tipo LinkedHashMap, se ha utilizado la combinación de la tabla Hash más la lista doblemente enlazada por su capacidad de ordenar y controlar los datos en función de su clave valor, siendo su clave la referencia única generada para cada instancia y el valor la instancia cliente completa.

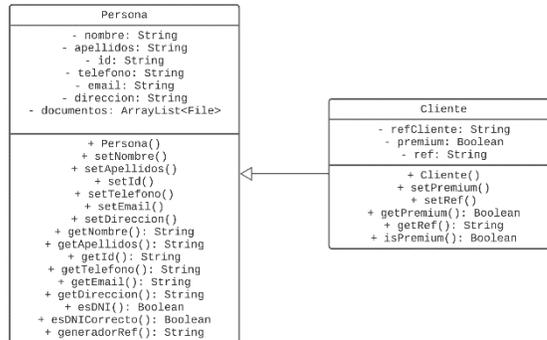


Ilustración 45 - Diagrama de Clase Cliente

La clase Clientes migrará a una clase Interface cuando la base de datos esté operativa, esta clase posee la función `getClientesEmpresas` que devuelve el número de clientes que no son particulares si no una empresa. Además, la función `editarCliente` que permite la modificación de este mediante la referencia única del mismo, encontrando al cliente por dicha referencia en nuestra lista y aprovechando el constructor que incluía está para crear una nueva instancia con los datos modificados.

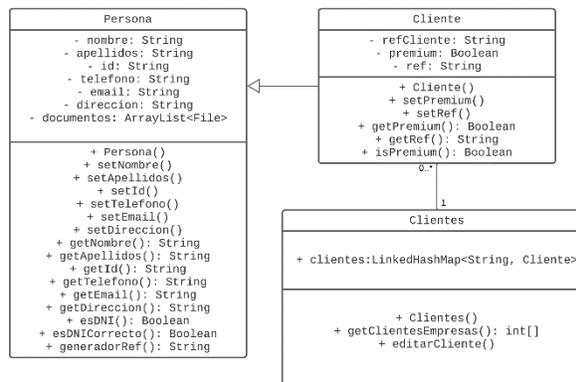


Ilustración 46 - Diagrama de Clase Clientes

Para acceder a la Agenda desde cualquier otro módulo usamos los botones del menú de navegación, al usar estos, el controlador de la aplicación realiza la petición de mapeo. La función llamada `agendaClientesModel` que solicita al controlador de la agenda que se visualice el apartado de clientes. El controlador a su vez genera la vista con los clientes, el número de clientes, número de empresas y los datos del usuario que realiza la petición.

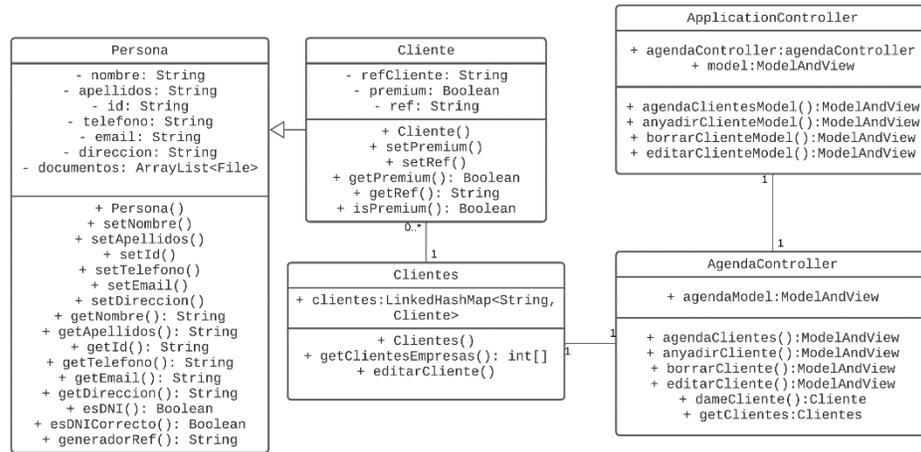
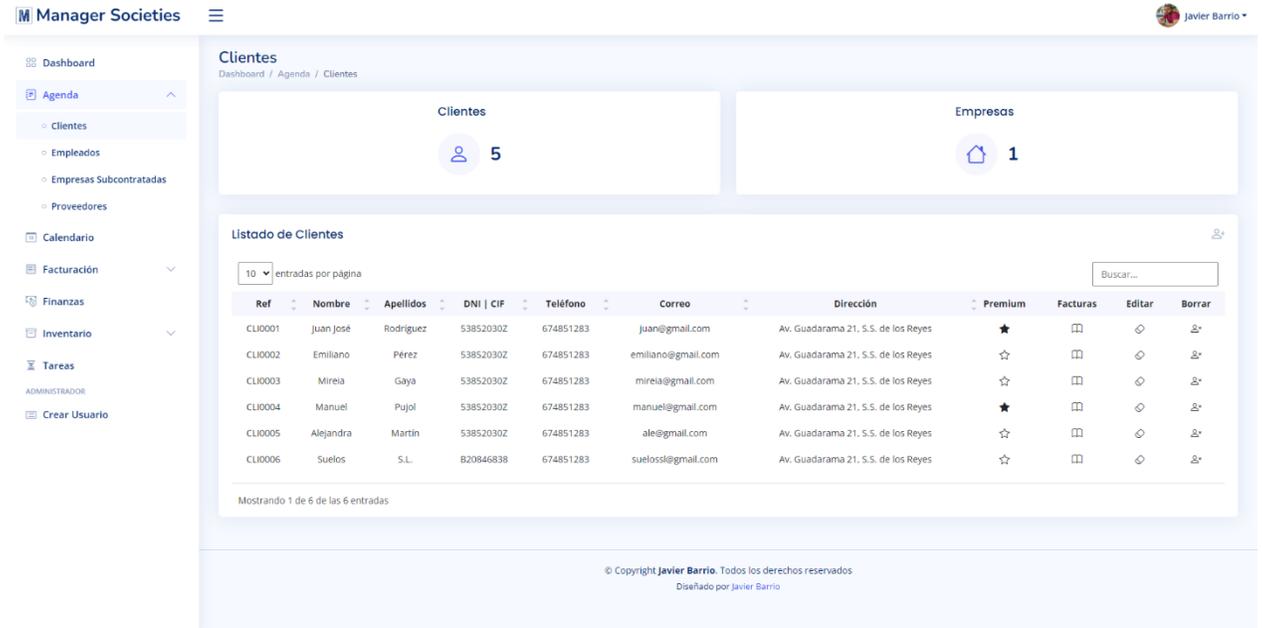


Ilustración 47 - Diagrama de Clases de Clientes

Además, el controlador de la aplicación posee cuatro métodos relacionados con la agenda de clientes que derivan en seis métodos. Estos nos permiten mostrar los clientes, añadir un cliente nuevo, borrarlo y editarlo.

El diagrama de la parte superior representa la relación entre las cinco clases, esta relación se repite en el mismo funcionamiento y métodos similares para cada sección de la agenda, y no repetiremos esto para cada uno de ellos, si no que explicaremos las diferencias si existiesen.



Manager Societies Javier Barrio

Clientes
Dashboard / Agenda / Clientes

Cientes 5 **Empresas** 1

Listado de Clientes

10 entradas por página Buscar...

Ref	Nombre	Apellidos	DNI CIF	Teléfono	Correo	Dirección	Premium	Facturas	Editar	Borrar
CLI0001	Juan José	Rodríguez	538520302	674851283	juan@gmail.com	Av. Guadarama 21, S.S. de los Reyes	★	☑	✎	🗑
CLI0002	Emiliano	Pérez	538520302	674851283	emiliano@gmail.com	Av. Guadarama 21, S.S. de los Reyes	☆	☑	✎	🗑
CLI0003	Mireia	Gaya	538520302	674851283	mireia@gmail.com	Av. Guadarama 21, S.S. de los Reyes	☆	☑	✎	🗑
CLI0004	Manuel	Pujol	538520302	674851283	manuel@gmail.com	Av. Guadarama 21, S.S. de los Reyes	★	☑	✎	🗑
CLI0005	Alejandra	Martin	538520302	674851283	ale@gmail.com	Av. Guadarama 21, S.S. de los Reyes	☆	☑	✎	🗑
CLI0006	Suelos	S.L.	B20846838	674851283	sueloski@gmail.com	Av. Guadarama 21, S.S. de los Reyes	☆	☑	✎	🗑

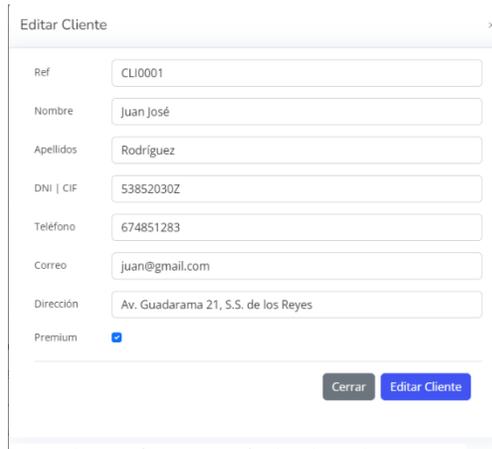
Mostrando 1 de 6 de las 6 entradas

© Copyright Javier Barrio. Todos los derechos reservados
Diseñado por Javier Barrio

Ilustración 48 - Sección de Clientes de la Agenda

Como se aprecia en la foto, el listado de clientes agrupa su referencia, nombre, apellidos, DNI o CIF, teléfono, correo, dirección, si es premium o no, y tres botones con diversas funcionalidades. El primero nos permite visualizar un listado de las facturas emitidas a dicho cliente, o crear una nueva, esta sección se explicará en detalle en el apartado de facturación.

En la parte superior existen dos contadores en los que se indica el número de clientes que son personas físicas o que poseen DNI y el número de clientes que son empresas, es decir, se han guardado con CIF. Al acceder al menú de edición se muestra un modal con los datos del cliente en cuestión, desde el modal se permite modificar todos los campos menos la referencia que es generada automáticamente.



Editar Cliente

Ref CLI0001

Nombre Juan José

Apellidos Rodríguez

DNI | CIF 53852030Z

Teléfono 674851283

Correo juan@gmail.com

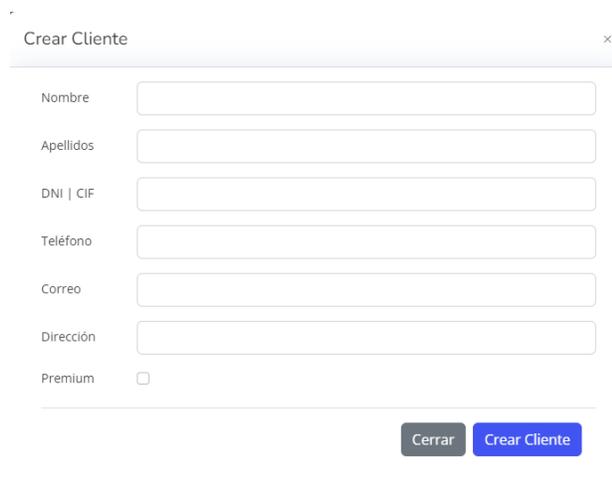
Dirección Av. Guadarama 21, S.S. de los Reyes

Premium

Cerrar Editar Cliente

Ilustración 49 - Sección de Editar Cliente

El botón de borrado efectúa el borrado del cliente en cuestión, mientras que el botón de añadir cliente muestra un modal similar al de edición, pero con los campos vacíos para ser rellenados.



Crear Cliente

Nombre

Apellidos

DNI | CIF

Teléfono

Correo

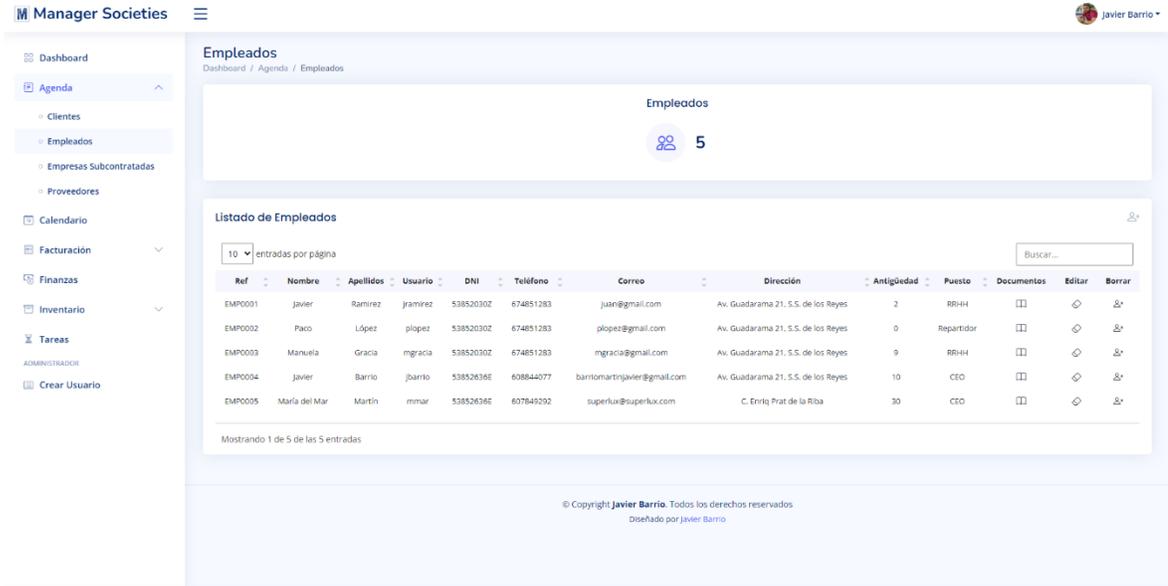
Dirección

Premium

Cerrar Crear Cliente

Ilustración 50 - Sección de Crear Cliente

Como se ha explicado el apartado de empleados, empresas subcontratadas y proveedores posee un comportamiento similar a los clientes con pequeños matices. En el caso de los empleados aparece un campo para el usuario si este tiene acceso a la aplicación o un -, en caso contrario. Además, presenta un campo numérico que indica la antigüedad en la empresa y su puesto en la misma. Por otro lado, dentro del desplegable de documentos podemos encontrar el contrato, y las nóminas, que se explicarán en la sección de facturación.



