Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

Grado en Ingeniería en Organización Industrial

Personalización de módulos en OpenERP 7.0: aplicación a la gestión de la flota de vehículos de una PYME

Autor:

Fernández Alonso, Yeray

Tutor:

Sanz Angulo, Pedro

Departamento de Organización de Empresas y Comercialización e Investigación de Mercados

Valladolid, julio 2014.

AGRADECIMIENTOS

Quería agradecer en estas líneas a las personas que me han ayudado en la realización del trabajo. En primer lugar a mi tutor durante este proyecto, Pedro Sanz Angulo, por todos los consejos y ayuda que me ha brindado durante estos meses. Y en segundo lugar a Rebeca, por todo su apoyo y cariño desde siempre.

RESUMEN

En el presente documento se analiza el software ERP, su importancia y su historia, y se presentan los distintos tipos que existen, así como las marcas más importantes a fin de seleccionar la herramienta más adecuada a los objetivos del TFG. La selección final es OpenERP, de la que se explica, a nivel general y funcional, su configuración, analizando la composición de sus módulos.

También se estudian los mecanismos existentes para modificar dichos módulos y, tras compararlas, se selecciona la más apropiada. A continuación se explica cómo utilizar el método elegido y se procede a personalizar un módulo destinado a la gestión de la flota de vehículos a modo de ejemplo. Para finalizar, se elabora un análisis económico del proyecto y se extraen las principales conclusiones de trabajo realizado, así como posibles líneas de actuación futuras.

PALABRAS CLAVE

OpenERP – Personalización – Módulo – Flota de vehículos – PYME

ÍNDICE

INTRODUCCIÓN

Con el fin de estudiar la materia objeto del proyecto es necesario, previamente, analizar en qué contexto se enmarca el mismo. De la misma manera, habrá que realizar una investigación acerca de las herramientas disponibles para el estudio del problema que se presenta y que será preciso resolver. Para ello, se realizará una exposición del procedimiento seguido para la resolución de dicho problema inicial.

CONTEXTO

Hoy en día la forma de realizar muchos procesos tradicionales está siendo evaluada y sometida a numerosos cambios. Dichas alteraciones suceden cada vez más deprisa y, por lo tanto, es necesario que las empresas se adapten a la misma velocidad.

Las consecuencias de no realizar dicha adaptación son grandes, ya que las compañías pueden ser expulsadas del mercado. La competitividad es cada vez mayor y la innovación es un factor clave para mantenerse en una posición ventajosa. Debido a esto, las empresas deben prestar atención a todos los nuevos recursos tecnológicos para incorporarlos cuanto antes.

Un factor determinante para la correcta evolución de las sociedades es el manejo eficaz y preciso de toda la información relevante. En ese sentido, se está desarrollando uno de esos cambios tecnológicos de los que el mundo de los negocios tiene que valerse para mejorar sus sistemas y progresar en el mercado.

El cambio consiste en la introducción de nuevas herramientas capaces de controlar y centralizar toda la información pertinente a la empresa, los ERP (*Enterprise Resource Planning*).

Estos sistemas integran la información de las diferentes áreas de la empresa (recursos humanos, logística, contabilidad, etc.), lo que facilita el flujo de información y permite acceder y modificar los datos de manera conjunta y global. Para que la herramienta sea realmente útil es preciso que se ajuste a las características individuales de cada empresa donde se instale.

Puede parecer que solamente las grandes compañías necesitan un ERP porque tienen la necesidad de manejar cantidades enormes de datos cada día. Pero la realidad es diferente, ya que las empresas de menor tamaño también necesitan de los beneficios de la implantación de estos sistemas. Además, las empresas medianas y pequeñas representan la gran mayoría del mercado y son verdaderamente importantes en la economía de las regiones.

Por todo ello, se establecen una serie de objetivos para el presente trabajo.

OBJETIVOS

El principal objetivo de este proyecto consiste en establecer las pautas y los mecanismos necesarios para efectuar la modificación de los módulos de un ERP, de modo que cualquier empresa o persona pueda utilizarlas de guía para personalizar su ERP y adaptarlo, así, a sus necesidades concretas sin tener que recurrir a consultores externos.

Para lograr este objetivo, se realizará un estudio y una comparativa de los distintos ERP existentes y se escogerá el más adecuado a nuestro objeto del trabajo. Una vez seleccionada la herramienta de trabajo, será preciso analizarlo de forma general, describiendo su funcionamiento básico y las posibilidades que ofrece. También, se evaluará en qué medida se adapta a las empresas a las que va dirigido el resultado final del proyecto.

Posteriormente, se estudiarán las funcionalidades que permitan realizar los cambios deseados en el programa para completar los módulos originales y básicos del sistema, ya que se busca personalizar la herramienta y adaptarla a cada una de las empresas y su forma de negocio.

Por último, serán preciso comprobar el correcto funcionamiento de los cambios implementados y que complementen verdaderamente a las funcionalidades que ya ofrecen los módulos originales del ERP, que son la base de la que se parte. Para ello, se trabajará en la personalización de un módulo orientado a la gestión de la flota de vehículos. El resultado final no será más que una versión genérica, pero que servirá de guía para que cada entidad desarrolle la suya.

ESTRUCTURA DEL TFG

A modo de resumen, a continuación se detalla el contenido de cada uno de los capítulos que componen este trabajo.

En el *primer capítulo* se realiza un breve estudio del contexto empresarial actual, donde destaca la necesidad de una adecuada y rápida gestión de la información. Como solución más eficaz a este problema surgieron los sistemas ERP y, por ello, se analizan sus características, historia, ventajas e inconvenientes. Además, se explican los tipos de ERP existentes hoy en día y se proponen unos criterios de elección entre las alternativas comerciales actuales. La mejor opción será un ERP de software libre, concretamente OpenERP.

Durante el *segundo capítulo* se explican las características generales de OpenERP para entender y manejar el programa. También se explica su estructura y otros elementos que pudieran servir de utilidad para comenzar a utilizarlo.

A lo largo del *tercer capítulo* se introducen los módulos como conjunto de archivos de dos lenguajes de programación: Python y XML. Se comenta la estructura de dichos archivos y se definen los elementos que los componen, que se pueden modificar. Este capítulo figura para aportar los conocimientos necesarios con el posible objetivo de una modificación del módulo por medio de la alteración de estos archivos de programación.

El *cuarto* es el *capítulo* central del trabajo. En él, se argumenta razonadamente cuál es el método de modificación más adecuado para este proyecto, que es la modificación vía interfaz. Además, se muestran los pasos para hacerlo.

En el *quinto capítulo* se concretan los cambios planteados sobre el módulo de gestión de la flota de vehículos. Se plantean unas propuestas de cambio a modo de ejemplo que puedan servir de utilidad a una PYME. El resultado es precisamente el módulo modificado.

En el *sexto capítulo* se hace un pequeño análisis económico de todo el proyecto finalizado, estimando los costes asociados a su ejecución y ofreciendo un posible precio de venta.

Para finalizar, el *séptimo capítulo* es en el que se exponen las conclusiones finales de todo el trabajo realizado. Además, se comentan nuevas posibles vías de investigación que están abiertas o por abrir sobre este tema. Este apartado está enfocado con miras al futuro, a un posible nuevo trabajo que pueda ser continuación de éste.

CAPÍTULO I

ERP EN LAS EMPRESAS

Las empresas se definen como la unidad principal de producción; son el motor de la economía y claves en el desarrollo de las sociedades. El gran tamaño de algunas compañías y la necesidad de acceder a toda la información necesaria de forma rápida y precisa las dirige hacia la utilización de sistemas de planificación de recursos tipo ERP (*Enterprise Resource Planning*). Las pequeñas sociedades tampoco escapan a esta necesidad, sobre todo si quieren sobrevivir en un entorno cada vez más competitivo y exigente.

En este capítulo se tratará de proporcionar una visión general de las empresas y conocer cómo se organizan. Además, a lo largo de este capítulo se realizará una breve introducción a los ERP, su historia, características y tipos, para acabar justificando la selección del software ERP que se empleará a lo largo del proyecto.

1.1. Las empresas

Una empresa es una institución o agente económico que toma decisiones sobre la utilización de factores de la producción para obtener bienes y servicios demandados por la sociedad. La actividad productiva llevada a cabo por ésta consiste en la transformación de bienes intermedios (materias primas y productos semielaborados) en bienes finales, mediante el empleo de factores productivos (básicamente trabajo y capital).

Para poder desarrollar su actividad, la empresa necesita combinar tecnología específica con unos factores productivos (recursos). Desde el punto de vista jurídico-mercantil, la empresa en una unidad patrimonial con personalidad jurídica independiente que está constituida por bienes, derechos y obligaciones.

La empresa debe coordinar y organizar una serie de actividades para realizar sus funciones de acuerdo con unos objetivos. Para ello requiere de herramientas que permitan su gestión, intentando reducir lo máximo posible el coste y aumentando la calidad. Como resultado ofrecerá un determinado bien o servicio útil para la sociedad.

1.1.1. Objetivos de las empresas

Los objetivos que persiguen las empresas son muy diversos, debido a la gran variedad de ellas. Pero aun así, se pueden destacar una serie de objetivos comunes que tienen todas las empresas independientemente de su tipo,

tamaño, complejidad, actividad, etc. Todas ellas quieren crecer y aumentar su beneficio con el mínimo coste. Para ello necesitan fidelizar a los clientes ofreciendo gran calidad en sus productos y servicios. De esta manera, pasan a formar parte del tejido empresarial de la sociedad.

Para el cumplimiento de esos objetivos la empresa debe intentar aplicar una mejora continua tanto en el proceso como en el producto final. Para lograr dicho fin, hay que considerar el peso de los trabajadores en la estructura empresarial, de manera que todos ellos deben remar en la misma dirección para en funcionamiento íntegro correcto. La meta buscada del día a día es la satisfacción del cliente.