Empleados
Dashboard / Agenda / Empleados

Empleados 5

Listado de Empleados

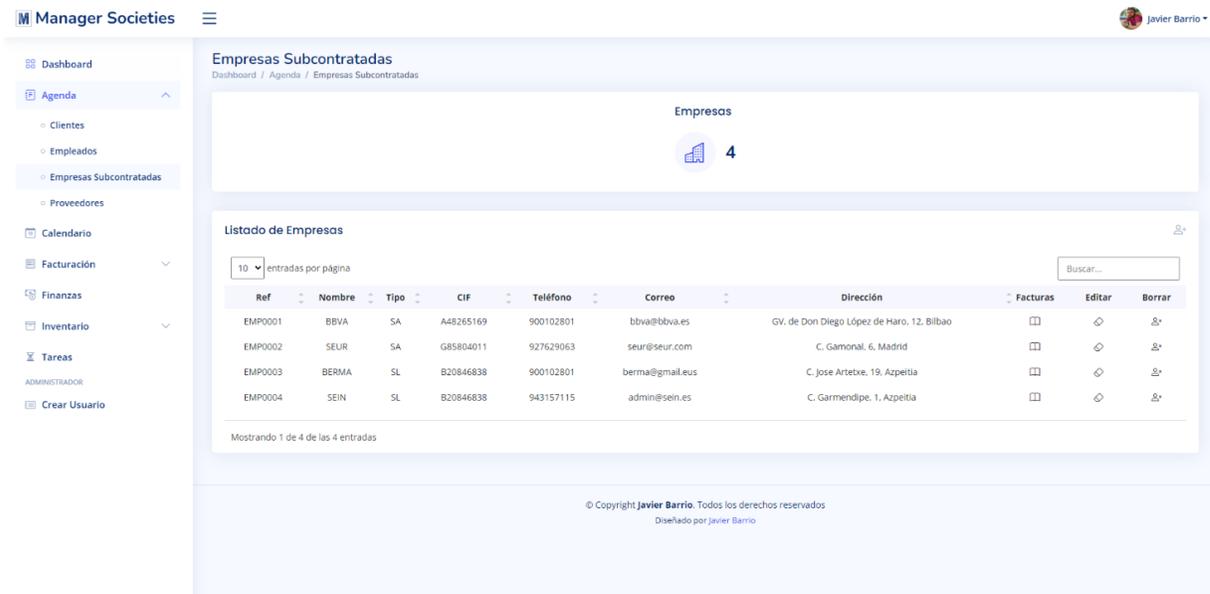
10 entradas por página

Ref	Nombre	Apellidos	Usuario	DNI	Teléfono	Correo	Dirección	Antigüedad	Puesto	Documentos	Editar	Borrar
EMP0001	Javier	Ramirez	jramirez	53852030Z	674851283	juan@gmail.com	Av. Guadarama 21, S.S. de los Reyes	2	RRHH	📄	✎	🗑️
EMP0002	Paco	López	plopez	53852030Z	674851283	plopez@gmail.com	Av. Guadarama 21, S.S. de los Reyes	0	Repartidor	📄	✎	🗑️
EMP0003	Manuela	Gracia	mgracia	53852030Z	674851283	mgracia@gmail.com	Av. Guadarama 21, S.S. de los Reyes	9	RRHH	📄	✎	🗑️
EMP0004	Javier	Barrio	jbarrio	53852636E	608844077	barriomartinjavier@gmail.com	Av. Guadarama 21, S.S. de los Reyes	10	CEO	📄	✎	🗑️
EMP0005	María del Mar	Martín	mymar	53852636E	607849202	superlux@superlux.com	C. Enriq Prat de la Riba	30	CEO	📄	✎	🗑️

Mostrando 1 de 5 de las 5 entradas

© Copyright Javier Barrio. Todos los derechos reservados
Diseñado por Javier Barrio

Ilustración 52 - Sección de Empleados



Empresas Subcontratadas
Dashboard / Agenda / Empresas Subcontratadas

Empresas 4

Listado de Empresas

10 entradas por página

Ref	Nombre	Tipo	CIF	Teléfono	Correo	Dirección	Facturas	Editar	Borrar
EMP0001	BBVA	SA	A48265169	900102801	bbva@bbva.es	GV. de Don Diego López de Haro, 12, Bilbao	📄	✎	🗑️
EMP0002	SEUR	SA	G85804011	927629063	seur@seur.com	C. Gamonal, 6, Madrid	📄	✎	🗑️
EMP0003	BERMA	SL	B20846838	900102801	berma@gmail.es	C. Jose Artetxe, 19, Azpetia	📄	✎	🗑️
EMP0004	SEIN	SL	B20846838	943157115	admin@sein.es	C. Garmendipe, 1, Azpetia	📄	✎	🗑️

Mostrando 1 de 4 de las 4 entradas

© Copyright Javier Barrio. Todos los derechos reservados
Diseñado por Javier Barrio

Ilustración 51 - Sección de Empresas Subcontratadas

La sección de empresas subcontratadas cambia el campo apellido por tipo de empresa, el DNI por el CIF, y volvemos a tener un listado de facturas asociadas a los trabajos o servicios efectuados por dicha empresa.

En el caso de los proveedores es un calco del funcionamiento de empresas subcontratadas, puesto que un proveedor es una empresa.

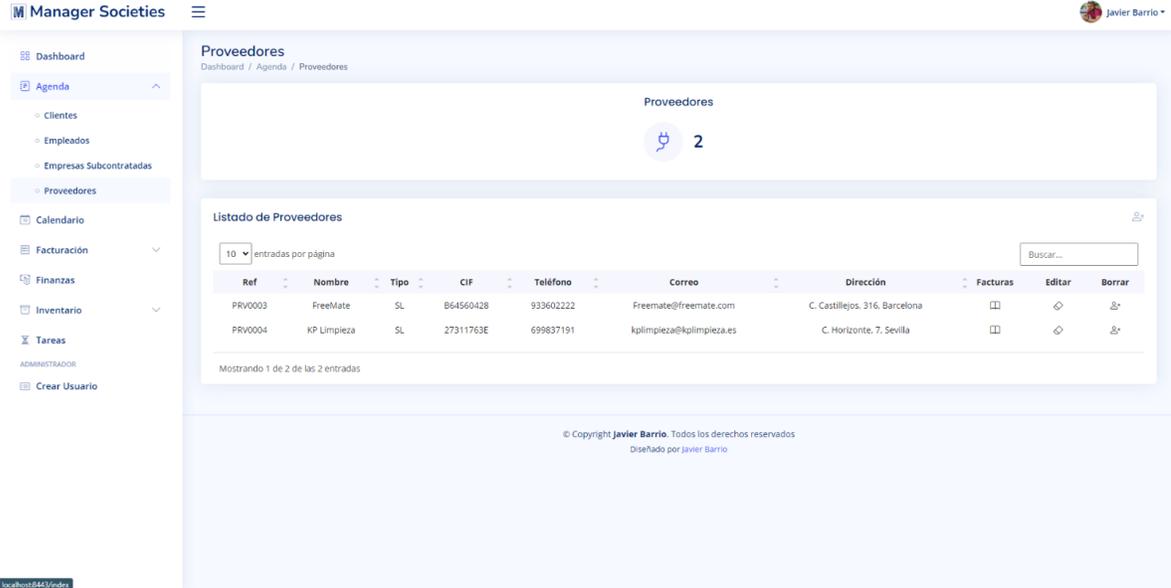


Ilustración 53 - Sección de Proveedores

5.2.8. Calendario:

El módulo del calendario en la aplicación es visible para todos los usuarios. Para esta sección se ha creado una clase calendario que se caracteriza por un día, mes, año, fecha actual, meses y las fechas de un mes desglosado en semanas.

Para guardar los meses se ha creado una clase adicional Mes que se caracteriza por el nombre del mes y el número de días correspondientes al mes, está clase solo tiene su constructor y sus métodos setter y getter.

El calendario posee sus constructores, setters, getters, un método para obtener la fecha actual, y la clasificación de los meses distribuido en las semanas del mes, y un par de métodos que dada una fecha te devuelve la fecha anterior y la posterior.

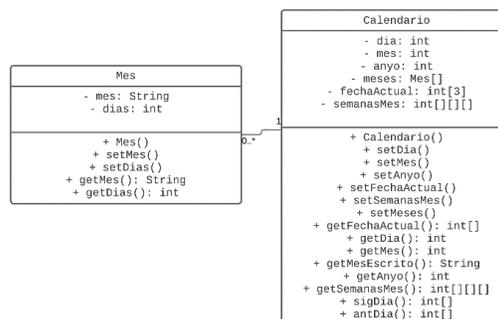


Ilustración 54 - Diagrama de Clases del Calendario

Para poder mostrar correctamente el calendario según el diseño planteado necesitamos un par de métodos específicos que se explicarán a continuación. El primero denominado `setMeses`, únicamente verifica si el año introducido como parámetro es bisiesto o no, y genera un array de la clase `Mes` creada previamente indicando el nombre y días del mes en cuestión.

El método `setSemanasMes` es más complejo, dado un mes de un año en cuestión, concretamente del que queremos mostrar por pantalla, obtenemos el primer día del mes y comprobamos en que día de la semana comienza este mes. Esto se hace para completar el mes mostrado con los días del mes anterior y el mes siguiente de tal forma que la matriz siempre este completa sin huecos en blanco.

Para poder rellenar estos huecos debemos saber cuántos días del mes anterior hay que mostrar y cuantos del mes siguiente. Finalmente, lo que devuelve este método es una matriz de tantas filas como semanas tenga el mes en cuestión, siempre siete columnas, ya que las semanas siempre tienen siete días, y en cada celda se guarda un array de tres posiciones con el día, mes y año, del día en cuestión.

Para poder cargar el calendario el controlador de la aplicación, solicita al controlador del calendario la creación de la vista de este. El controlador del calendario posee dos constructores, uno para renderizar el calendario según la fecha actual, y uno, dado un mes y un año específico, esto es debido a la posibilidad de navegación del calendario a fechas anteriores y futuras. Además, de la muestra del calendario, este permite la creación de eventos para la gestión de las tareas de la empresa.

Para ello, la aplicación posee una clase `Evento`, caracterizada por una referencia de evento, la referencia de la tarea asociada, si es que la tuviese, un título, fecha de inicio y de fin, y la hora de inicio y fin. Aparte de sus constructores, setter y getter, posee un par de funciones auxiliares para formatear fechas básicos.

La lógica de la aplicación no es que el `Calendario` sea el encargado de gestionar los eventos, por lo que, realmente el calendario no “tiene” eventos, estos estarán guardados en la base de datos y es el controlador del calendario mediante su interfaz correspondiente la que suministra al controlador de la aplicación los eventos, que son mostrados por el Front-End, en el `Calendario`. Así, permitimos la encapsulación y la separación del código.

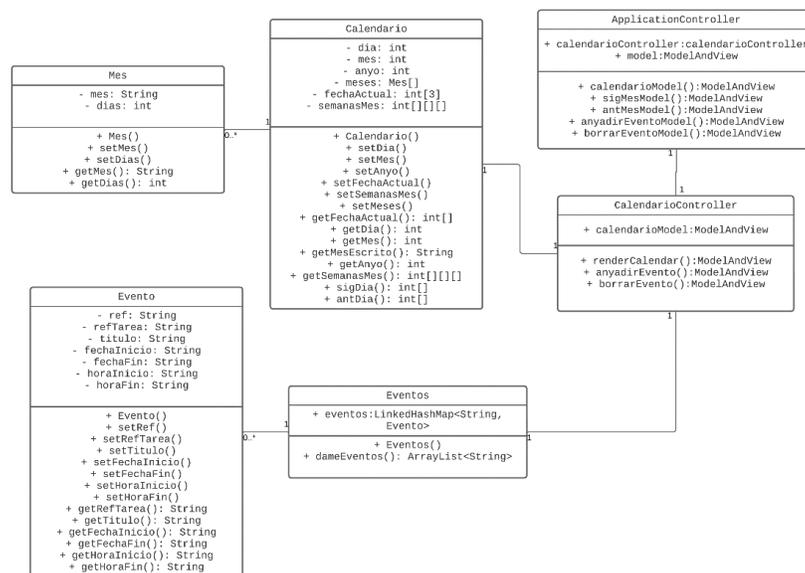


Ilustración 55 - Diagrama de Clases del Calendario

Por otro lado, el controlador del calendario puede tanto añadir como borrar eventos, para ello, es el controlador de la aplicación el que le solicita al controlador del calendario la creación y borrado y este el que gestiona dichos métodos.

Los eventos solo son mostrados a los perfiles administradores, o en caso de un perfil empleado solo podrán ver en el calendario los eventos asociados a tareas propias, esto se explicará en el apartado de tareas con más detalle.

A nivel usabilidad de usuario en la sección Calendario visualizamos este, y cuatro botones, uno que nos permite ir al día actual, otro para añadir eventos, y dos botones de navegación. Para borrar un evento debemos clicar en él, se abrirá un desplegable donde aparecerá la información del evento y el botón para borrarlo.