1.1.2. Funciones de las empresas

Las diferentes funciones que realizan las empresas están condicionadas por su actividad, pero al igual que los objetivos, existen muchas comunes a todas o a la mayoría de las empresas, como: generar valor, riqueza y empleo, coordinar su producción, buscar fuentes de financiación, etc.

Según Henri Fayol [1] toda empresa industrial debe presentar estos grupos de funciones:

- *Funciones técnicas:* con las que se realiza la producción de bienes y servicios.
- *Funciones comerciales:* para hacer sus compras y sus ventas.
- *Funciones financieras:* relacionadas con la búsqueda y gerencia de capitales.
- *Funciones de seguridad:* protección de las personas y bienes de la compañía contra robos, incendios, inundaciones, etc.
- *Funciones contables:* relacionadas con los inventarios, registros, balances, costos y estadísticas.
- *Funciones administrativas:* se encargan de coordinar e integrar las otras cinco funciones.

Dentro de las funciones administrativas, que son las que más nos interesan, distingue:

[1] Fayol, H. (1916). *Administration industrielle et générale; prévoyance, organisation, commandement, coordination, controle.* París: H. Dunod et E. Pinat.

- *Planear:* anticipar el futuro y trazar el plan de acción.
- *Organizar:* constituir el organismo material y social de la empresa.
- *Dirigir:* guiar y orientar al personal.
- *Coordinar:* armonizar todos los actos y todos los esfuerzos colectivos.
- *Controlar:* verificar que todo suceda de acuerdo con las reglas establecidas y las órdenes dadas.

Las funciones administrativas, aunque puedan considerarse las fundamentales de la alta dirección, son responsabilidad de todos los niveles de la empresa. Fayol afirma que conforme se asciende en la escala jerárquica de la organización deben predominan las funciones administrativas en detrimento de las funciones técnicas.

Los seis bloques comentados anteriormente se encuentran desarrollados en mayor o menor medida en todas las empresas, grandes o pequeñas. Las funciones requerirán capacidades específicas, por lo que se debe contar con el personal adecuado para cada puesto.

1.1.3. Áreas y organización de empresas

Las empresas tienen la necesidad de establecer una serie de divisiones internas para su correcto funcionamiento. Según aumentan su tamaño y complejidad, se requiere de personal especializado para encargarse de cada una de ellas.

Suelen encontrarse las siguientes áreas fundamentales: dirección, administración, ventas, producción y contabilidad y finanzas. La mayoría de empresas que se dedican a la producción de bienes también cuentan con áreas de mantenimiento y almacén. Es evidente la correspondencia entre las áreas y las funciones de las que se habla en el apartado anterior.

Los departamentos que se fijan internamente dependen de cada una de las empresas, pero las funciones básicas aparecerán en todas ellas. De este modo, mismas funciones en empresas diferentes se realizan en departamentos distintos. También varía de una a otra el tamaño de las divisiones, nombre, número de personas que las integran, etc.

1.1.4. Crecimiento de las empresas

Ante el crecimiento de grandes factorías o marcas, cada vez es más necesario un sistema que sea capaz de coordinar el intenso flujo de información que corre entre los departamentos.

Una de la causas de este crecimiento generalizado es la globalización, que implica una mayor compraventa de bienes y servicios, un aumento de las exportaciones y la importaciones, así como de conocimientos, transmisión de la experiencia, capitales, etc. Se consigue, por tanto, una expansión internacional que, en muchos casos, abre nuevos mercados, crea necesidades y permite el acceso a otros nuevos productos.

Para la continuidad de la cadena productiva, los proveedores se tienen que adaptar a las nuevas condiciones de sus clientes, surgiendo un crecimiento en efecto dominó en unos casos o la búsqueda de nuevos clientes en otros. Aunque lo más normal es que las compañías que no acompañen el desarrollo internacional de las empresas globalizadas acaben desapareciendo.

La competitividad generada abre rutas de búsqueda de nuevos negocios, una inversión en I+D+i mayor, permanente interés por reducir costes y mejorar los procesos, etc. Los que consigan mejores resultados en estos ámbitos se convertirán en los líderes de los mercados internacionales. Si bien es cierto que la mayoría de las empresas compiten en un ámbito local y regional (especialmente en España está muy arraigada la tradición de empresa típicamente local, incluso de carácter familiar), muchas de ellas se lanzan ahora a la exportación de sus productos.

1.2. Los sistemas ERP

Ya se ha comentado anteriormente la vital importancia que tiene la información en las organizaciones hoy en día. La velocidad a la que ésta se genera en la empresa es muy elevada debido a las continuas operaciones que se llevan a cabo, siendo necesario almacenar muchos datos. Teniendo en cuenta los que hay en cada departamento, hace que en toda la empresa haya que gestionar innumerables de ellos.

En ocasiones, sucede que los datos son comunes para varios departamentos, por lo que la información está duplicada. En otras, el dato que se necesita en una sección no se encuentra en la que debería estar. Ambos casos representan pérdidas en la organización, porque o se está ocupando más

espacio de almacenamiento de lo debido o hay que emplear tiempo en encontrar la información, respectivamente.

Los ERP son una herramienta que surge para solucionar problemas como los mencionados. Estas herramientas proporcionan el control y la centralización de la información para asegurar que la toma de decisiones se ejecute basándose en los datos adecuados en el momento preciso. En las empresas es clave tener la información oportuna de cada departamento en el instante que corresponda, ya que es necesaria para que puedan realizar correctamente sus actividades. De esta manera, se crea una base de datos donde se pueda manejar la información en tiempo real y extraer los datos requeridos rápidamente.

1.2.1. Características de los ERP

Los ERP (*Enterprise Resource Planning*) son sistemas de planificación de recursos empresariales que integran y facilitan la gestión de las organizaciones, uniendo todos los departamentos e incluyendo todos los procesos.

Lo más destacable de un ERP es que unifica y ordena toda la información de la empresa en un una única herramienta. De este, modo cualquier movimiento o transformación de la misma queda a la vista de forma inmediata, posibilitando la toma de decisiones de forma más rápida y segura, acortando los ciclos productivos. Con un ERP tendremos la empresa bajo control e incrementaremos la calidad de nuestros servicios y productos.

La implantación de un ERP conlleva la eliminación de barreras ínter-departamentales; la información fluye por toda la empresa eliminando la improvisación por falta de información.

Las características más destacables de un sistema ERP son las siguientes:

- *Integral:* ordena y unifica la información de toda la empresa. Facilita el control de los diferentes procesos de la compañía y los flujos de trabajo. Como todos los departamentos están relacionados entre sí, el ERP establece las conexiones entre ellos. Cada resultado de una actividad en el seno de la empresa es el comienzo de otra. De este modo, se evita la posible aparición de la información duplicada que se presta a errores, porque el flujo de información entre los departamentos no se manipula y se encuentra protegido. La base de datos está centralizada y contiene toda la información.

- *Modular:* las funciones que incluye un sistema ERP vienen divididas en módulos (arquitectura abierta). De esta manera, se podrán instalar unos u otros dependiendo de las necesidades de la empresa, lo que supone una ventaja técnica y un ahorro, ya que las funcionalidades que no se van a utilizar no se incorporan al sistema. Existen muchos tipos de módulos, de entre los más comunes cabe destacar: Gestión Financiera, Gestión de Ventas, Gestión de Compras, Gestión de la Distribución y Logística, Gestión y Planificación de la Producción, Gestión de Proyectos, Gestión de Recursos Humanos,...

- *Adaptable:* los procesos que se introduzcan se pueden configurar para que el sistema ERP se ajuste a las características de cada empresa. Esto se hace mediante la parametrización de los procesos de acuerdo con las salidas de cada uno. Esta adaptabilidad o flexibilidad permite que el sistema se pueda modificar para desempeñar funciones distintas de aquellas para las cuales fue diseñado, consiguiendo un cambio simultáneo con el entorno.

  Otras características de este tipo de sistema son:

- Conectividad: el sistema no se limita al espacio físico de la empresa, sino que permite la conexión con otras entidades (clientes, proveedores, etc.).
- Permite la selección de diferentes formas de negocio.
- Simulación de la realidad: se elaboran los documentos necesarios para la ayuda a la toma de decisiones mediante distintas simulaciones de la realidad empresarial.
- Sirve para realizar evaluaciones y seguimientos de procesos y otras actividades en la empresa.
- En un sistema ERP los datos se ingresan sólo una vez y deben ser consistentes, completos y comunes.

En la Figura 1.1 se muestra la relación entre el sistema ERP y los diferentes departamentos de la empresa, haciendo de nexo de unión entre todos ellos y proporcionando la coherencia necesaria al conjunto.

Figura 1.1: *Relación entre los departamentos y el ERP.*

1.2.2. Historia de los ERP

Los sistemas de Control y Planificación de Manufactura (MPC, *Manufacturing Planning and Control*) tienen su origen en la Revolución Industrial y su fin era la automatización de diferentes tareas, mejorando la exactitud y la predictibilidad de la producción.

En la Segunda Guerra Mundial, el Gobierno de los Estados Unidos creó programas especializados en los grandes ordenadores (que se comenzaron a introducir en la década de los 40). Dichos programas servían para llevar un seguimiento de la logística de las unidades bélicas.

A finales de los 50, se comenzaron a utilizar estos procedimientos en un ámbito empresarial, dejando atrás las funciones en el ejército estadounidense. Pasó a tener un papel protagonista el ROP (Punto de Reorden, por sus siglas en español), que se comenzó a implementar en los primeros mainframes. En un principio se incluyeron las funciones más básicas como el control de inventario, facturación y pago y administración de nóminas.

A mediados de los 60 surgen los sistemas de Planificación de Requerimiento de Materiales (MRP, *Material Requirement Planning*), que sustituyeron a los sistemas ROP. Los MRP ofrecían una búsqueda hacia adelante, es decir, tanto la planificación como el control de la fabricación

estaban fundamentados en la demanda. También introdujeron herramientas de reporte de producción muy básicos, pero computarizados. Estos informes se usaban para ajustar la demanda prevista.