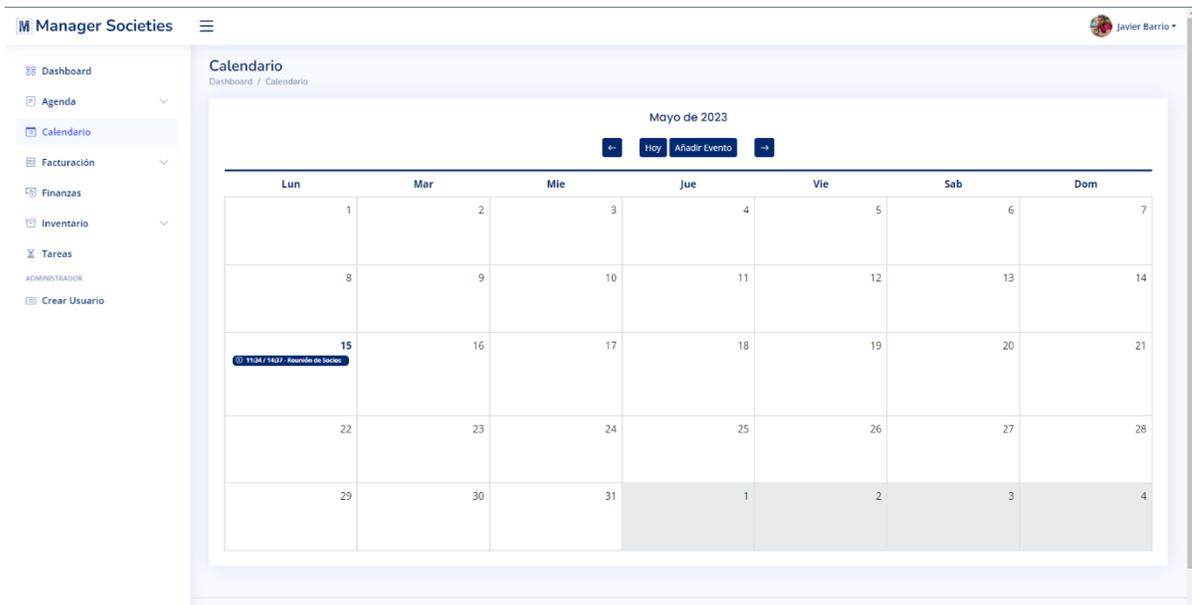


Ilustración 56 - Sección de Calendario

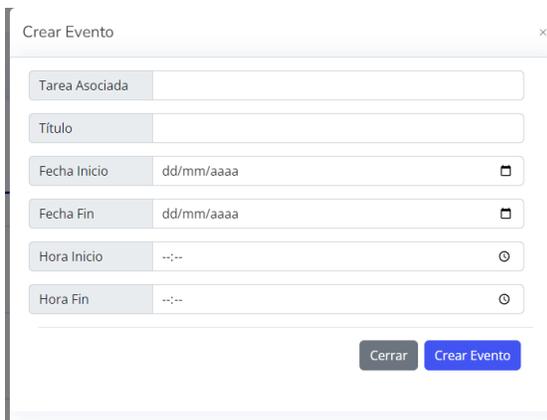
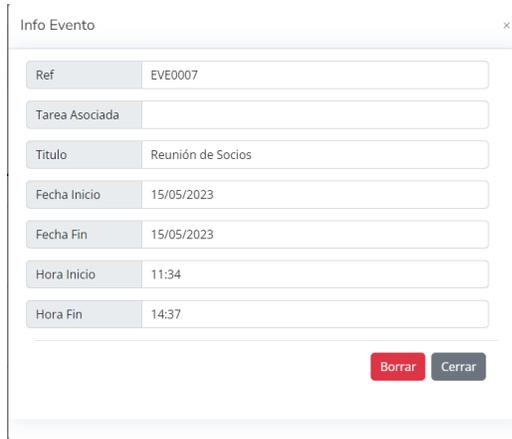


Ilustración 57 - Sección de Creación de Evento



The screenshot shows a web form titled "Info Evento" with a close button (x) in the top right corner. The form contains several input fields with labels and values:

Ref	EVE0007
Tarea Asociada	
Título	Reunión de Socios
Fecha Inicio	15/05/2023
Fecha Fin	15/05/2023
Hora Inicio	11:34
Hora Fin	14:37

At the bottom right of the form, there are two buttons: "Borrar" (red) and "Cerrar" (grey).

Ilustración 58 - Información de Evento

5.2.9. Inventario:

El módulo del inventario en la aplicación es exclusivo del rol de administrador y agrupa cinco secciones, un apartado para herramientas, máquinas, materiales, productos y vehículos. Aunque todos ellos poseen un diseño común mantenido a lo largo de la aplicación como en el módulo de la agenda, tienen sus pequeñas variaciones.



Ilustración 59 - Navegación del Inventario

Para el correcto funcionamiento del inventario se ha generado una clase Objeto con características comunes al resto de clases del módulo. Un objeto se caracteriza por su marca, modelo y un listado de facturas.

Dentro de las funciones de un Objeto podemos encontrar sus setter y getter, necesarios para la modificación y obtención de los datos desde otras clases Java. Por último, se ha diseñado una función imprescindible denominada generadorRef que dado un número y un String elabora una combinación de ambos en la que obtendremos una referencia única para cada instancia de nuestra aplicación.

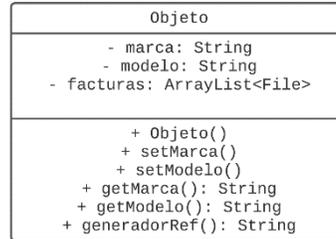


Ilustración 60 - Diagrama de Clase Objeto

La primera sección del Inventario es la sección de Herramientas, para su diseño se ha elaborado el archivo CSS, en el que entraremos en detalle a continuación, la clase Herramienta hereda las características de la clase Objeto, pero además posee un generadorId numérico, variable que aparecerá en todas las clases, ya que, permite controlar el número exacto de instancias para generar la próxima referencia. Este será eliminado al crear la base de datos, ya que se consultará a ella cuál será dicho número.

Por otro lado, una constante de texto que denomina la referencia para las herramientas *HER*, un valor real para el precio, un entero para la cantidad en el inventario, y la propia referencia completa. La clase Herramienta posee tres constructores disponibles, el primero en el que no se le pasa ningún valor, un segundo en el que se construye la instancia a partir de todas sus variables menos la referencia que será generada dentro del propio constructor, y, en último lugar, otro en el que la referencia es dada.

El resto de las funciones de la clase son getter y setter. A partir de la clase Herramienta se genera una clase Herramientas que servirá de base de datos provisional hasta que la aplicación tenga la suya propia, en ella podemos visualizar una variable clientes de tipo LinkedHashMap, se ha utilizado la combinación de la tabla Hash más la lista doblemente enlazada por su capacidad de ordenar y controlar los datos en función de su clave valor, siendo su clave la referencia única generada para cada instancia y el valor la instancia cliente completa.

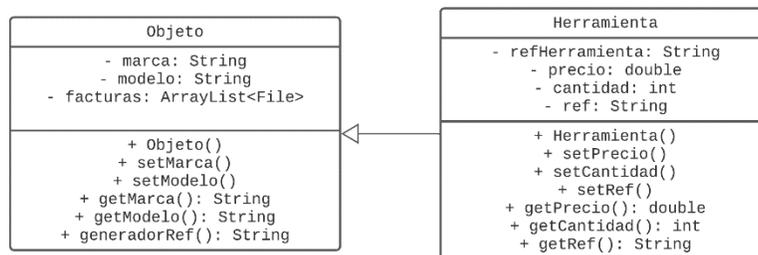


Ilustración 61 - Diagrama de Clases Herramienta

La clase Herramientas migrará a una clase Interface cuando la base de datos esté operativa, esta clase posee la función editarHerramienta que permite la modificación de esta mediante la referencia única de la misma, encontrando a la herramienta por dicha referencia en nuestra lista y aprovechando el constructor que incluía está para crear una nueva instancia con los datos modificados.

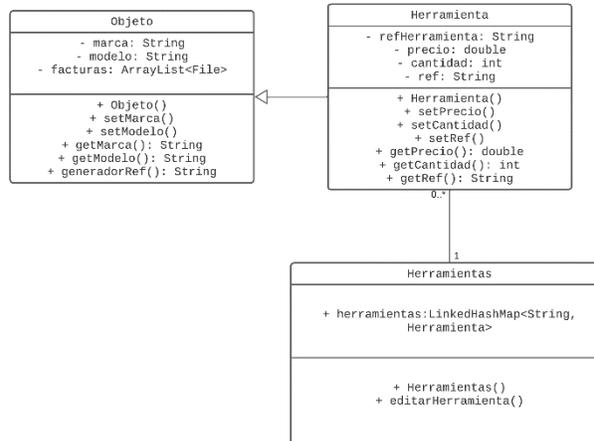


Ilustración 62 - Diagrama de Clases Herramientas

Para acceder al Inventario desde cualquier otro módulo usamos los botones del menú de navegación, al usar estos, el controlador de la aplicación realiza la petición de mapeo. La función llamada es `inventarioHerramientasModel` que solicita al controlador de la agenda que se visualice el apartado de herramientas. El controlador a su vez genera la vista con las herramientas, el número de estas y los datos del usuario que realiza la petición.

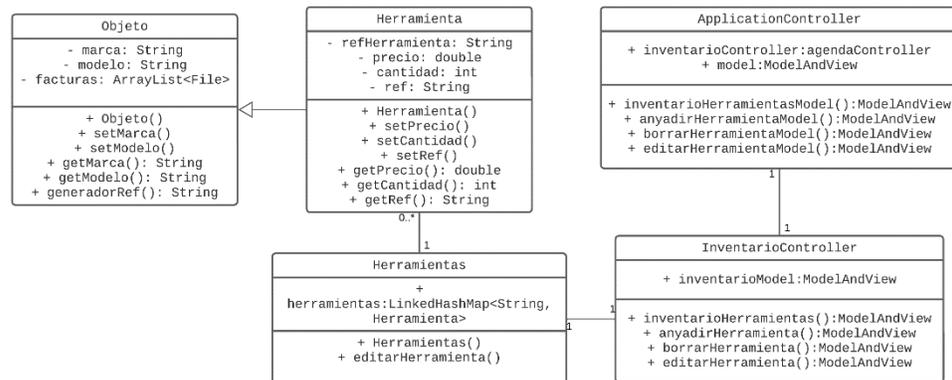
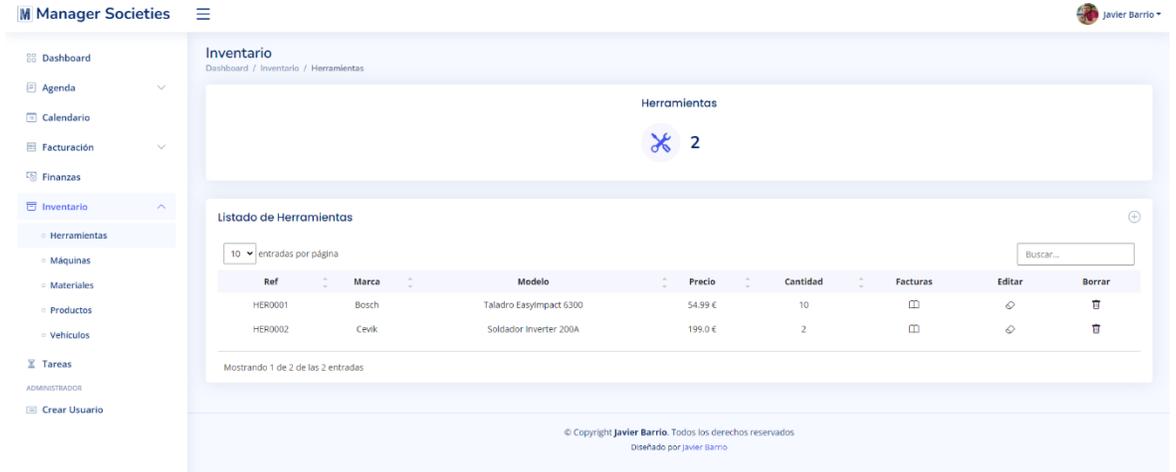


Ilustración 63 - Diagrama de Clases de Herramientas

Además, el controlador de la aplicación posee cuatro métodos relacionados con el inventario de herramientas que derivan en cuatro métodos. Estos nos permiten mostrar las herramientas, añadir una herramienta nueva, borrarla y editarla.

El diagrama de la parte superior representa la relación entre las cinco clases, esta relación se repite con el mismo funcionamiento y métodos similares para cada sección del inventario, y no repetiremos esto para cada uno de ellos, si no que explicaremos las diferencias si existiesen.



Manager Societies Javier Barrio

Inventario
Dashboard / Inventario / Herramientas

Herramientas
✂ 2

Listado de Herramientas

10 entradas por página Buscar...

Ref	Marca	Modelo	Precio	Cantidad	Facturas	Editar	Borrar
HER0001	Bosch	Taladro EasyImpact 6300	54,99 €	10	📄	🔍	🗑
HER0002	Cevik	Soldador Inverter 200A	199,0 €	2	📄	🔍	🗑

Mostrando 1 de 2 de las 2 entradas

© Copyright **Javier Barrio**. Todos los derechos reservados.
Diseñado por **javier barrio**

Ilustración 64 - Sección de Herramientas del Inventario

Como se aprecia en la foto, el listado de herramientas agrupa su referencia, marca, modelo, precio, cantidad, y tres botones con diversas funcionalidades. El primero nos permite visualizar un listado de las facturas emitidas en la compra de dicha herramienta, o crear una nueva, esta sección se explicará en detalle en el apartado de facturación.

En la parte superior existe un contador en el que se indica el número de herramientas. Al acceder al menú de edición se muestra un modal con los datos de la herramienta en cuestión, desde el modal se permite modificar todos los campos menos la referencia que es generada automáticamente.



Editar Herramienta ×

Marca

Marca

Modelo

Precio

Cantidad

Cerrar **Editar Herramienta**

Ilustración 65 - Sección de Editar Herramienta

El botón de borrado efectúa el borrado de la herramienta en cuestión, mientras que el botón de añadir herramienta muestra un modal similar al de edición, pero con los campos vacíos para ser rellenados.