En los años 80 los MRP II (*Manufacturing Resource Planning*) ocuparon el lugar de los sistemas MRP anteriores. A diferencia de estos, reconocían que en las empresas podían ocurrir paradas en la producción y limitaciones en los recursos. Estos sistemas añadieron a los MRP la planificación de los requerimientos de capacidad (CRP, *Capacity Resource Planning*), pudiendo integrar las capacidades con la demanda.

En general, todos estos métodos pretendían facilitar la planificación y hacer que las operaciones de fabricación fueran más eficientes. Los sistemas ROP, MRP y MRP II utilizaban computadoras mainframe, bases de datos jerárquicas y sistemas de procesamiento de transacciones complejas. Esto se ajustaba a la producción de pocos productos en elevado volumen y con escasa variación de la demanda, lo que conducía a problemas de inflexión cuando se variaban estas condiciones.

Como consecuencia de los problemas surgidos y por la evolución de las tecnologías se comienza a desarrollar un entorno de producción más dinámico, en el que los productos y procesos cambian más rápidamente. La solución llegó a principios de los años 90 con los nuevos sistemas MES (*Manufacturing Execution Systems*), que unifican los procesos de fabricación con los requerimientos del cliente a través de un sistema de valor de entrega. Así se mejoró la flexibilidad, se consiguió la ejecución en tiempo real, la retroalimentación y un mejor control de los procesos de fabricación.

A principios de los 90, las soluciones ERP ligaron diversas tareas críticas en un sistema de intercambio de información único, permitiendo la compartición de datos relativos a las operaciones y a la empresa.

En los años finales de esta década, se produjo un incremento en los niveles de competitividad global. Esto junto con los avances tecnológicos, hizo que muchas compañías optaran por renovar sus productos y servicios, así como su estructura y operaciones. De esta manera, las entidades inician un proceso de mejora continua, en parte gracias a los sistemas ERP que permiten mejorar los procedimientos y lograr la integración funcional. Se puede ver un resumen de la historia de los ERP en la Figura 1.2:

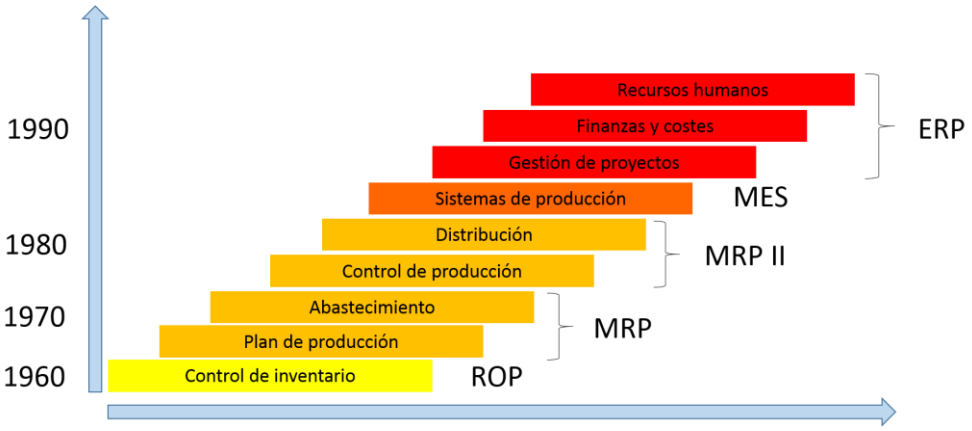


Figura 1.2: *Evolución de los sistemas ERP.*

1.2.3. Ventajas y desventajas de los ERP

Por un lado, un sistema ERP posee muchas *ventajas* para la organización que lo implanta:

- Con él, todas las áreas de una empresa son gestionadas de forma simultánea y a tiempo real de manera eficiente y fiable.
- Se ajusta a las necesidades de la empresa, por lo que se puede optimizar para sus propias actividades y departamentos.
- Previene de errores en los datos, porque son introducidos una sola vez y son comunes para todas las áreas del negocio.
- Almacena con seguridad toda la información de la empresa ante ataques del exterior y posibles estafas provenientes del interior de la organización.
- Acelera los procesos, mejorando la calidad de la empresa. Esto se puede reflejar en la satisfacción de los clientes.
- Ofrece la posibilidad de acceder desde diferentes dispositivos al servicio.
- Minimiza errores porque se automatizan los procesos.
- Unifica los recursos TI de la empresa, facilitando el mantenimiento de los sistemas. Como se utiliza un único programa para la gestión, se reducen los gastos en estos recursos.
- En muchos casos, moderniza la estructura administrativa y estandariza los procesos, mejorando el conocimiento que se tiene de ellos.
- Puede crear nuevas oportunidades de venta.
- Permite la medición continua de resultados.

Las principales funciones de los sistemas ERP se muestran de manera resumida en la Figura 1.3.

Figura 1.3: *Principales funciones de un sistema ERP.*

Por otro lado, también hay *desventajas* en un ERP:

- Es un sistema de difícil implantación, ya que necesitan un desarrollo personalizado para cada empresa partiendo de la parametrización inicial de la aplicación. Las personalizaciones y desarrollos particulares requieren un gran esfuerzo en tiempo y en dinero.
- Es necesario formar al personal de la empresa en el uso del ERP que se va a instalar. El éxito o no de la implantación vendrá dado por las habilidades y experiencia de los trabajadores y el conocimiento que adquieran durante los ciclos de formación (suelen venir incluidos en el precio).
- Normalmente, los sistemas ERP requieren licencias de uso que son caras y además hay que renovarlas independientemente del tamaño o las ganancias de la empresa. Bien es cierto que los costos del hardware y software comienzan a reducirse rápidamente por el desarrollo y la competencia del mercado tecnológico.
- Un fallo de uno de los empleados o en uno de los departamentos en el uso del ERP, se puede extender a toda la empresa (problema del "eslabón más débil").
- Se pueden dar incompatibilidades con otros sistemas y programas que utilice la empresa.

- Las modificaciones, una vez instalado el sistema, suelen ser bastante costosas.
- Con un sistema ya instalado, el coste de cambio a otro es muy elevado, con consecuencias importantes para la compañía.
- Para ciertas organizaciones, la estructura y funcionamiento pueden ser excesivamente complejos.
- En cierta medida y en ocasiones, algunas de las estructuras, denominaciones, fichas de proceso, etc., que utiliza la empresa pueden tener que adaptarse al nuevo sistema. Alinear los procesos de la organización con los del ERP se denomina Reingeniería de Procesos.

1.2.4. Tipos de ERP

Se pueden distinguir entre tres grandes clasificaciones de los ERP: sistemas propietarios, sistemas ERP *Opensource* (de software libre) y SaaS (*Software as a Service,* o software como servicio en castellano). A continuación se realiza una descripción de cada uno de los tipos con sus aspectos positivos y negativos.

1.2.4.1. Sistemas propietarios

Se denominan sistemas propietarios a las soluciones ERP que requieren de una licencia para poder utilizarse. En la adquisición de la licencia se suele pagar una parte fija y una parte variable dependiendo del número de puestos para los que el sistema va estar operativo. Esto hace que este tipo de soluciones sean bastante caras, en muchos casos inalcanzables para las pequeñas y medianas empresas.

Las grandes empresas desarrolladoras de software son las que proveen este tipo de medios, por ejemplo SAP y Microsoft. Sus productos se encuentran ya en estado de madurez, ofrecen un gran soporte técnico y suelen incluir cursos para que los empleados aprendan a trabajar con el software.

También existen empresas de tamaño más reducido que pueden proporcionar soluciones ERP. En este apartado destacan Deister y Solmicro. Su servicio suele estar más orientado a un sector en concreto, a diferencia que en el caso anterior, que son de carácter más general.

Las empresas que adquieren un ERP de sistema propietario tienen limitaciones para usarlo, modificarlo y redistribuirlo. El código fuente (conjunto de instrucciones que sigue la computadora para ejecutar un programa) puede

estar o no disponible, pero en caso afirmativo, se encuentra protegido por un acuerdo de licencia o tecnología anticopia. El código le pertenece a una persona física o jurídica (compañía, corporación, fundación…) que posee los derechos de autor. Normalmente sólo se le permite al cliente ejecutar el software en ciertas condiciones.

Las ventajas con las que cuentan este tipo de sistemas son:

- Han aprobado diferentes controles de calidad en los departamentos de las grandes empresas desarrolladoras.
- Se han invertido gran cantidad de recursos en su desarrollo e investigación.
- El personal que ha trabajado en el software está altamente capacitado.
- Es fácil encontrar información y ayuda acerca de un software tan conocido.
- Se pueden utilizar para aplicaciones muy específicas.
- Son sistemas fáciles de adquirir, ya que existen en tiendas o se encuentra preinstalado en el dispositivo hardware.
- Una gran compatibilidad multimedia y de hardware.
- Las interfaces gráficas están muy bien diseñadas.

Entre los inconvenientes encontramos:

- Los cursos de aprendizajes son costosos y necesarios para utilizar el software propietario eficientemente.
- El código fuente está fuertemente restringido. En muchas ocasiones es comparable a trabajar con una caja negra.
- Es muy complicada la adaptación de un módulo a las necesidades particulares de la empresa. Es necesario encargarlo al proveedor del software con un coste realmente elevado, ante la imposibilidad de hacerlo internamente.
- Imposibilidad de copia, modificación y redistribución.
- Restricciones de uso incluidas en la licencia.
- Excesiva dependencia de la empresa proveedora, tanto para el soporte técnico como para la formación.
- No existen aplicaciones para todas las plataformas (Windows y Mac SO).
- El coste en general es muy elevado.

Para finalizar este apartado, es importante destacar el poder que tienen las empresas proveedoras de sistemas propietarios, que les permite tener gran solvencia financiera y ofrecer servicio de gran calidad, lo que incluye un

contacto rápido y eficaz, realizar mejoras a sus productos y poner a disposición de sus clientes cursos de aprendizaje para el uso del software.