Crear Herramienta ×

Marca

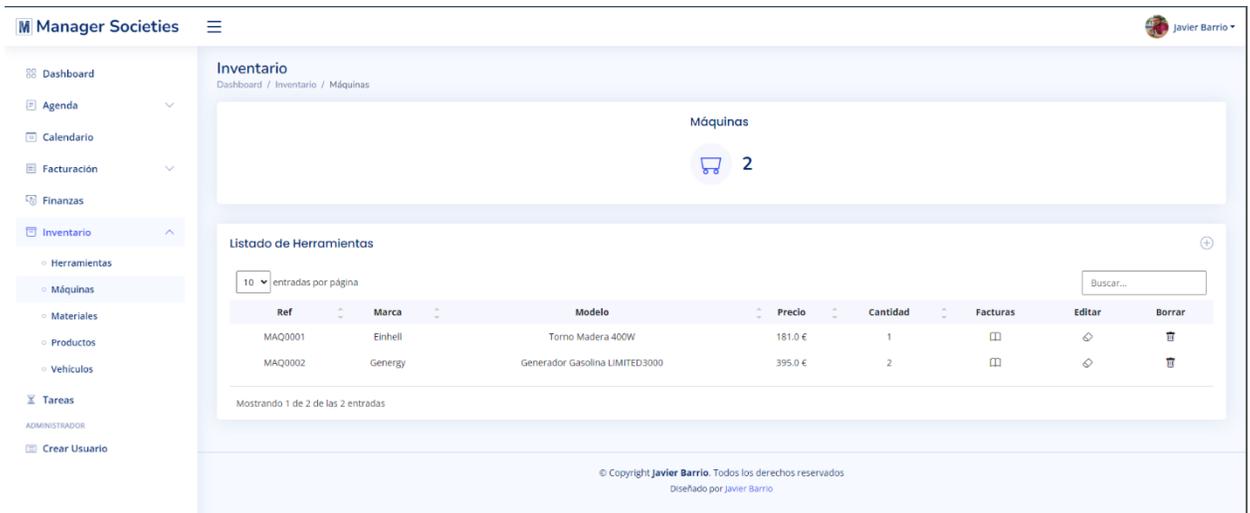
Modelo

Precio

Cantidad

Ilustración 66 - Sección de crear Herramienta

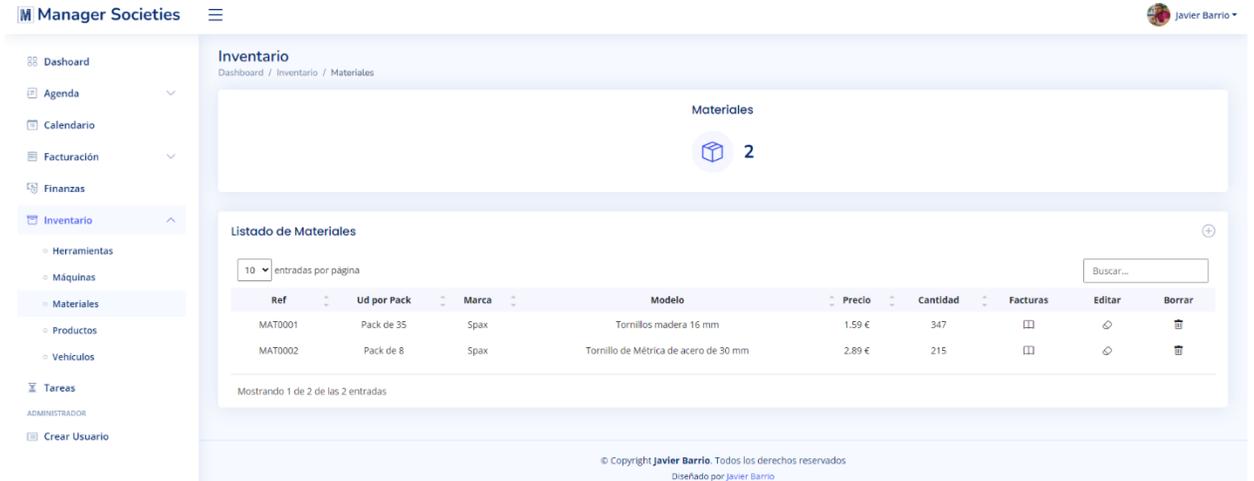
Como se ha explicado el apartado de herramientas, máquinas, materiales, productos y vehículos poseen un comportamiento similar a las herramientas con pequeños matices. En el caso de los máquinas no existe diferenciación alguna.



Ref	Marca	Modelo	Precio	Cantidad	Facturas	Editar	Borrar
MAQ0001	Einhell	Torno Madera 400W	181.0 €	1	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="✖"/>
MAQ0002	Genenergy	Generador Gasolina LIMITED3000	395.0 €	2	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="✖"/>

Ilustración 67 - Sección de Máquinas

En los materiales se añade un campo de unidades por pack, ya que la compra suele ser en packs.



Inventario
Dashboard / Inventario / Materiales

Materiales
2

Listado de Materiales

10 entradas por página

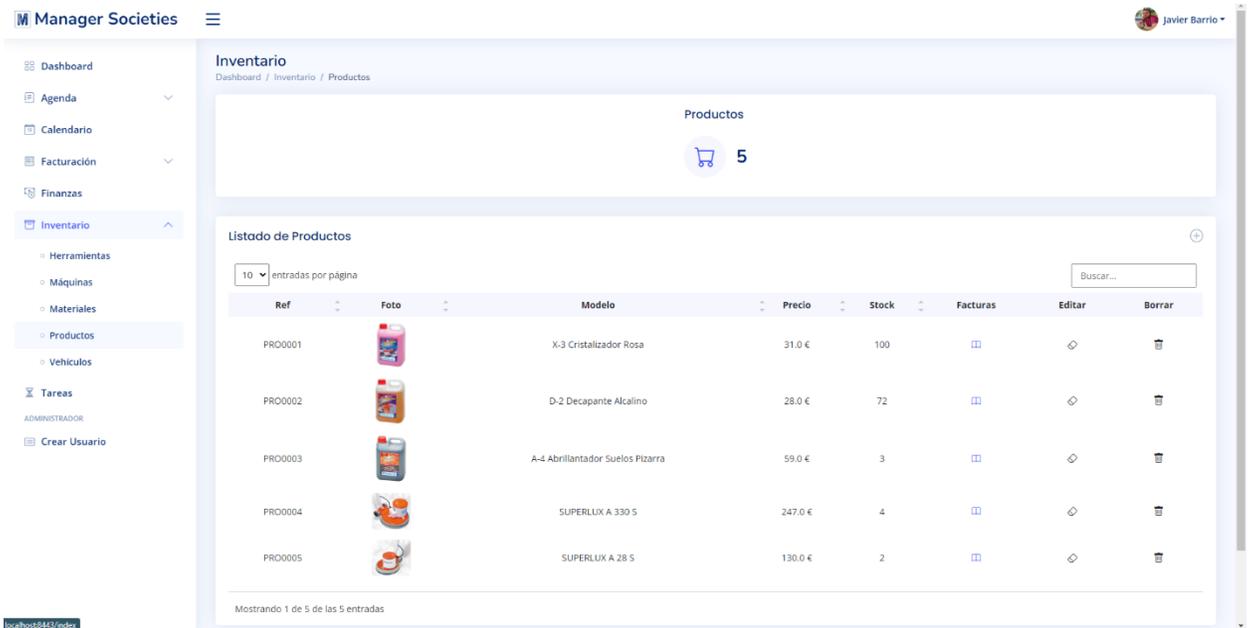
Ref	Ud por Pack	Marca	Modelo	Precio	Cantidad	Facturas	Editar	Borrar
MAT0001	Pack de 35	Spax	Tornillos madera 16 mm	1.59 €	347	<input type="checkbox"/>		
MAT0002	Pack de 8	Spax	Tornillo de Métrica de acero de 30 mm	2.89 €	215	<input type="checkbox"/>		

Mostrando 1 de 2 de las 2 entradas

© Copyright Javier Barrio. Todos los derechos reservados
Diseñado por Javier Barrio

Ilustración 68 - Sección de Materiales

Los productos son un apartado un tan especial, ya que, son los productos que vende la empresa que hace uso de la aplicación, por lo que se añade una foto del producto, y la cantidad cambia por el stock en el almacén.



Inventario
Dashboard / Inventario / Productos

Productos
5

Listado de Productos

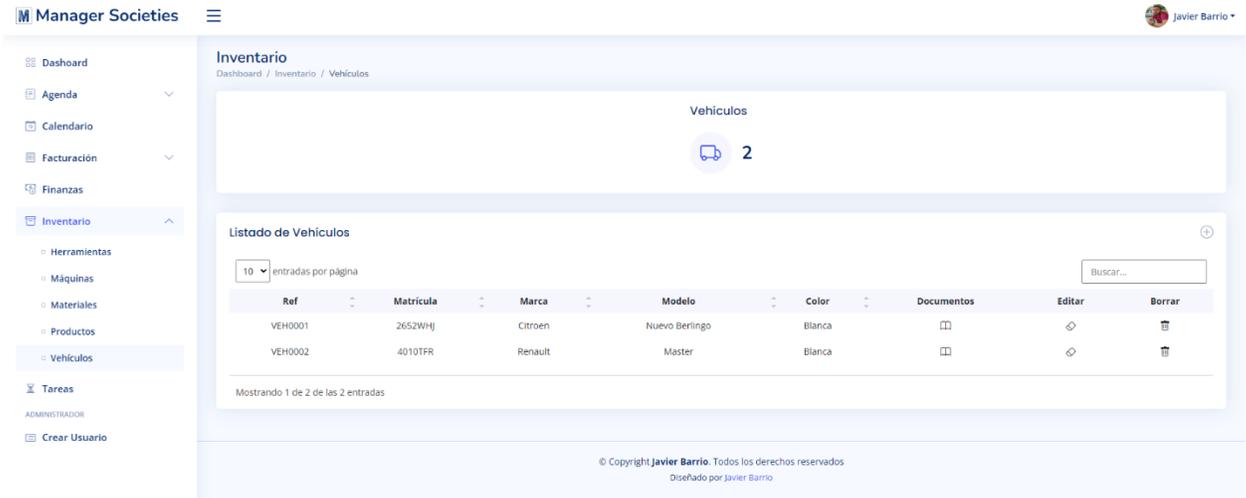
10 entradas por página

Ref	Foto	Modelo	Precio	Stock	Facturas	Editar	Borrar
PRO0001		X-3 Cristalizador Rosa	31.0 €	100	<input type="checkbox"/>		
PRO0002		D-2 Decapante Alcalino	28.0 €	72	<input type="checkbox"/>		
PRO0003		A-4 Abrillantador Suelos Pizarra	59.0 €	3	<input type="checkbox"/>		
PRO0004		SUPERLUX A 330 S	247.0 €	4	<input type="checkbox"/>		
PRO0005		SUPERLUX A 28 S	130.0 €	2	<input type="checkbox"/>		

Mostrando 1 de 5 de las 5 entradas

Ilustración 69 - Sección de Productos

Finalmente, los vehículos poseen la matrícula y el color de estos. Por otro lado, dentro del desplegable de documentos podemos encontrar y añadir documentos de interés para cada sección, cómo facturas, o papeles de los vehículos, estos, se explicarán en la sección de facturación.



Inventario
Dashboard / Inventario / Vehículos

Vehículos 2

Listado de Vehículos

10 entradas por página

Buscar...

Ref	Matrícula	Marca	Modelo	Color	Documentos	Editar	Borrar
VEH0001	2652WHJ	Citroen	Nuevo Berlingo	Blanca	📄	✎	🗑️
VEH0002	4010TFR	Renault	Master	Blanca	📄	✎	🗑️

Mostrando 1 de 2 de las 2 entradas

© Copyright **Javier Barrio**. Todos los derechos reservados
Diseñado por Javier Barrio

Ilustración 70 - Sección de Vehículos

5.2.10. Tareas:

El módulo de las tareas en la aplicación es visible tanto para el rol de administrador como de empleado y no posee subsecciones. Mantiene el diseño mostrado a lo largo de la aplicación.

Para el correcto funcionamiento de la agenda se ha generado una clase Tarea. Una Tarea se caracteriza con su referencia, cliente asociado, fecha de inicio y fin, hora de inicio, empleados que la llevarán a cabo, un gasto que supone la tarea, un inventario para su realización, información de interés, y el estado de esta.

Dentro de las funciones de una Tarea podemos encontrar los setter y getter, necesarios para la modificación y obtención de los datos desde otras clases Java. Además, de un par de funciones para formatear y quitar formato a fechas. Pero, de interés las funciones contieneEmpleado y contieneInventario que comprueban si una tarea cuenta con empleados o inventario asignado.

Por último, se ha diseñado una función imprescindible denominada generadorRef que dado un número y un String elabora una combinación de ambos en la que obtendremos una referencia única para cada instancia de nuestra aplicación.

La clase Tareas migrará a una clase Interface cuando la base de datos esté operativa, esta clase posee la función editarTarea que permite la modificación de este mediante la referencia única del mismo, encontrando la tarea por dicha referencia en nuestra lista y aprovechando el constructor que incluía está para crear una nueva instancia con los datos modificados.

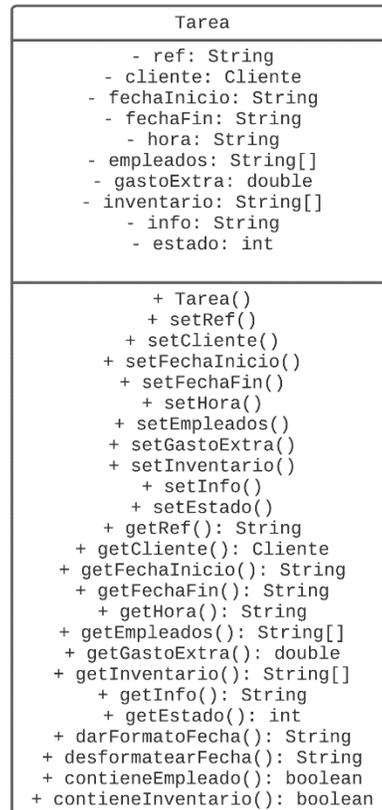


Ilustración 71 - Diagrama de Clase Tarea

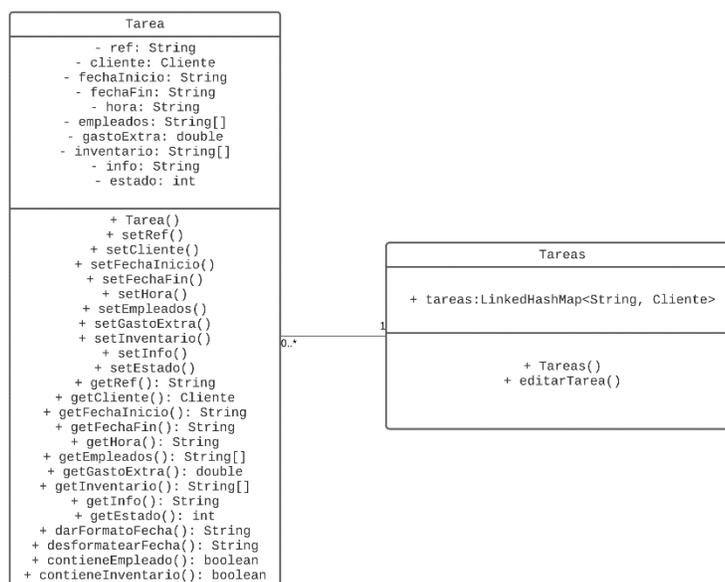


Ilustración 72 - Diagrama de Clase Tareas

Para acceder a las Tareas desde cualquier otro módulo usamos los botones del menú de navegación, al usar estos, el controlador de la aplicación realiza la petición de mapeo. La función llamada es tareasModel que solicita al controlador de las tareas que se visualice el apartado de tareas. El controlador a su vez genera la vista con estas, la cantidad de estas, y los datos del usuario que realiza la petición.