1.2.4.2. Sistemas de software libre

A diferencia de lo que puede parecer, los sistemas ERP *Opensource* o de software libre no son gratuitos. Estos programas son desarrollados por empresas que suelen tener una comunidad de *partners* que ofertan los servicios de implantación, configuración, parametrización y formación de usuarios.

El código abierto que utilizan permite mayor libertad y oportunidad para la implantación y la formación, ya que no tiene que ser la misma empresa proveedora la que capacite a los empleados para trabajar en él. Esta libertad también se extiende a las posibilidades de la modificación del código y su distribución. Según la *Free Software Foundation*, se da libertad a los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

Entre las ventajas que presentan este tipo de sistemas destacan las siguientes:

- Existen aplicaciones para todas las plataformas (Linux, Windows, Mac SO).
- El precio de las aplicaciones es reducido; normalmente son gratuitas.
- Libertad de copia, modificación y redistribución.
- Libertad de uso con cualquier fin.
- Posibilidad de diferentes traducciones.
- Independencia del autor del software.

Los aspectos negativos de este sistema de software libre son:

- Las aplicaciones en Linux son complicadas de instalar.
- Inexistencia de garantía por parte del autor.
- Interfaces gráficas menos intuitivas y más rudimentarias.
- Menor compatibilidad con el hardware.
- Poca estabilidad y flexibilidad en el campo de multimedia.

1.2.4.3. Sistemas SaaS

Está surgiendo un nuevo modelo de entrega y mantenimiento de software, "el software como servicio" o SaaS (*Software as a Service*). Además de la instalación se hace cargo del servicio técnico y la ayuda necesaria para el cliente, que tiene el sistema hospedado en la compañía de IT. El acceso y la administración se realizan a través de una red, los clientes tienen acceso remoto a las aplicaciones mediante web (de forma no directa). Esta distribución (parecida al modelo "uno a muchos") permite que se apliquen actualizaciones centralizadas, por lo que el usuario no tiene que parchear el sistema. No se paga por la licencia, sino por el uso de los servicios.

El software se puede distribuir por este método a cualquier sector del mercado, desde consumidores caseros hasta corporaciones. La modalidad SaaS es compatible con los tipos Propietario y *Opensource*, es decir, puede existir un ERP SaaS propietario y un SaaS de software libre.

Los aspectos favorables de este tipo de programa son:

- La empresa IT contratada se encarga de la instalación, puesta a punto, mantenimiento, operaciones y servicios. Por ello, la empresa cliente no tiene que dedicarse a ello ni asignar un área para estos fines, pudiendo ser más barato que otras soluciones y con menor riesgo.
- Se garantiza la disponibilidad y correcta funcionalidad del servicio por la atención continua por parte de la compañía proveedora del software.
- Ofrece servicio de protección de datos.
- Permite libertad y flexibilidad en cuanto a sistema operativo.

Los inconvenientes que presentan son:

- En función del contrato con la empresa IT, la empresa cliente no puede realizar modificaciones en el programa. De la misma manera, como los datos del cliente están almacenados por parte del proveedor, el usuario no puede acceder directamente a ellos, poniendo en peligro su seguridad y privacidad.
- No existe la posibilidad de migrar a otra solución ERP, al menos a corto plazo, ya que el servicio y el programa ya dependen de una compañía. También depende del tipo de contrato pero, de todos modos, acarrearía un aumento importante de los costes.
- Dependencia de la conexión a internet. Si la conexión falla por algún motivo, el cliente no podrá acceder al programa hasta que se restaure el servicio.

- Posibilidad de falta de entendimiento entre las empresas. Hay que tener en cuenta lo que implica dejar en manos de un tercero el servicio de gestión de la información de la propia empresa.

1.2.5. Elección e implantación de un ERP

A la hora de elegir el ERP más apropiado para implantar en nuestra empresa, lo más aconsejable sería realizar un sondeo de los ERP existentes en el mercado. En esta búsqueda conviene tener en cuenta fuentes de información externas (webs de ayuda, publicaciones de información especializada, etc.) y acudir a los resultados de las investigaciones hechas por los profesionales de Tecnología de la Información (IT).

La tarea de elección final no debe provenir total y directamente de un tercero. Las personas responsables de la empresa son las que mejor conocen sus actividades y funcionamiento y pueden determinar qué solución se ajusta más a sus necesidades. En cuanto al ámbito económico, la decisión de implantar uno de estos sistemas en la empresa debe ser considerada una inversión.

Los factores que influyen en la decisión de la adquisición de un sistema ERP son:

- Comprobar si la solución ERP se ajusta o no a lo que requiere la empresa, es decir, asegurarse de que la herramienta incluye todas las funcionalidades necesarias.
- Informes de consultoría que se hayan solicitado. Estudio de mercado y opciones posibles y viables.
- Precio del sistema: en este apartado hay que tener en cuenta el precio total del sistema o, si éste viene en módulos, el precio de adquisición de aquellos en los que estemos interesados. Se debe comprobar la existencia de posibles descuentos para futuras adquisiciones de las versiones siguientes del software.
- Mantenimiento del sistema: si la empresa en la que se va a instalar no tiene gran capacidad técnica en este tipo de soluciones o carece de experiencia, será de valorar las soluciones que requieran menor mantenimiento o que éste sea realizado por profesionales externos.

En la Figura 1.4 se muestra una comparativa de los tres tipos de ERP analizados en los apartados anteriores:

	Propietarios	Software libre	SaaS
Dificultad de uso	Alta	Media	Media
Coste de adquisición	Muy alto	Bajo	Moderado
Rapidez de despliegue (instalación)	Media	Media	Rápida
Coste de mantenimiento	Alto	Medio	Bajo
Capacidad de personalización	Baja	Alta	Baja
Disponibilidad de información	Buena	Buena	Media
Versatilidad	Media	Buena	Media

Figura 1.4: *Comparativa entre los tipos de ERP.*

La conclusión que se puede extraer de la tabla es que los ERP de sistema propietario presentan mayores desventajas con respecto a los otros dos tipos, aunque no hay que infravalorarlos, ya que por algo los utilizan muchas de las grandes compañías.

Una creencia bastante extendida es que un software libre de gravámenes legales no puede competir con uno más popular y desarrollado por una gran empresa. Esto no es así; de hecho, quienes hayan utilizado herramientas de software libre en el ámbito de gestión empresarial saben por experiencia propia que este tipo de soluciones supera las expectativas y consigue unos resultados equiparables e incluso mejores a los sistemas propietarios.

Los ERP de sistema propietario están orientados más bien a las grandes compañías y presentan menor flexibilidad. La realidad es que hoy en día la mayor parte de las empresas son PYME y sufren multitud de cambios, ya sea en la demanda externa o en factores internos de la asociación. Los ERP de software libre han sabido adaptarse a esta realidad y se han enfocado hacia este mercado, permitiendo un gran crecimiento en poco tiempo.

Por ello, se han podido dedicar mayores esfuerzos a la mejora de los servicios y productos, aumentando la calidad y permitiendo actualizaciones frecuentes (lo que ayuda a adaptarse a esa variabilidad comentada anteriormente). Esta evolución constante permite el avance en tecnologías de última generación que se adaptan a las necesidades cambiantes del mercado actual.

Todo esto ha permitido a los sistemas de software libre ganar terreno a los propietarios y superarlos. Debido a este motivo, las firmas privadas se están fijando cada vez más en los adelantos de las empresas que ofertan productos *opensource*, intentando mejorar ellas mismas.

En cuanto a la competencia de ambos sistemas tradicionales con los que se utilizan como servicio, hay que tener presentes diferentes factores que influyen en la elección:

- Personalización y control: los sistemas propietarios ofrecen un mayor nivel de ajuste por parte de la empresa cliente, en cambio los sistemas SaaS poseen menos herramientas de personalización. Esto tiene repercusión directa en la flexibilidad que ofrecerá la solución final.
- Sencillez: como es lógico, las soluciones SaaS no tienen mayor complejidad de uso porque no se necesitan habilidades técnicas (ya las aportan los proveedores del sistema). Los ERP propietarios requieren del trabajo de empleados de la empresa cliente.
- Accesibilidad: en los sistemas SaaS es necesario una conexión a internet para poder estar en contacto con el servidor externo (cada vez más se está avanzando hacia el *cloud computing*). En un ERP más tradicional solamente es necesaria una red interna de la empresa.
- Coste: en este sentido, la solución SaaS puede ser más barata a corto plazo ya que no son necesarios grandes cambios en la estructura empresarial, pero a largo plazo el coste se incrementa considerablemente por el precio de los servicios. En los propietarios puede llegar a ser necesaria la contratación de personal técnico especializado (por eso se adecua más a grandes empresas con mayor capacidad para contratar o que ya cuenten con un área o equipo que se hagan cargo de esas tareas). Teniendo en cuenta la contratación y mantenimiento en el tiempo, a la larga un sistema propietario puede resultar más rentable.

Resumiendo, los sistemas SaaS surgieron ante la necesidad de las empresas medianas y pequeñas de instalación de un ERP ante la imposibilidad técnica y económica de adquirir uno de tipo tradicional.

Una vez vistos con detenimiento los tipos de ERP existentes, características, ventajas y desventajas, es el momento de elegir cuál será el sistema que se utilizará en la realización del presente trabajo. La opción más adecuada es un ERP de software libre por las condiciones favorables que presenta. Además, este trabajo está orientado a una pequeña o mediana empresa y como se ha comentado en diversas ocasiones a lo largo del capítulo, el sistema *Opensource* es el más adecuado para este tipo de organización.

Ahora sólo queda seleccionar cuál de los ERP de software libre existentes en el mercado es el que se va utilizar. Para ello, se muestran a continuación los más destacables dentro de la categoría:

- **Adempiere:** incluye todas las funciones básicas de ERP, como la planificación de recursos, administración de la relación con los clientes (CRM) y gestión de la cadena de suministro (SCM). Es un completo sistema de gestión empresarial que soporta base de datos PostgreSQL y permite generar documentos PDF. El logotipo de Adempiere aparece en la Figura 1.5.