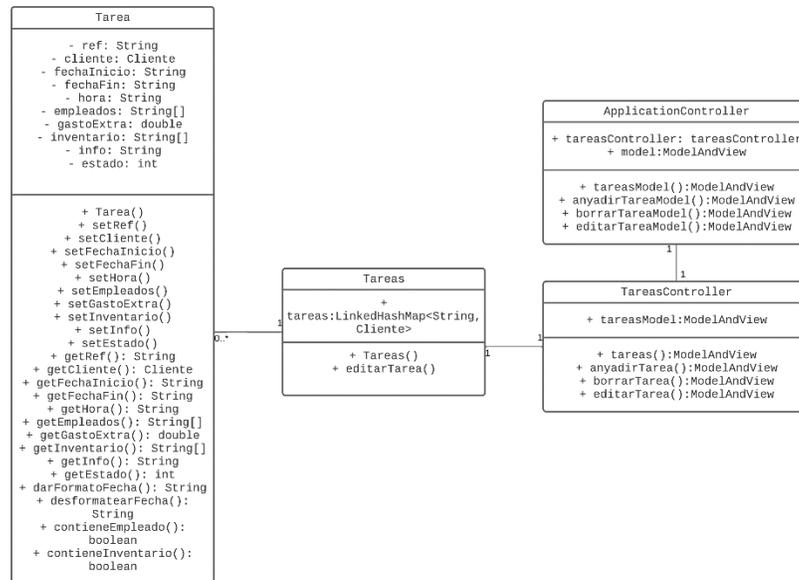
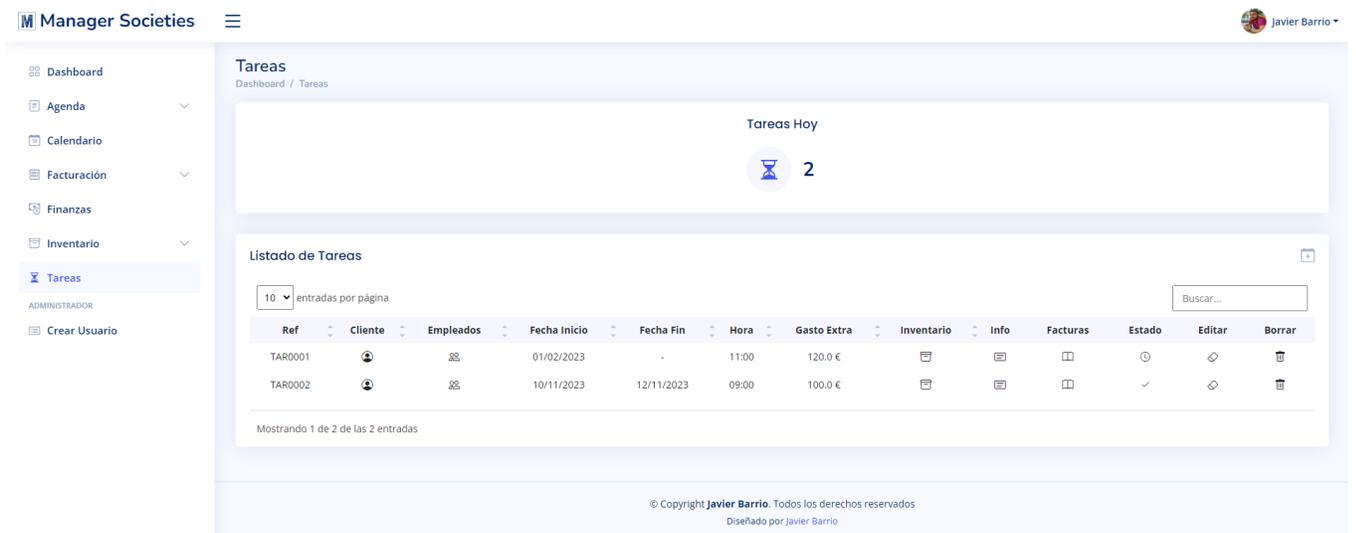


Ilustración 73 - Diagrama de Clases de Tareas

Además, el controlador de la aplicación posee tres métodos relacionados con las tareas. Estos nos permiten mostrar las tareas, añadir una nueva, borrarla y editarla. Al añadir una tarea automáticamente esta se muestra en el calendario para los administradores y todos aquellos empleados asociados a esta.



The screenshot shows the 'Tareas' section of the Manager Societies application. The interface includes a sidebar with navigation options like Dashboard, Agenda, Calendario, Facturación, Finanzas, Inventario, and Tareas (selected). The main content area displays 'Tareas Hoy' with a count of 2 and a 'Listado de Tareas' table. The table has columns for Ref, Cliente, Empleados, Fecha Inicio, Fecha Fin, Hora, Gasto Extra, Inventario, Info, Facturas, Estado, Editar, and Borrar. Two tasks are listed: TAR0001 and TAR0002. The footer contains copyright information for Javier Barrio.

Ilustración 74 - Sección de Tareas

La sección de tareas es una de las más complejas debido a sus interrelaciones y extensión. Por esto, debido a la cantidad de información a mostrar se ha decidido a utilizar modales desplegables para mostrar dicha información a través de botones.

Por ejemplo, para mostrar la información del cliente o de los empleados que van a realizar el trabajo se utilizan dos modales mostrables al hacer clic sobre ellos.

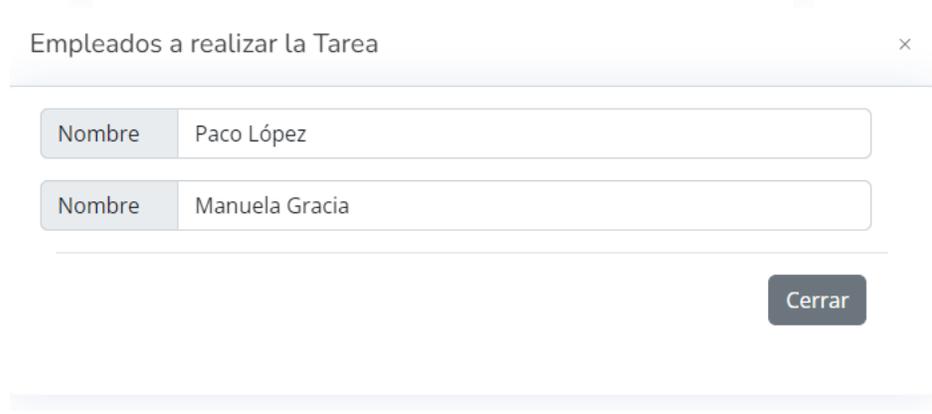


Info Cliente ×

Ref	CLI0001
Nombre	Juan José Rodríguez
DNI CIF	53852030Z
Teléfono	674851283
Correo	juan@gmail.com
Dirección	Av. Guadarama 21, S.S. de los Reyes

Cerrar

Ilustración 75 - Desplegable Información Cliente



Empleados a realizar la Tarea ×

Nombre	Paco López
Nombre	Manuela Gracia

Cerrar

Ilustración 76 - Desplegable Información Empleados

Sucede lo mismo con el inventario, la información, o las facturas de la tarea en cuestión. Por último, hay que mencionar que el estado es de un reloj si no se ha acabado el trabajo, de un tick en caso de estar finalizado, y de un doble tick en caso de estar finalizado y cobrado. De forma similar al resto de la aplicación tenemos un botón de creación y otro de edición.

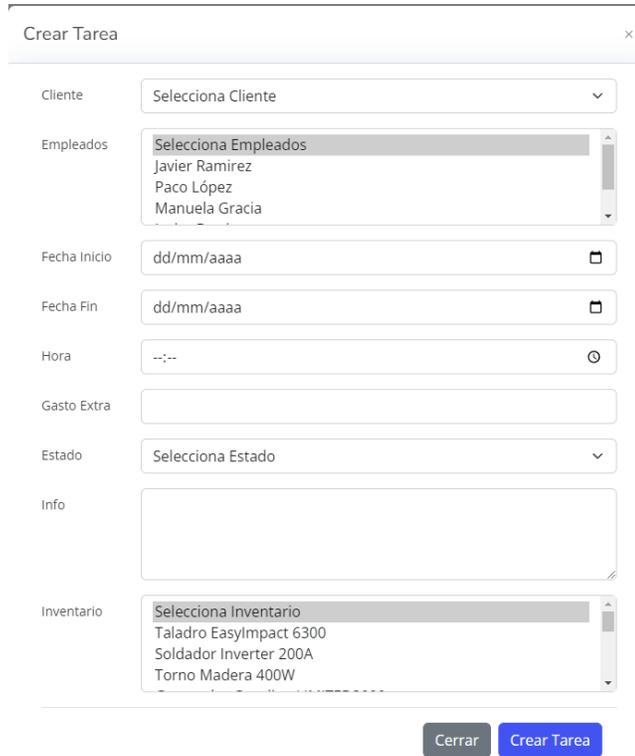


Ilustración 77 - Desplegable Creación de Tarea

5.2.11. Migración a Base de Datos:

Para poder utilizar una base de datos hemos añadido al archivo `application.properties` la url de la base de datos creada previamente en SQL, así como el usuario, contraseña, la ubicación del Driver de mysql, el dialecto utilizado de hibernate, y el tipo de ddl que recibe la base de datos, en este caso se ha utilizado la configuración de actualización.

Además, se ha añadido al `pom.xml` las dependencias de `spring jpa` y de `mysql connector` para permitir la conexión con la base de datos. Utilizando la herramienta `MySQLWorkbench` hemos generado el esquema de la base de datos para poder generar las futuras tablas y almacenar los datos.

Para explicar la migración de la aplicación a base de datos se va a tomar de ejemplo la sección de agenda, concretamente el apartado de clientes. En un primer momento se han reestructurado los directorios de esta de la forma `modulo > service | submodulo > entidad | repository`.

Dentro del subdirectorio `entidad` encontramos la clase `Cliente` para el ejemplo dado, dentro del subdirectorio `repository` el repositorio de clientes, y en el subdirectorio `service` la migración del controlador de la agenda al servicio de la agenda.

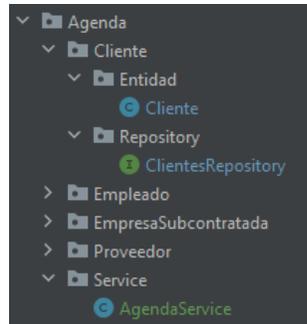


Ilustración 78 - Refactor de Directorios de la Agenda

La clase Cliente recibe la anotación Entity, esta anotación indica a Spring Boot que estamos ante una entidad JPA, además, se añade la anotación Table, con el nombre clientes, para indicar que existirá una tabla de clientes con dicho nombre.

Dentro de las variables de Cliente encontramos la anotación Id y GeneratedValue que nos permite dar un id único generado por la base de datos para cada entidad de la base de datos. Por otro lado, cada variable almacenada en base de datos recibe la anotación Column con un nombre de referencia para la tabla.

```
Javibarrío6
@Entity
@Table(name = "clientes")
public class Cliente implements Serializable {

    // Inicio Variables

    no usages
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int generadorId;
    1 usage
    private static final String refCliente = "CLI";

    2 usages
    @Column(name = "nombre")
    private String nombre;
}
```

Ilustración 79 - Clase Cliente

La clase ClientesRepository añadida es una interfaz que posee la anotación Repository para indicar a Spring Boot que esta clase posee las funciones de repositorio, además, extiende la clase JpaRepository. En este tipo de clase encontraremos funciones que realizan consultas a base de datos y nos traen datos para ser utilizados en la aplicación.

Existe una última anotación de interés en esta clase que es Query, que nos permite añadir una consulta personalizada a la base de datos que nos devuelve los datos solicitados a través de una función de java.

Para finalizar, el antiguo controlador de la agenda se ha modificado por un Service que recibe dicha anotación, este pierde los métodos de adicción, borrado y edición que pasan al controlador de la aplicación, pero mantiene el mapeado de cada submódulo de la Agenda.

5.2.12. Dashboard:

En Dashboard es la sección mostrada por defecto y la única sección que puede ver un usuario sin rol de administrador. Dentro del Dashboard se pueden visualizar ocho subsecciones. Las primeras son mostradas conjuntamente, la sección del dashboard es una sección de consulta, en la que no se modifican datos y sirve de resumen en el mostrado de datos de posible interés.

Estas tres primeras subsecciones son tres contadores sobre las ventas, ingresos y número de clientes, las dos primeras poseen tres filtros para mostrar el número de ventas de hoy, de este mes o de este año, en el mismo caso los ingresos, el número de clientes refleja los clientes del año.



Ilustración 80 - Sección de Ventas, Ingresos y Clientes del Dashboard

A continuación de estas tres subsecciones encontramos otra que refleja las ventas recientes del día de hoy, se pueden ver la referencia de la factura, el cliente facturado, los productos comprados y el precio total facturado.

Ventas Recientes | Hoy

10 entradas por página

Ref	Cliente	Producto	Precio
FAC0045	Suelos SL	D-2 Decapante Alcalino D-2 Decapante Alcalino D-2 Decapante Alcalino D-2 Decapante Alcalino D-2 Decapante Alcalino SUPERLUX A 330 S	3251.27 €

Mostrando 1 de 1 de las 1 entradas

Ilustración 81 - Ventas Recientes del Dashboard

Junto a la subsección de ventas recientes tenemos una que refleja los productos más vendidos por la empresa, ya que esto nos permite saber que productos son los más demandados por nuestros clientes. En esta tabla se refleja la foto del producto, el nombre o modelo de este, el precio por unidad, las unidades vendidas en el año actual, y el dinero total ingresado.

Más Vendidos | Este Año

10

Foto	Nombre	Precio	Unid.	Total
	A-4 Abrillantador Suelos Pizarra	59.0 €	50	2950.0 €
	X-3 Cristalizador Rosa	32.0 €	29	928.0 €
	D-2 Decapante Alcalino	28.0 €	25	700.0 €
	SUPERLUX A 330 S	247.0 €	15	3705.0 €
	SUPERLUX A 28 S	130.0 €	11	1430.0 €

Mostrando 1 de 5 de las 5 entradas

Ilustración 82 - Sección de más vendidos

En la parte derecha del Dashboard nos encontramos con otras tres subsecciones claramente diferenciadas, la primera subsección es un controlador de eventos, este controlador registra todas las acciones efectuadas en la aplicación y muestra hasta un total de quince en orden descendente por tiempo, es decir, se muestran los cambios más recientes primero. En la tabla queda constancia el tiempo que ha pasado desde dicho cambio, la sección a la que pertenece el cambio, y una breve descripción indicando el cambio.

Actividad Reciente | Hoy

0 min	● [Inventario] ¡Nuevo Vehículo Registrado!
1 min	● [Agenda] ¡Nuevo Proveedor Registrado!
7 min	● [Facturación] ¡Nueva Factura Registrada!
7 min	● [Facturación] ¡Nuevo Albarán Registrado!
9 min	● [Tareas] ¡Nueva Tarea Registrada!
5 hrs	● [Facturación] ¡Nueva Factura Registrada!
5 hrs	● [Facturación] ¡Nuevo Albarán Registrado!
5 hrs	● [Tareas] ¡Nueva Tarea Registrada!
5 hrs	● [Facturación] ¡Nueva Factura Registrada!
5 hrs	● [Facturación] ¡Nueva Factura Registrada!
5 hrs	● [Facturación] ¡Nueva Factura Registrada!
5 hrs	● [Facturación] ¡Nueva Factura Registrada!
5 hrs	● [Facturación] ¡Nuevo Albarán Registrado!
5 hrs	● [Tareas] ¡Nueva Tarea Registrada!
5 hrs	● [Facturación] ¡Nuevo Albarán Registrado!
5 hrs	● [Tareas] ¡Nueva Tarea Registrada!

Ilustración 83 - Sección de Actividad Reciente

A continuación, se ha añadido un pequeño calendario, la funcionalidad de este es muy simple, ya que su única utilidad es indicar el día de hoy, o visualizar el día de una fecha concreta, eso sí, nos permite movernos por los meses mediante flechas e incluso se le ha añadido un botón de mostrar el día actual.

Para su elaboración se ha generado un JS muy simple con el cuál mostramos tantos días del mes anterior hasta completar la primera semana del mes, los días del mes indicado hasta completar el mes en su día y semana respectiva y finalmente, y de igual manera que al principio tantos días el mes posterior hasta completar la última semana del mes.

Además del JS, se ha generado un css nuevo imitando el diseño de la plantilla Bootstrap para algunos retoques como el círculo que selecciona el día actual. Para la botonera o los textos se ha utilizado el css previamente diseñado.

Calendario

Febrero 2023 📅 ⏪ ⏩

Lun	Mar	Mie	Jue	Vie	Sab	Dom
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	1	2	3	4	5

Ilustración 84 - Calendario del Dashboard

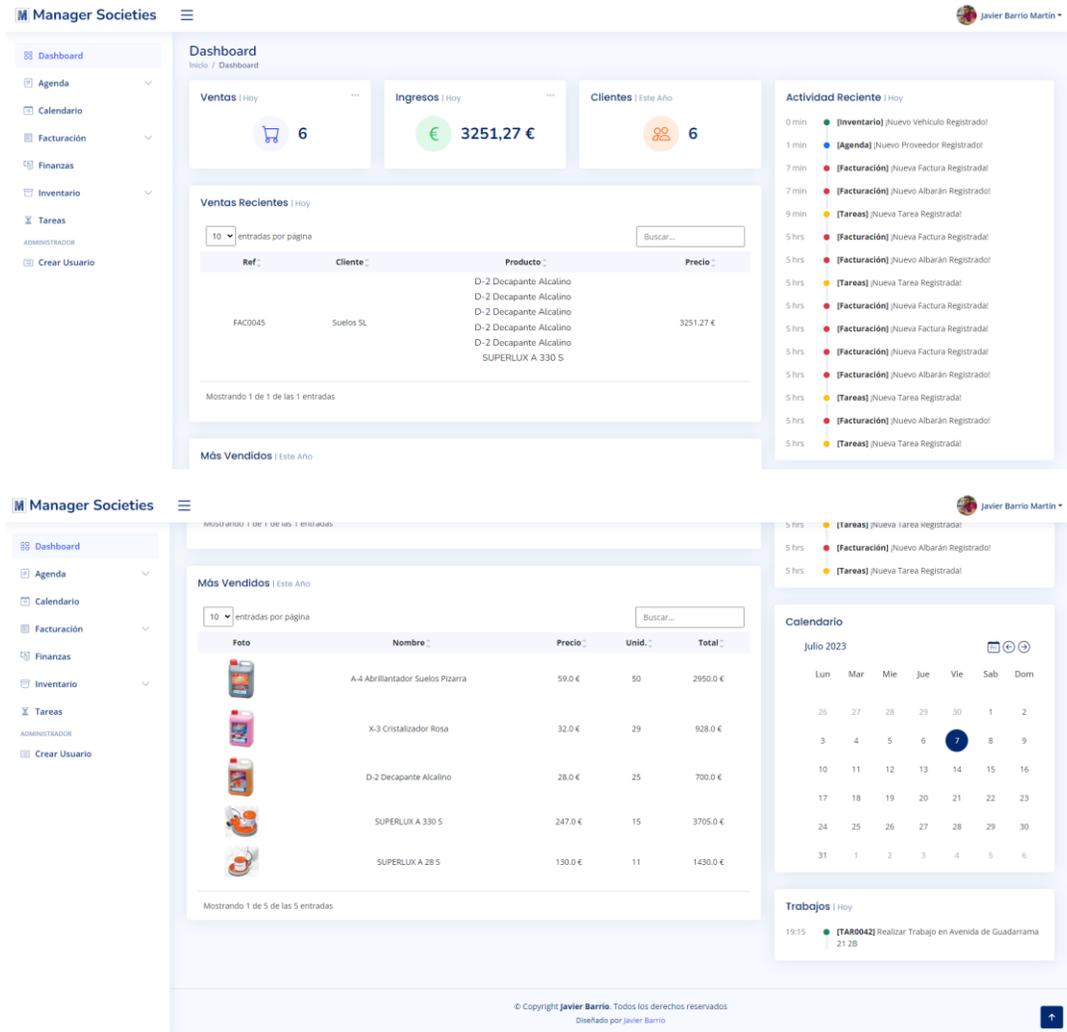
Por último, encontramos una pequeña subsección que indica si existe alguna tarea en el día actual, para tenerlas controladas. Se muestra junto a la hora de la tarea, la referencia entre paréntesis y una pequeña descripción.

Trabajos | Hoy

19:15 ● [TAR0042] Realizar Trabajo en Avenida de Guadarrama
21 2B

Ilustración 85 - Subsección de Trabajos

La sección del Dashboard no tiene una entidad como tal propia, la única entidad relacionada es el controlador de eventos o actividades, a su vez, posee su repositorio y su servicio, al ser una sección de consulta se basa en consultas a la base de datos para mostrar los datos indicados.



The screenshot displays the 'Manager Societies' dashboard. The top navigation bar includes 'Dashboard', 'Agenda', 'Calendario', 'Facturación', 'Finanzas', 'Inventario', 'Tareas', and 'Crear Usuario'. The main content area is divided into several sections:

- Dashboard Summary:** Shows 'Ventas | Hoy' (6), 'Ingresos | Hoy' (€ 3251,27 €), and 'Clientes | Este Año' (6).
- Ventas Recientes | Hoy:** A table listing recent sales with columns for Ref., Cliente, Producto, and Precio. The first entry is for 'FAC0045' from 'Suelos SL' with a price of 3251,27 €.
- Más Vendidos | Este Año:** A table listing top-selling products with columns for Foto, Nombre, Precio, Unid., and Total. The top product is 'A-4 Abrillantador Suelos Pizarra' with a total of 2950,0 €.
- Actividad Reciente | Hoy:** A list of recent activities such as 'Inventario', 'Agenda', and 'Facturación' with timestamps.
- Calendario:** A calendar view for July 2023.
- Trabajos | Hoy:** A list of tasks, including 'Realizar Trabajo en Avenida de Guadarrama'.

At the bottom of the dashboard, there is a copyright notice: '© Copyright Javier Barrio. Todos los derechos reservados. Diseñado por Javier Barrio'.

Ilustración 86 - Sección del Dashboard

5.2.13. Facturación:

El módulo de facturación es el más importante de la aplicación puesto que es el corazón de los registros de los ingresos económicos de esta. Posee cuatro subsecciones, albaranes, facturas, gastos externos y nóminas.

Es una sección que está relacionada con cada una de las otras secciones de la aplicación por lo que la lógica en código es elevada. En cuanto a la subsección de albaranes, nos encontramos con el diseño que se ha ido manteniendo a lo largo del proyecto. En la parte superior cuenta con un contador que indica el número total de albaranes.

A continuación, nos encontramos con una tabla en la que podemos visualizar la referencia del albarán, el cliente tratado, la fecha del albarán, el precio total del mismo y cuatro botones, cada uno con una

funcionalidad concreta. El primero de los botones es dinámico, visiblemente es el icono de un dólar, este botón cumple la funcionalidad de facturar un albarán, que no es más que crear la factura asociada a dicho albarán, en este momento el botón del dólar es sustituido por un cuadrado con un check, indicando que dicho albarán ya está facturado e impidiendo volver a ser facturado.

A nivel lógico para la aplicación en este momento ese albarán se relaciona con una factura y se cataloga como un ingreso en la misma. A la derecha del botón de facturar encontramos el botón de desglose, este al ser pulsado muestra un modal con toda la información detallada del albarán desglosada.

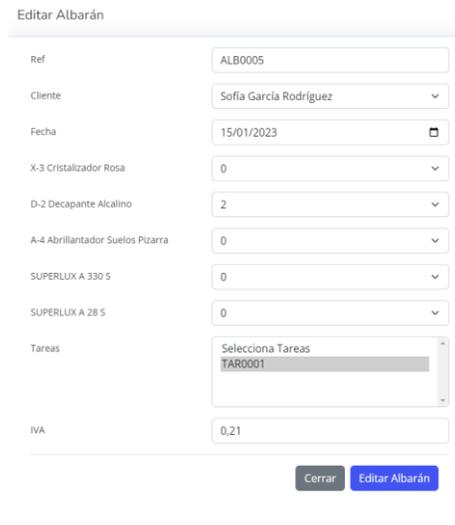


Desglose Albarán	
Ref	CLI0001
Nombre	Sofia García Rodríguez
DNI	53852030Z
Fecha	15/01/2023
Productos	
D-2 Decapante Alcalino	28,00 €
D-2 Decapante Alcalino	28,00 €
Tareas	
TAR0001	3000,00 €
Precio Sin IVA	3056,00 €
IVA	0,21 %
Precio con IVA	3697,75 €

Cerrar

Ilustración 87 - Desglose de un Albarán

Los dos botones restantes son los que hemos estado viendo a lo largo de la aplicación, uno que permite la edición del albarán y otro para eliminarlo.



Editar Albarán	
Ref	ALB0005
Cliente	Sofia García Rodríguez
Fecha	15/01/2023
X-3 Cristallizador Rosa	0
D-2 Decapante Alcalino	2
A-4 Abrillantador Suelos Pizarra	0
SUPERLUX A 330 S	0
SUPERLUX A 28 S	0
Tareas	Selecciona Tareas TAR0001
IVA	0,21

Cerrar Editar Albarán

Ilustración 88 - Edición del Albarán

Podemos además crear un albarán, en este modal podemos seleccionar uno de los clientes de la aplicación y la fecha de este. A su vez, se muestran todos los productos registrados en la aplicación con un seleccionable para la cantidad del 0 al stock almacenado en el inventario. Del mismo modo, aparecerán las tareas que no están cobradas en la aplicación. Para finalizar, aparece una celda para indicar el IVA, por defecto, aparecerá un IVA del veintiún por ciento.

Crear Albarán
✕

Cliente Selecciona Cliente ▾

Fecha dd/mm/aaaa 📅

X-3 Cristallizador Rosa Selecciona Cantidad ▾

D-2 Decapante Alcalino Selecciona Cantidad ▾

A-4 Abrillantador Suelos Pizarra Selecciona Cantidad ▾

SUPERLUX A 330 S Selecciona Cantidad ▾

SUPERLUX A 28 S Selecciona Cantidad ▾

Tareas Selecciona Tareas ▾

IVA 0,21

Cerrar
Crear Albarán

Ilustración 89 - Crear Albarán

Manager Societies
Javier Barrio Martín

Albaranes
Dashboard / Facturación / Albaranes

14

Listado de Albaranes

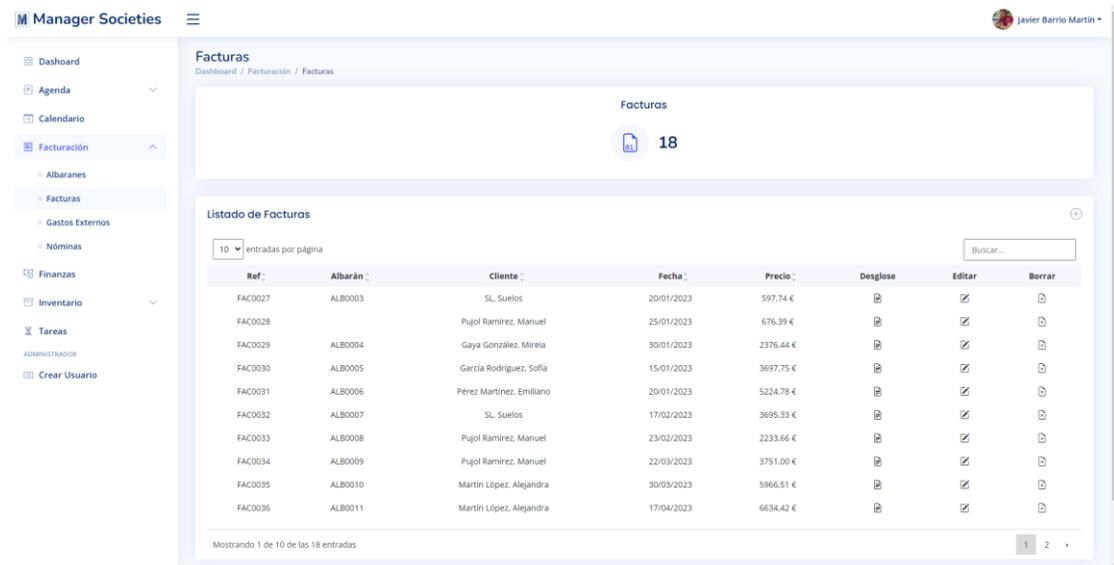
10 entradas por página Buscar...