Figura 1.5: *Logotipo del ERP Adempiere.*

- **CK-ERP:** tiene 20 módulos con las funciones de negocio básicas e imprescindibles. Muestra una interfaz gráfica muy sencilla e intuitiva. Adecuado para un primer contacto con los sistemas de gestión empresarial. Es totalmente configurable y adaptable a las necesidades de la empresa. El gran inconveniente es que funciona en todos los sistemas operativos excepto en Windows (el más extendido). En la Figura 1.6 se muestra el logotipo de este ERP.

Figura 1.6: *Logotipo de CK-ERP.*

- **GNUe:** sistema muy sencillo con un conjunto de aplicaciones coordinadas que le permiten trabajar como un ERP. No obstante, cada una de ellas se ha desarrollado por separado y no es la opción más adecuada. Su logotipo se expone en la Figura 1.7.

Figura 1.7: *Logotipo de GNUe.*

- **Openbravo:** destinado a empresas de pequeño y mediano tamaño con dos versiones: *Community* (de libre distribución y con acceso al código abierto, no sin restricciones) y Profesional (código propietario y con todas las funciones activas). Consiste en una aplicación de arquitectura cliente/servidor web escrita en Java. Admite bases de datos PostgreSQL y Oracle. Su distintivo está visible en la Figura 1.8.

Figura 1.8: *Logotipo del ERP Openbravo.*

- **SugarCRM:** herramienta ideal para administrar empresas pequeñas de forma simple. También adecuada para las dedicadas a los servicios por la función de agenda de contactos, gestión de ventas, etc. Fue desarrollado en base a PHP y MySQL. Se puede apreciar el logotipo de este software en la Figura 1.9.

Figura 1.9: *Logotipo de SugarCRM.*

- **OpenERP:** software ERP desarrollado especialmente para cubrir las necesidades de pequeñas y medianas empresas. Puede ser utilizado en sistemas operativos Windows y Linux. Cuenta con una lista muy amplia de módulos descargables y utilizables. El lenguaje de programación utilizado es Python, se acopla sobre PostgreSQL y usa XML y librerías QT. La figura 1.10 ha sido su marca distintiva desde que en 2008 dejó de llamarse *TinyERP*. En junio de 2014 ha pasado a denominarse *Odoo*.

Figura 1.10: *Logotipo de OpenERP.*

- **Tryton:** es una aplicación de tres capas que usa licencia GPL-3. Está escrito en lenguaje Python y usa PostgreSQL como base de datos. Permite diseñar complejas soluciones empresariales gracias a que es modular, escalable y seguro. Tiene un lado negativo: las dificultades de implantación. Su logotipo es el que aparece en la imagen de la Figura 1.11.

Figura 1.11: *Logotipo de Tryton.*

Una vez vistas las distintas posibilidades de software libre existentes, se puede elegir cuál de ellos será el utilizado en el presente trabajo. El seleccionado es OpenERP, ya que abarca un abanico muy amplio de posibilidades, es relativamente sencillo y potente. Además, se adapta perfectamente a las necesidades que pueda tener una empresa no muy grande, hacia la que va dirigido el resultado de este proyecto.

CAPÍTULO II

OPENERP Y SU CONFIGURACIÓN

En el primer capítulo se mostró una visión general acerca de la empresa y porqué necesitaban de los ERP. En cuanto a estos sistemas, se explicaron sus características, historia, ventajas y desventajas, tipos y, por último, se exhibieron los aspectos a tener en cuenta en la elección e implantación de la solución ERP adecuada a la empresa. Como conclusión, se expusieron los motivos que impulsaron hacia la selección final, la de OpenERP.

Este capítulo está orientado a la explicación y desarrollo de los principales atributos a nivel funcional de OpenERP. Se estudiará la forma de trabajar con el programa así como su configuración y control. Todo esto desde la interfaz del sistema, aún sin entrar a modificar el código fuente.

2.1. OpenERP

La versión de OpenERP con la que se va a tratar es la última hasta la fecha. Se trata de OpenERP 7.0. En este apartado se dará una visión general de este ERP, centrándose en la licencia, arquitectura, el entorno de desarrollo y las diferencias más destacables frente a la versión anterior.

2.1.1. Licencia

Por un lado, la mayor parte de los módulos que componen el sistema existen bajo la licencia AGPL, (*Affero General Public License,* licencia pública general de Affero). Esta es una licencia *copyleft* que proviene de la Licencia Pública General de GNU. La GNU es la más usada en el mundo del software y garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software.

Por otro lado, la otra fracción de módulos utiliza la *Mozilla Public License*, que es una licencia de código abierto (de la *Open Source Initiative)* y de software libre (cumple con las cuatro libertades enunciadas por la *Free Software Foundation*), aunque se deja abierta la posibilidad de restringir la reutilización del código.

El estar bajo estas licencias libres permite que OpenERP no precise del desembolso de ninguna tasa para ser utilizado, lo que le da su condición de software *Opensource.* Mientras se respeten los términos de dichas licencias se podrá disfrutar de un uso sin restricciones y será posible la modificación inmediata de la herramienta.

2.1.2. Arquitectura

OpenERP es un programa multiusuario con una arquitectura de programación por capas. Formado básicamente por un núcleo y unos módulos instalados alrededor de éste. Tanto el núcleo como los módulos están escritos en Python.

2.1.2.1. Arquitectura cliente servidor

El software presenta una arquitectura cliente-servidor típica. El servidor se encarga de manejar la lógica de negocio y está relacionada con la base de datos. Por su parte, el cliente muestra la información a los diferentes usuarios y hace de nexo de unión entre éstos y el servidor.

Los servidores dan respuesta a las peticiones de los usuarios. Todos los clientes están conectados a un servidor, que contiene los recursos y multitud de aplicaciones cliente y los pone a disposición de los usuarios cuando lo solicitan. Entonces el servidor es el que concentra y almacena todas las gestiones y requerimientos. Gestiona las peticiones prioritarias, los archivos de uso público y restringido, los archivos de sólo lectura y los modificables, etc.

Una ventaja muy importante de este tipo de funcionamiento es que puede utilizarse conjuntamente, algo indispensable para un software del tipo de OpenERP. Varios usuarios pueden interactuar al mismo tiempo con un mismo servidor, pudiendo compartir información y espacio de trabajo.

Por ejemplo, en la lógica de negocio, que se ocupa de las tareas relacionadas con los distintos procesos que se dan en un negocio, como las compras y el control del inventario. Consiste básicamente en rutinas de entrada, consulta y procesamiento de datos. El software se comunica con el usuario a través de una interfaz para mostrar los resultados de dicho proceso. Un esquema del funcionamiento de este sistema se puede observar en la Figura 2.1 que ha sido extraída de la sección de OpenERP de Wikimedia Commons.

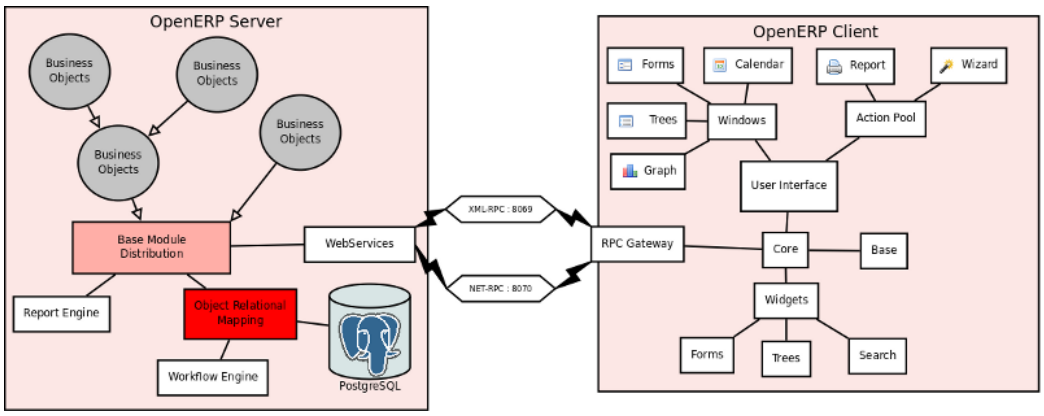


Figura 2.1: *Esquema de la arquitectura cliente-servidor.*

2.1.2.2. Lenguajes utilizados en OpenERP

OpenERP se vale de dos tipos de lenguaje para funcionar: Python y XML. A continuación se pasa a realizar una breve presentación de cada uno de ellos:

- *Python*: es un lenguaje de programación de alto nivel. Está pensado para favorecer la limpieza en la sintaxis consiguiendo un código fácilmente legible. Es un lenguaje muy versátil ya que está orientado a objetos, pero también puede usarse para el desarrollo web o incluso para aplicaciones Windows. Además, soporta programación imperativa y, en menor medida, programación funcional, lo que le convierte en un lenguaje multiparadigma.
  Se trata de un lenguaje interpretado, por lo que es más rápido de desarrollar. Otra característica de Python es que es dinámicamente tipado, es decir, una misma variable puede tomar valores de distinto tipo en diferentes momentos.
  A todas las ventajas expuestas en este epígrafe se añade que Python es multiplataforma y gratuito, incluso en entornos empresariales.
  Python es administrado por la *Python Software Foundation License*, que es compatible con la Licencia Pública General de GNU.

- *XML*: es un lenguaje de marcas que se utiliza para almacenar datos de forma legible. Proviene del lenguaje SGML (*Standar Generalized Markup Language*, o Estándar de Lenguaje Marcado Generalizado) y permite definir la gramática de lenguajes específicos para estructurar grandes documentos. Sirve de soporte para bases de datos y es especialmente útil para la comunicación entre aplicaciones o para la integración de la información. Es una tecnología sencilla que se complementa con otras

para mayores prestaciones y se puede usar en multitud de aplicaciones como bases de datos, editores de texto, hojas de cálculo, etc.

2.1.2.3. Servidor y módulos

Como ya se ha dicho, el servidor y los módulos están escritos en código Python. El cliente se relaciona con el servidor a través de XML-RPC, que es un protocolo de llamada a un procedimiento remoto. Usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. Según esta especificación los tipos de datos son: *array*, *base64*, *boolean*, *date/time*, *double*, *integer*, *string*, *struct* y *nil*.

La funcionalidad de negocio se organiza en módulos, que se pueden entender como archivos con una estructura predefinida, escritos en Python y que contienen otros archivos XML. En la Figura 2.2 se resume la estructura de distribución de los módulos.

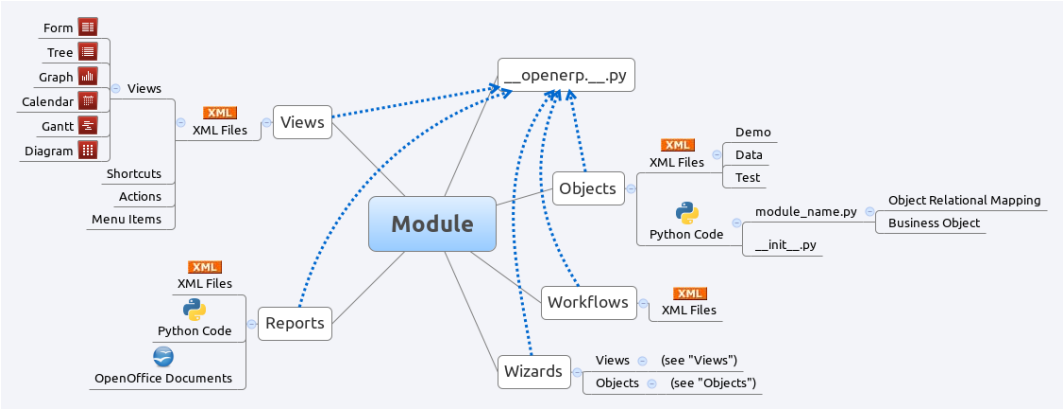


Figura 2.2: *Esquema del funcionamiento de un módulo.*

Los módulos suelen hacer uso de la tecnología de Mapeo Objeto-Relacional (ORM) que permite diseñar cualquier base de datos y gestionarla desde una interfaz simple autogenerada y accesible en cualquier sistema operativo.

2.1.2.4. Aplicaciones cliente

Los modos en los que se puede disponer el sistema son dos: como aplicación web y como aplicación de escritorio.

- Aplicación web: implementada como servidor HTTP. De esta manera los usuarios se conectan a través de algún navegador de internet.
- Aplicación de escritorio: utiliza el kit de herramientas GTK+ y está escrita en Python.

Adicionalmente, en la comunidad han surgido otro tipo de clientes que se pueden utilizar como alternativa a estos dos.

2.1.2.5. Base de datos

El sistema gestor de bases de datos (SGBD) que utiliza OpenERP, y viene incluido en su descarga, es PostgreSQL. Se trata de un SGBD relacional orientado a objetos y libre, bajo licencia BSD (*Berkeley Software Distribution*), de escasas restricciones.

Un SGBD es un conjunto de herramientas para la introducción, modificación y extracción de la información de una base de datos. También suelen incluir funciones de análisis de los datos. Además, administran el acceso de los usuarios y facilitan la presentación de los datos en distintos formatos como informes, gráficos y tablas.

Tener un buen SGBD es importante, ya que se requiere una elevada fiabilidad en todo lo relacionado con la información de la empresa, ya sea por motivos de seguridad o de funcionalidad. El ERP va a extraer la información necesaria en cada momento y de forma oportuna de entre todos los datos almacenados en la base. El SGBD ayuda en esta labor, haciendo de intermediario con la base de datos.

2.1.2.6. Código fuente y contribuciones

El código fuente de OpenERP se aloja en Launchpad, que es una plataforma de desarrollo colaborativo de software libre por medio de un servicio gratuito de sitio web. Utiliza como sistema de control de versiones el software libre Bazaar, que está escrito en Python y forma parte de la GNU.

Se pueden realizar aportaciones y colaborar con la documentación utilizando también Launchpad. Existen sitios web destinados a recoger toda la documentación desde hace años.

Por ejemplo en *OpenERP Spanish Localization Project*, se realiza intercambio de información, reporte de bugs, contribuciones a los módulos o código fuente, adelantos de nuevas versiones, etc.

2.1.3. Entorno de desarrollo

La modificación y el desarrollo de módulos se realiza principalmente mediante la edición de archivos de los lenguajes citados anteriormente: Python y XML. Para dicha tarea se pueden utilizar editores como Eclipse, que contiene un conjunto de herramientas de programación de código abierto multiplataforma. IDLE Python (Figura 2.3) es otro sencillo entorno de desarrollo de archivos de Python, que también puede dar cabida a los archivos XML. Este será el usado durante el presente trabajo por su sencillez y por la posibilidad de editar ambos tipos de archivo en una misma plataforma.

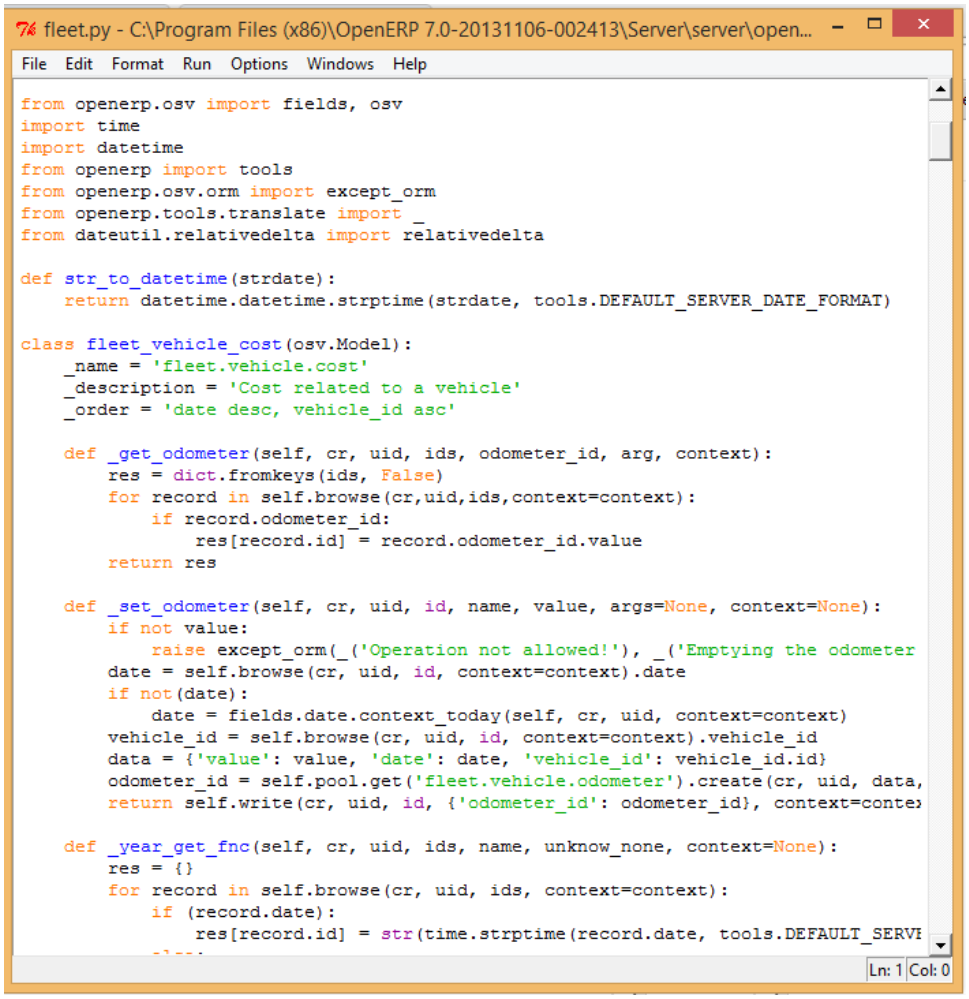


Figura 2.3: Interfaz IDLE Python.

Además, se pueden hacer pequeños cambios directamente desde la interfaz de OpenERP, pero sólo será posible alterar campos ya personalizados y no los campos base.

2.1.4. Cambios respecto a la versión anterior

A lo largo de la vida del software, han ido saliendo nuevas y mejoradas versiones del mismo. Este apartado está destinado a destacar las diferencias más destacables entre la versión 7.0 de OpenERP y la versión inmediatamente anterior, la 6.1. Las características más significativas son:

- Nueva interfaz web: mucho más simplificada e intuitiva.
- Configuración de los módulos: también se ha simplificado el proceso de modificación de los módulos y se han dividido en aplicaciones. Más adelante se explicarán las posibilidades y limitaciones a la hora de personalizar un módulo.
- Personalización de la barra de herramientas
- Unificación de los informes: ahora todos los reportes se encuentran en el mismo sitio y son más fáciles de consultar.
- La barra de búsqueda: permite agrupar los resultados de búsqueda, creando vistas más compactas y ordenadas.

2.2. Configuración

En este apartado se va a exponer un breve resumen acerca de las configuraciones más generales que se pueden hacer en la OpenERP con la que se ha trabajado. Como se dijo en la introducción de este capítulo, de momento no se va a entrar en el código del software, sino que esto sólo está orientado a la modificación de los aspectos más funcionales del programa.

2.2.1. Compañías

En OpenERP existe la posibilidad de gestionar más de una empresa al mismo tiempo. En el apartado “Compañías” dentro de la configuración general se pueden crear las diferentes compañías que estarán disponibles para su administración. Se pueden fijar multitud de características y relacionarlas con una gran cantidad de datos. Por ejemplo, para cada una de las entidades empresariales se pueden definir la dirección, sitio web, teléfono de contacto, cuenta bancaria, etc.