Ref.	Cliente	Fecha	Precio	Facturar	Desglose	Editar	Borrar
ALB0003	SL Suelos	20/01/2023	597.74 €	☑	📄	✎	🗑
ALB0004	Gaya González, Mireia	30/01/2023	2376.44 €	☑	📄	✎	🗑
ALB0005	García Rodríguez, Sofia	15/01/2023	3697.75 €	☑	📄	✎	🗑
ALB0006	Pérez Martínez, Emiliano	20/01/2023	5224.78 €	☑	📄	✎	🗑
ALB0007	SL Suelos	17/02/2023	3695.33 €	☑	📄	✎	🗑
ALB0008	Pujol Ramirez, Manuel	23/02/2023	2233.66 €	☑	📄	✎	🗑
ALB0009	Pujol Ramirez, Manuel	22/03/2023	3751.00 €	☑	📄	✎	🗑
ALB0010	Martin López, Alejandra	30/03/2023	5966.51 €	☑	📄	✎	🗑
ALB0013	Pérez Martínez, Emiliano	07/02/2023	883.30 €	☑	📄	✎	🗑
ALB0014	García Rodríguez, Sofia	20/02/2023	629.19 €	☑	📄	✎	🗑

Mostrando 1 de 10 de las 14 entradas 1 2

Ilustración 90 - Subsección de Albaranes

La subsección de facturas es un clon de la subsección de albaranes con dos pequeñas particularidades, por razones obvias no posee el botón de facturar, ya que, una factura siempre estará facturada. Además, añade una columna que puede estar vacía o no, indicando la referencia del albarán asociado a la factura, en el caso de estar vacía significa que es una factura final sin albarán creado previamente.



Ref.	Albarán	Cliente	Fecha	Precio	Desglose	Editar	Borrar
FAC0027	ALB0003	SL Suelos	20/01/2023	597.74 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
FAC0028		Pujol Ramírez, Manuel	25/01/2023	676.39 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
FAC0029	ALB0004	Gaya González, Nireia	30/01/2023	2376.44 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
FAC0030	ALB0005	García Rodríguez, Sofía	15/01/2023	3697.75 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
FAC0031	ALB0006	Pérez Martínez, Emiliano	20/01/2023	5224.79 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
FAC0032	ALB0007	SL Suelos	17/02/2023	3695.33 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
FAC0033	ALB0008	Pujol Ramírez, Manuel	23/02/2023	2233.66 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
FAC0034	ALB0009	Pujol Ramírez, Manuel	22/03/2023	3751.00 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
FAC0035	ALB0010	Martín López, Alejandra	30/03/2023	5966.51 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
FAC0036	ALB0011	Martín López, Alejandra	17/04/2023	6634.42 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Ilustración 91 - Subsección de Facturas

La siguiente subsección es la de gastos externos, estos gastos externos están presentes en la sección de tareas, es decir, si al crear una tarea se indica un gasto extra, se generará automáticamente en esta subsección. Estos gastos tendrán como motivo la referencia de la tarea, en otros casos el motivo será elegido por el creador del gasto. Además, se mostrará la referencia del gasto, el empleado al que se asocia el gasto extraordinario, la fecha y la cantidad del gasto. Por otro lado, se visualizan los botones de editado, borrado y creación.

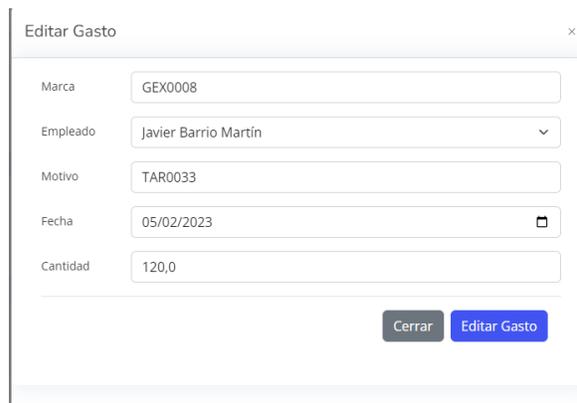


Ilustración 92 - Edición de Gasto

Cerrar x

Empleado

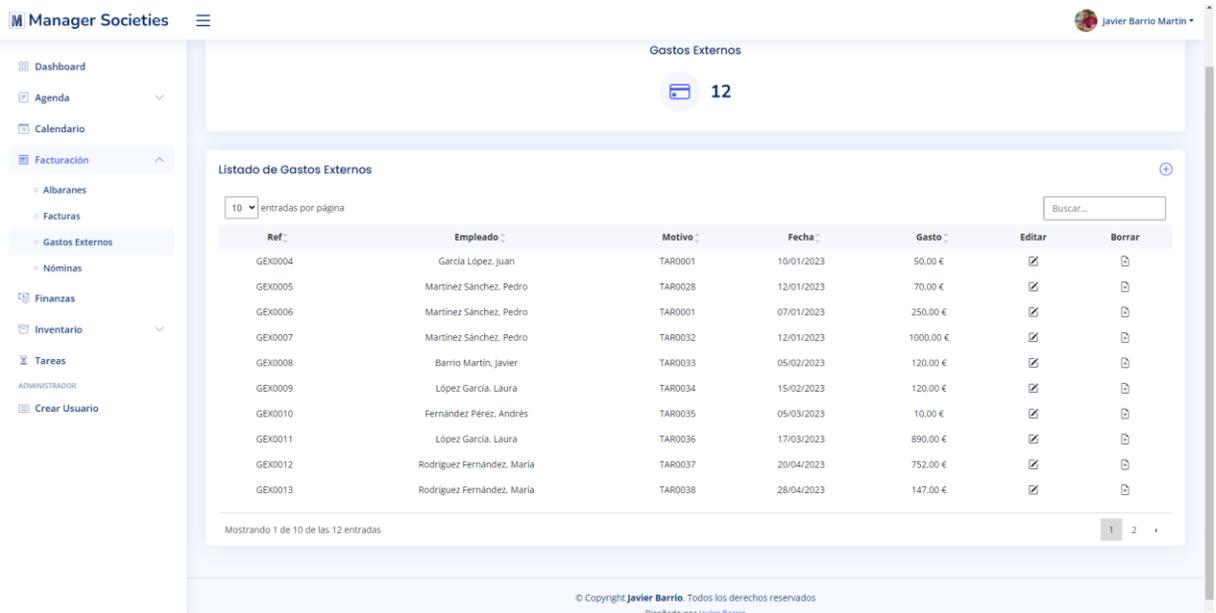
Motivo

Fecha

Gasto

Cerrar
Crear Gasto

Ilustración 93 - Creación de Gasto



Gastos Externos 12

Listado de Gastos Externos

10 entradas por página Buscar...

Ref.	Empleado	Motivo	Fecha	Gasto	Editar	Borrar
GEX004	García López, Juan	TAR0001	10/01/2023	50,00 €	✓	🗑️
GEX005	Martínez Sánchez, Pedro	TAR0028	12/01/2023	70,00 €	✓	🗑️
GEX006	Martínez Sánchez, Pedro	TAR0001	07/01/2023	250,00 €	✓	🗑️
GEX007	Martínez Sánchez, Pedro	TAR0032	12/01/2023	1000,00 €	✓	🗑️
GEX008	Barrio Martín, Javier	TAR0033	05/02/2023	120,00 €	✓	🗑️
GEX009	López García, Laura	TAR0034	15/02/2023	120,00 €	✓	🗑️
GEX0010	Fernández Pérez, Andrés	TAR0035	05/03/2023	10,00 €	✓	🗑️
GEX0011	López García, Laura	TAR0036	17/03/2023	890,00 €	✓	🗑️
GEX0012	Rodríguez Fernández, María	TAR0037	20/04/2023	752,00 €	✓	🗑️
GEX0013	Rodríguez Fernández, María	TAR0038	28/04/2023	147,00 €	✓	🗑️

Mostrando 1 de 10 de las 12 entradas 1 2

© Copyright Javier Barrio. Todos los derechos reservados
Diseñado por Javier Barrio

Ilustración 94 - Subsección de Gastos

La siguiente subsección es junta a la de facturas la más importante, si la subsección de facturas hace referencia al grueso de los ingresos de la aplicación, la subsección de nóminas hace referencia al grueso de los gastos fijos que posee la empresa, que no es más que las nóminas de sus empleados.

Como en el resto de las subsecciones encontramos un contador con el número de nóminas almacenadas en la aplicación, para a continuación, mostrar en una tabla, la referencia de la nómina, el empleado relacionado, la fecha de la nómina, el sueldo bruto y neto de esta, y tres botones, uno de desglose que muestra la información detallada, el de edición y el de borrado. Además, el de creación de nóminas en la parte superior derecha.

El desglose muestra todos los campos registrados en la nómina, la información completa del empleado, la fecha, los días trabajados en el mes y el sueldo bruto, este se desglosa en el salario por convenio, la prestación por accidente, el complemento salarial, los ingresos asociados al teletrabajo, el complemento de productividad y las pagas extras, además, se muestran las deducciones de contingencias, los gastos en formación, por desempleo, el % de IRPF aplicado y su retención, para mostrar al final el sueldo neto total.

Ref	EMP0001
Nombre	Juan García López
DNI	47564785A
Fecha	01/01/2023
Días Trabajados	30
Sueldo Bruto	1950,00 €
Salario Convenio	900,00 €
Prestación Accidente	450,00 €
Complemento Salarial	300,00 €
Teletrabajo	0,00 €
Comp. Productividad	300,00 €
Paga Extra / mes	0,00 €
Contingencias	- 135,00 €
Gasto Formación	- 0,00 €
Desempleo	- 135,00 €
% IRPF	20.16 %
Retención IRPF	- 272,00 €
Sueldo Neto	1408,00 €

Cerrar

Ilustración 95 - Desglose de Nómina

El desglose muestra todos los campos registrados en la nómina, la información completa del empleado, la fecha, los días trabajados en el mes y el sueldo bruto, este se desglosa en el salario por convenio, la prestación por accidente, el complemento salarial, los ingresos asociados al teletrabajo, el complemento de productividad y las pagas extras, además, se muestran las deducciones de contingencias, los gastos en formación, por desempleo, el % de IRPF aplicado y su retención, para mostrar al final el sueldo neto total.

A la hora de crear una nómina se nos solicitan todos los campos mencionados anteriormente, con la diferencia de que se solicitan los ingresos tanto como los devengos con el precio por día, este se multiplica por los días trabajados y se registra en la base de datos.

Cerrar Crear Nómina

Empleado

Fecha Creación

Días Trabajados

Salario Convenio / día

Prestación Accidente / día

Complemento Salarial/ día

Teletrabajo / día

Comp. Productividad / día

Paga Extra / paga

Contingencias / día

Gasto Formación / día

Desempleo / día

Ilustración 96 - Creación de Nómina

Manager Societies
Javier Barrio Martín

Nóminas
Dashboard / Facturación / Nóminas

Nóminas 23

Listado de Nóminas

10 entradas por página Buscar...

Ref.	Empleado	Fecha	Bruto	Neto	Desglose	Editar	Borrar
NOM0001	García López, Juan	01/01/2023	1950,00 €	1408,00 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
NOM0002	Rodríguez Fernández, María	01/01/2023	1800,00 €	1053,00 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
NOM0003	Martínez Sánchez, Pedro	02/01/2023	2160,00 €	1125,00 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
NOM0004	López García, Laura	02/01/2023	2040,00 €	1236,00 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
NOM0005	Barrio Martín, Javier	01/01/2023	3719,10 €	2473,86 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
NOM0006	Fernández Pérez, Andrés	01/01/2023	2040,00 €	1071,00 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
NOM0007	García López, Juan	01/02/2023	1950,00 €	1128,00 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
NOM0008	Rodríguez Fernández, María	01/02/2023	1800,00 €	1053,00 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
NOM0009	Rodríguez Fernández, María	01/02/2023	2160,00 €	1125,00 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>
NOM0010	López García, Laura	01/02/2023	2040,00 €	1071,00 €		<input checked="" type="checkbox"/>	<input type="checkbox"/>

Mostrando 1 de 10 de las 23 entradas 1 2 3 >

Ilustración 97 - Subsección de Nóminas

5.2.14. Finanzas:

La última sección de la aplicación es la de finanzas, al igual que el dashboard, es una sección de consultas por lo que su funcionamiento se basa en peticiones de información a base de datos que son gestionadas por el controlador de la aplicación y el propio servicio de finanzas.



Ilustración 98 - Ingresos, Gastos y Balance

Al acceder nos encontramos con los ingresos, gastos y balance total del año, permitiéndonos en todo momento conocer la salud económica de la empresa. A continuación, se nos muestra un gráfico en el que se nos permite delimitar los objetivos de ingresos mensuales, para a su vez, mostrar la situación de ingresos mes a mes agrupada por cuatrimestre.

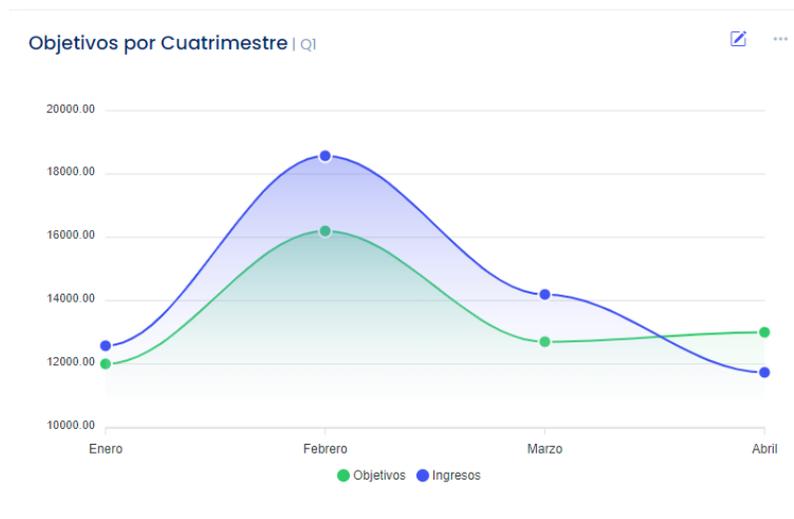


Ilustración 99 - Objetivos por Cuatrimestre

En la parte derecha tenemos un gráfico de evolución de los beneficios en el que se indican los ingresos fijos y variables, así como los gastos fijos y variables y el balance total de estos.

Evolución de Beneficios

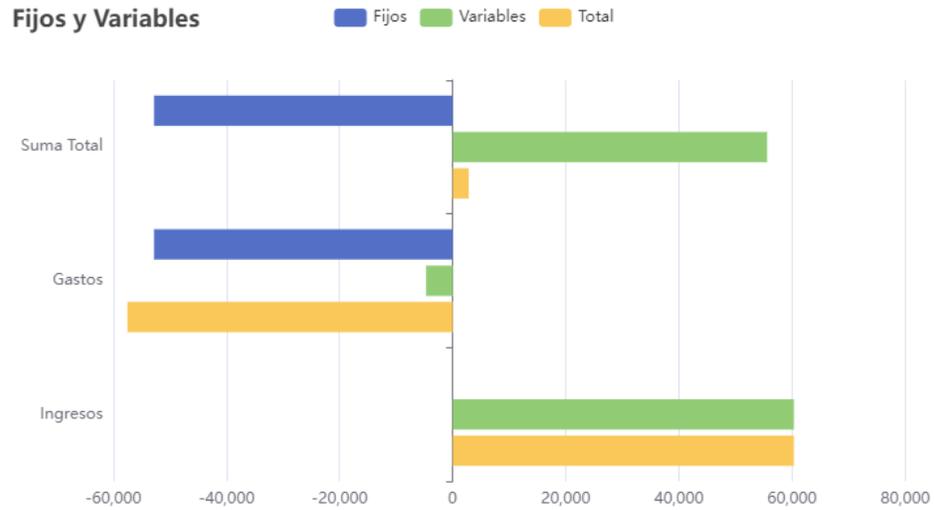


Ilustración 100 - Evolución de los Beneficios

Por último, se muestra un gráfico de sectores en el que se indica el % de ingresos por cliente de los mejores cuatro, indicando como otros el cómputo global del resto de clientes de la aplicación.

Mejores Clientes

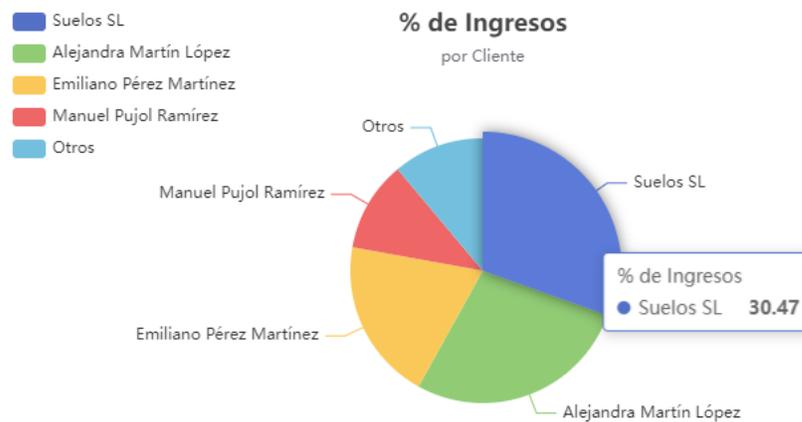


Ilustración 101 - Mejores Clientes

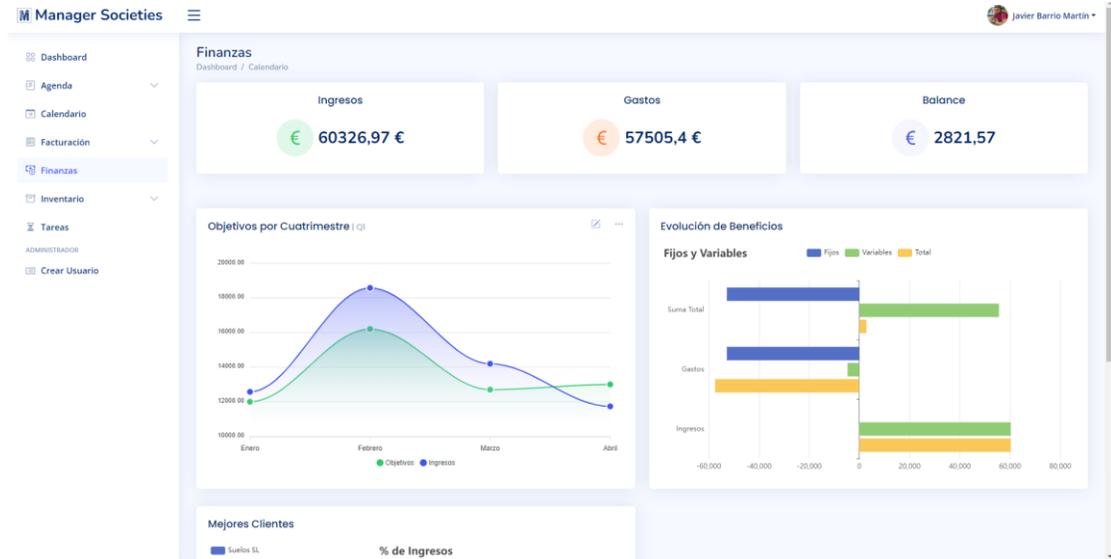


Ilustración 102 - Sección de Finanzas

6. Conclusiones y Trabajos Futuros:

En el presente trabajo se propuso un modelo y metodología para el desarrollo de una aplicación web desde cero, considerando los procesos clave del desarrollo, partiendo de un trabajo de investigación sobre empresas que podrían necesitar una aplicación de este tipo y recogiendo necesidades que debían ser resueltas en este proyecto.

Como continuación de este trabajo de tesis y como en cualquier proyecto de investigación, han quedado diversas líneas abiertas y en las que es posible continuar trabajando. Algunas de estas líneas abiertas se esperan atacar en un futuro cercano.

Por ejemplo, desligar el diseño de la aplicación al uso de Bootstrap y generar así un diseño aún más propio y personal, permitiendo reducir los archivos CSS, ya que, gran parte de lo contenido en estos archivos no se ha utilizado.

Por otro lado, realizar una refactorización sobre los templates HTML, añadiendo alguna funcionalidad pensada en un inicio y que no se ha podido llevar a cabo en la versión final.

Además, realizar una refactorización del código de la parte backend buscando optimizarlo y evitar posibles errores, para ello, se podrían utilizar herramientas de apoyo como SonarQube. Incluso, rediseñar el esquema de la base de datos buscando uno óptimo y eficiente con menos dependencias.

Incluso el diseño de tests unitarios que comprueben el buen funcionamiento de la aplicación, test de integración, *smoke tests*, y por supuesto, test *end to end*, mediante Cypress o Selenium, que nos permitan comprobar el estado y calidad de la aplicación de forma automática.

En último lugar, el siguiente paso más ambicioso es conseguir una versión del proyecto que pueda entrar en el mercado laboral, dicho de una forma más casual, sacar la aplicación al mercado.

7. Bibliografía:

- [1] Infoautónomos, «Los Problemas de los Autónomos,» [En línea]. Available: <https://www.infoautonomos.com/autonomos-espana-ley/la-problematika-de-los-autonomos/>. [Último acceso: 24 02 2021].
- [2] European Knowledge Cente, «¿Qué es un sistema ERP y para qué sirve?,» [En línea]. Available: <https://www.ticportal.es/temas/enterprise-resource-planning/que-es-sistema-erp>. [Último acceso: 2021].
- [3] R. K. a. Z. Khan, «A Research Study on the ERP System Implementation and Current Trends in ERP,» *Shanlax International Journal of Management*, p. 6, 10 2020.
- [4] European Knowledge Center, «ERP en la nube,» [En línea]. Available: <https://www.ticportal.es/temas/enterprise-resource-planning/erp-nube>. [Último acceso: 17 02 2021].
- [5] Abeerden Group, «ERP in Manufacturing,» [En línea]. Available: <https://www3.technologyevaluation.com/es/research/white-paper/erp-in-manufacturing-2012-the-evolving-erp-strategy.html>. [Último acceso: 17 02 2021].
- [6] I. autónomos, «Info Autónomos,» 2006. [En línea]. Available: <https://www.infoautonomos.com/autonomos-espana-ley/evolucion-y-caracteristicas-de-los-autonomos/>. [Último acceso: 2022].
- [7] E. País, «El País,» 2020. [En línea]. Available: https://cincodias.elpais.com/cincodias/2020/02/15/autonomos/1581757952_736650.html. [Último acceso: 2022].
- [8] Chara, «Two Worlds,» 2017. [En línea]. Available: <https://dosmundos.com/2017/03/24/generaciones-y-la-tecnologia-generacion-x/>. [Último acceso: 2022].
- [9] TIC.Portal, «TIC.portal,» 30 junio 2022. [En línea]. Available: <https://www.ticportal.es/temas/enterprise-resource-planning/coste-erp#:~:text=En%20el%2070%25%20de%20los,de%20mayor%20tama%C3%B1o%20y%20coste..>
- [10] R. Schwarz, «The Power World Of Mouth Has in Marketing,» 2020. [En línea]. Available: <https://www.forbes.com/sites/forbescommunicationscouncil/2020/09/08/the-power-word-of-mouth-has-in-marketing-and-how-to-cultivate-it/?sh=76781b143df0>.
- [11] T. d. Información, «Los 10 mejores ERP,» [En línea]. Available: <https://www.tecnologias-informacion.com/erp/mejores-erp.html>. [Último acceso: 2022].
- [12] SAP, «Financial Management,» 2023. [En línea]. Available: sap.com/spain/products/financial-management.html.
- [13] B. Avantis, «Avantis,» 2019. [En línea]. Available: <https://blog.avantis.mx/cuanto-cuesta-sap-business-one-implementacion>. [Último acceso: 2022].
- [14] Cronomia, «Cronomia ERP Oracle,» 2022. [En línea]. Available: <https://www.cronomia.com/software/oracle#:~:text=Oracle%20ERP%20Cloud%20se%20presenta,%3A%2080%E2%82%AC%2Fusuario%2Fmes>. [Último acceso: 2022].
- [15] Intuit, «Quickbooks,» 2022. [En línea]. Available: <https://quickbooks.intuit.com/>. [Último acceso: 2022].
- [16] Kanbanize, «Qué es Kanban,» 2023. [En línea]. Available: [https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban#:~:text=La%20palabra%20japonesa%20%22kanban%22%2C,%22justo%20a%20tiempo%22\)..](https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban#:~:text=La%20palabra%20japonesa%20%22kanban%22%2C,%22justo%20a%20tiempo%22)..)
- [17] Asana, «Qué es Scrum y cómo aplicarlo en gestión de proyectos,» 17 agosto 2022. [En línea]. Available: <https://asana.com/es/resources/what-is-scrum>.

- [18] Atlassian, «Trello,» 2023. [En línea]. Available: <https://trello.com/es>.
- [19] Anónimo, «GitHub,» [En línea]. [Último acceso: 2022].
- [20] Anónimo, «Git,» [En línea]. Available: <https://es.wikipedia.org/wiki/Git>. [Último acceso: 2022].
- [21] Lucidchart, «Qué es el lenguaje unificado de modelado (UML),» [En línea]. Available: <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>. [Último acceso: 2022].
- [22] U. d. Alicante, «Modelo Vista Controlador,» [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>.
- [23] Anónimo, «HTML,» [En línea]. Available: <https://es.wikipedia.org/wiki/HTML>.
- [24] Anónimo, «CSS,» [En línea]. Available: <https://es.wikipedia.org/wiki/CSS>.
- [25] Anónimo, «Bootstrap Framework,» [En línea]. Available: [https://es.wikipedia.org/wiki/Bootstrap_\(framework\)](https://es.wikipedia.org/wiki/Bootstrap_(framework)).
- [26] Anónimo, «JavaScript,» 2023. [En línea]. Available: <https://es.wikipedia.org/wiki/JavaScript>.
- [27] Anónimo, «jQuery,» [En línea]. Available: <https://es.wikipedia.org/wiki/JQuery>.
- [28] Anónimo, «Maven,» [En línea]. Available: <https://es.wikipedia.org/wiki/Maven>.
- [29] Anónimo, «Lenguaje de Programación Java,» [En línea]. Available: [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)).
- [30] CodMind, «¿Qué es Spring Boot?,» [En línea]. Available: <https://blog.codmind.com/que-es-spring-boot/>.
- [31] Anónimo, «Thymeleaf,» [En línea]. Available: <https://es.wikipedia.org/wiki/Thymeleaf>.
- [32] B. Made, «Bootstrap Made - Nice Admin,» 21 12 2022. [En línea]. Available: <https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template/>.
- [33] Thymeleaf, «Thymeleaf,» [En línea]. Available: <https://www.thymeleaf.org/>.

8. Anexos:

- Imágenes mostradas a lo largo del documento.
- Diagrama de Componentes Original.
- Repositorio de Git-Hub: <https://github.com/JaviBarrio6/ManagerSocieties>.
- Presentación en PPT.