2.2.2. Usuarios

OpenERP está pensado para que diferentes usuarios de las distintas partes de la empresa, con diversos puestos y distinta formación utilicen la herramienta de la forma más conveniente para su trabajo. Por ello, existe la posibilidad de crear diferentes perfiles de usuarios en el mismo sistema. Incluso se puede situar a cada empleado en la compañía en la que ejerce su labor profesional.

En principio, es el usuario administrador el que ejerce la posibilidad de crear y configurar los distintos perfiles para los también variados empleados. Para cada uno de ellos se pueden definir: el nombre que dicho usuario tendrá en todo el sistema de gestión, el usuario con el que accederá al ERP, la compañía de la que forma parte y su estado (activo o no). Entre las preferencias se pueden modificar campos como el idioma en el que el empleado trabajará con el sistema, zona horaria (por si se encuentra en otra parte del mundo), seleccionar la acción inicial entre una gran lista de tareas para hacer y la acción de menú (a elegir según los módulos que se hayan instalado).

Otras de las preferencias será la posibilidad de determinar qué mensajes serán recibidos por correo electrónico, la dirección pública de email y la firma que tendrá en la compañía (véase Figura 2.4).

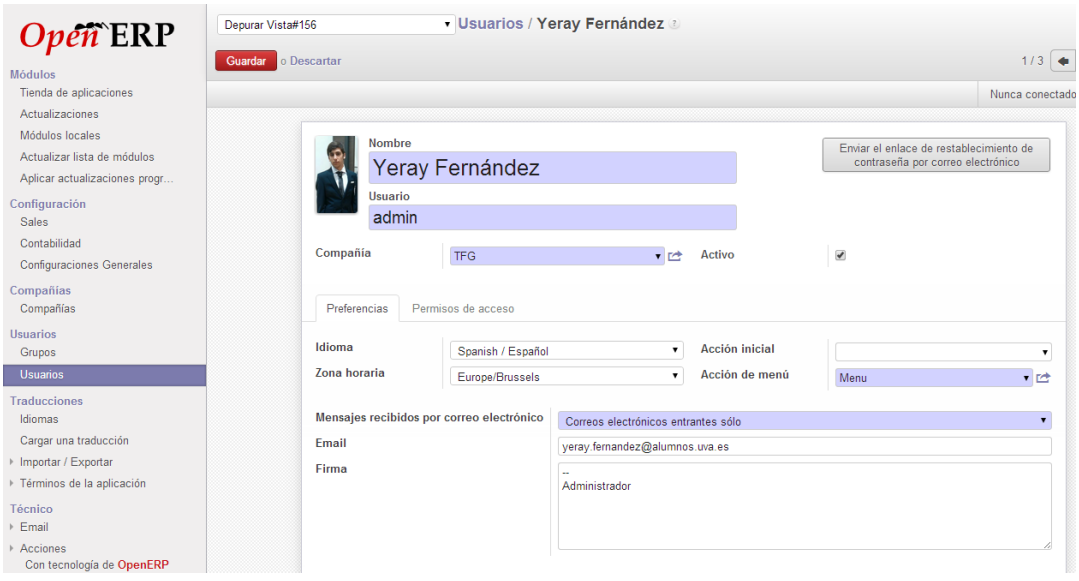


Figura 2.4: *Preferencias del usuario administrador.*

El otro gran bloque de elementos que se puede modificar para los usuarios es posiblemente el más importante. Se trata de los permisos de

acceso, con los que se puede restringir las partes del ERP a las que puede acceder el empleado y definir qué papel tendrá en aquellas aplicaciones en las que participa. Así podrá ser responsable de alguna aplicación, usuario de otra y encargado de la configuración de una tercera, por ejemplo.

En los permisos de acceso también se puede realizar la configuración técnica, que consiste en la activación de distintos apartados como la utilización de multidivisas, verificación de facturas de proveedor, gestionar múltiples unidades de medida o las facturas pro-forma.

En la "Usabilidad" se encuentran las opciones de "Múltiples compañías" y de "Características técnicas" que se explicarán más adelante.

En la última sección de permisos de acceso existen cuatro campos, a saber: creación de contactos, survey/user (para limitar los apartados a los que se tiene acceso), portal (otorga derechos de acceso específicos, tales como reglas de registro y menús restringidos) y anónimo (concede a los usuarios permisos específicos, pero no pertenecen a los grupos usuales de OpenERP). Se puede ver una plantilla de los permisos de acceso de un usuario en la imagen que se muestra a continuación, en la Figura 2.5.

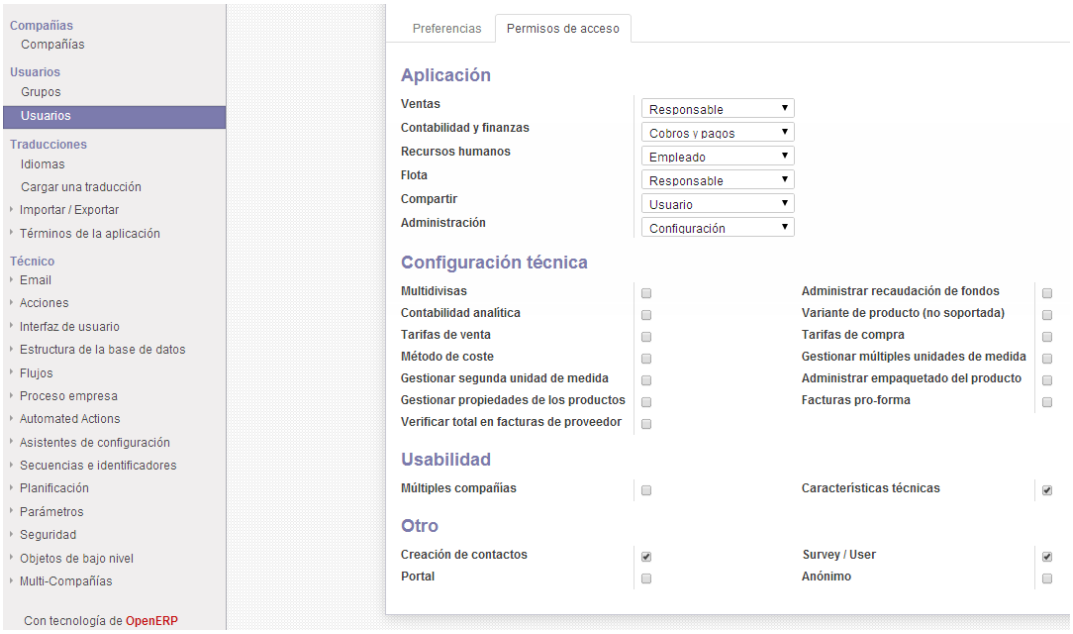


Figura 2.5: *Permisos de acceso del usuario.*

2.2.3. Traducciones

En las traducciones se pueden administrar las lenguas que aparecen en el sistema. En el apartado "Idiomas" se pueden ver los que ya están instalados y disponibles para su utilización y asignación. Para cada uno de ellos también se pueden modificar aspectos como los códigos (local e ISO) y formatos (separador, de fecha y de hora).

Para añadir idiomas nuevos no hace falta más que ir a la sección "Cargar una traducción" y allí seleccionar la lengua que se desee entre los disponibles en la lista (más de 70 incluyendo sus variantes). Si se marca la opción de sobrescribir términos existentes, las traducciones personalizadas serán sobrescritas y reemplazadas por las oficiales.

Otra posibilidad que ofrece OpenERP es la de importar traducciones que se tengan en el ordenador. La opción de exportar permite cargar un lenguaje a una aplicación concreta, para ello es necesario que dicho idioma tenga activada la traducción.

Dentro de "Términos de la aplicación" se pueden comprobar los términos que se encuentran traducidos y se encuentran a lo largo de todo el programa ya sea en campos, vistas, selecciones, ayudas,... En "Sincronizar términos" se puede sincronizar traducciones de aquellos idiomas que tengan la opción traducible activada. Se puede ver el menú de traducciones en la Figura 2.6.

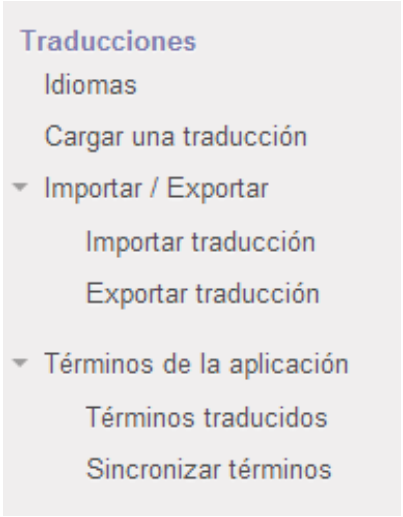


Figura 2.6: *Menú de traducciones.*

2.2.4. Módulos

Como ya se explicó en el capítulo primero, el ERP se divide en módulos que aportan muchas funcionalidades al sistema. En OpenERP 7.0 existen más de 1800 módulos que se pueden descargar e instalar desde su propia página web de forma libre. También se pueden conseguir directamente desde la interfaz del programa (véase Figura 2.7).

Figura 2.7: *Tienda de aplicaciones de OpenERP 7.0.*

Dependiendo de las necesidades particulares, es conveniente utilizar unas u otras aplicaciones. Según se vayan añadiendo módulos, irán apareciendo nuevos apartados en el menú general, de manera que las nuevas funcionalidades estarán listas para su configuración y uso.

A continuación se definen los módulos que se consideran más importantes en cualquier ERP y que, por supuesto, se encuentran en OpenERP:

- *Módulo CRM*: permite la gestión de las relaciones con los clientes (ya sean empresas o individuos), ayudando a su fidelización, debido a que estos obtendrán un servicio mucho más personalizado. También gestiona la relación con proveedores.
  Aquí es posible registrar y consultar toda la información relativa a los contactos, como, por ejemplo, facturas registradas, pedidos de presupuesto, oportunidades, direcciones, llamadas telefónicas, servicios post-venta, iniciativas, reclamaciones, etc.

Además, existen herramientas para incrementar la productividad y facilitar la conectividad, entre las que destacan un editor de documentos compartido con Open Office y la sincronización de contactos y calendarios con Outlook.

- *Módulo MRP*: facilita la gestión de la planificación productiva así como sus recursos. Elementos principales: mano de obra, materias primas, gestión de pedidos, suministros, costes de producción y estructura de planta.

- Módulo Recursos Humanos: ofrece funcionalidades para administrar todo lo relacionado con los empleados: rendimiento, perfiles, responsabilidades, contratos, calendario de vacaciones…

- *Módulo de ventas*: para la administración y control de pedidos, facturación, notas de crédito y débito, albaranes… El proceso que normalmente se sigue en OpenERP hasta que se produce una venta es el siguiente: iniciativa (primera intención de contacto con el cliente potencial, con recogida de información), oportunidad (cuando existe interés del cliente potencial), presupuesto (una vez se haya ganado la oportunidad a través de una negociación) y pedido de venta (cuando el cliente acepta el presupuesto y se genera la factura). Esto se resume en la Figura 2.8.

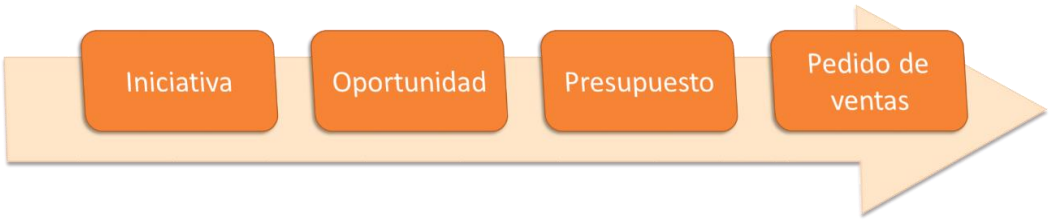


Figura 2.8: *Proceso general de venta en OpenERP.*

- *Módulo de almacén*: permite la gestión del inventario, tanto de materias primas y componentes como de productos. Incluye opciones para almacenes en diferentes ubicaciones. Es utilizado principalmente para controlar la rotación de inventario y los niveles de stock, valoración del stock, gestión de lotes y números de serie, ejecución de envío con albaranes de entrega y cálculo de gastos de envío, ejecución de las órdenes de empaquetado y definición de reglas para reabastecimiento. Se puede realizar un seguimiento exhaustivo en todo momento de las mercancías, ya que el movimiento se define con el origen y el destino.

- *Módulo de contabilidad y finanzas*: permite llevar al día la contabilidad, cobros y pagos, optimizando la gestión global de la empresa. La contabilidad en OpenERP es de doble entrada y se expande a las diferentes secciones de la compañía. Como no podía ser de otra forma, está adaptada a multitud de países, con sus normas y monedas. Los asientos contables se generan automáticamente por el sistema.
  Incluye: cuentas por cobrar, administración y control de créditos, tesorería (cajas y bancos), etc.

- *Módulo de gestión de proyectos*: ofrece facilidades para la planificación y control de los proyectos (y subproyectos) que tengan lugar en la empresa. Con él se pueden definir tareas, subdividirlas en trabajos y asignar recursos a los mismos de manera eficiente. Provee diagramas de Gantt para el seguimiento de los trabajos y para sincronizarlo con el calendario. Otras de las funcionalidades más importantes de este módulo son el control de costes y la posibilidad de comunicación directa con los *partners*.

Esto no es más que un breve resumen, pero como ya se ha comentado al principio hay cientos y cientos de variantes y opciones al alcance de la mano en OpenERP.

2.2.5. Informes

En OpenERP se encuentra preinstalado un potente gestor documental. Con él, se puede asociar cualquier tipo de archivo a cualquier objeto, lo que ayuda a mantener la información clasificada y unificada. Así se consigue un ahorro de tiempo y una mejor organización, lo que es muy importante en cualquier compañía.

En OpenERP todas las aplicaciones tienen su propia generación de informes y estadísticas dinámicas a las que se les puede aplicar filtros y se pueden agrupar y exportar.

Además, el sistema de generación de informes incluido en el sistema se puede mejorar incluyendo el diseñador de informes de Open Office. En esta versión es posible instalarlo directamente como aplicación de forma sencilla.

2.2.6. Técnico

En el apartado "Técnico" de OpenERP se abren nuevas posibilidades de configuración y personalización que no se plantean en otra parte del programa. Esta opción no es la que viene por defecto, pero ayuda en gran medida a la adaptación requerida por el usuario.

El menú que se encuentra en esta sección es el que se puede observar en la figura 2.9:

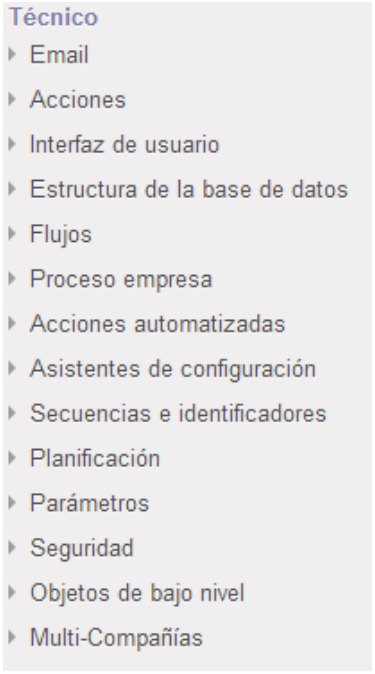


Figura 2.9: *Menú técnico.*

Entre las operaciones que se pueden realizar en este apartado destacan principalmente:

- *Acciones*: se pueden modificar las acciones ejecutadas en cada submenú. Ésta es una característica importante porque permite realizar cambios que en versiones anteriores habría que hacer en el código interno. Así se puede llevar a cabo de forma más sencilla y desde la propia interfaz de OpenERP.

- *Flujos y procesos de empresa*: desde aquí se pueden editar los diferentes procedimientos y etapas que tienen lugar dentro de la empresa. Consiste en hacer una recreación virtual del proceso que tiene

lugar en la realidad para ir dejando constancia de toda la información necesaria en la base de datos.

- *Acciones automatizadas*: permite cambiar el modo de funcionamiento de ciertas labores de manual a automático.

- *Asistentes de configuración*: sirve para determinar las acciones ejecutadas sobre los objetos o grupos y cómo se inician (manual o automáticamente).

- *Secuencias e identificadores*: muestra todos los registros y los modelos a los que pertenecen. Incluye su número de secuencia, su ID completo y el identificador externo.

- *Acciones planificadas*: se utiliza para gestionar las acciones que ya están programadas y modificar los tiempos. Estas acciones están incluidas en el calendario y por ello en este apartado existe una vista calendario.

- *Parámetros*: indica varias claves y su valor correspondiente. Por ejemplo, la dirección web desde la que se está conectado.

- *Multi-Compañías*: da opciones de configuración y gestión de las distintas organizaciones que estén registradas en el sistema.

Aunque la opción más importante para el objeto del presente trabajo reside en el apartado “Estructura de la base de datos”. La manera de proceder en este submenú se presenta a lo largo del cuarto capítulo.

CAPÍTULO III

COMPOSICIÓN DE UN MÓDULO EN OPENERP

En el capítulo anterior se realizó una introducción general al software de OpenERP y a su configuración más externa, se mostraron los aspectos comunes en cuanto a módulos y funcionamiento y se explicaron los principales menús.

Esto sólo sirve para cubrir las necesidades básicas de gestión de la empresa pero, como se ha mencionado en varias ocasiones, cada una tiene sus características y peculiaridades que pueden hacer necesaria la personalización, en mayor o menor grado, del software. Mediante la modificación de módulos se va a poder conseguir esa adaptación de las funcionalidades del sistema al funcionamiento real y particular de la compañía.

Por todo esto, en el presente capítulo se realizará un análisis de la estructura de los módulos de OpenERP. Se describirá la composición de los archivos que componen dichos módulos y las funciones de los elementos. Todo ello tiene como objetivo sentar las bases para poder realizar la personalización de un módulo para la gestión de una flota de vehículos que se realizará en el capítulo siguiente.

3.1. Introducción

El papel que juegan los módulos en un ERP es muy importante, ya que son la composición en sí del programa y permiten realizar las funciones de gestión para las que están pensados.

Dentro de la arquitectura cliente-servidor de OpenERP ya descrita con anterioridad, los módulos se sitúan en la estructura del servidor como las herramientas que permiten dar respuesta a las peticiones del cliente.

La estructura de los módulos está pensada para optimizar su funcionamiento como ORM y facilitar el uso normal de la herramienta.

3.2. Estructura de un módulo

Los módulos que acompañan a OpenERP se puede encontrar en el directorio: OpenERP 7.0/Server/server/openerp/addons. Allí aparecen los módulos instalados, cada uno con su carpeta propia. Dentro de cada uno encontraremos diferentes archivos y carpetas.

Todos los módulos han de estar formados, al menos, por los siguientes archivos:

- Archivo *_init_.py*
- Archivo *_openerp.py*
- Archivo *<nombre_modulo>.py*
- Archivo *<vista_modulo>.xml*

3.2.1. Archivo _init_.py

Éste es el fichero que indica la existencia de un módulo en el interior de esa carpeta y que permite cargarlo. En él se importan los demás archivos necesarios para el funcionamiento del módulo.

3.2.2. Archivo _openerp.py

Este archivo sirve básicamente para detallar: el nombre, descripción, autores, versión, licencia, categoría, etc., del módulo, así como los archivos XML utilizados en el módulo (que se comprueban en el arranque del servidor) y las dependencias con otros módulos.

Alberga un diccionario Python con la descripción del contenido del módulo, en el que se pueden encontrar[2] los siguientes valores principales:

- *name*: el nombre del módulo.
- *version*: la versión del módulo.
- *description*: una descripción del módulo.
- *author*: persona o grupo responsable del desarrollo del módulo.
- *sequence*: el orden en el que aparece en el menú el módulo.
- *website*: sitio web del desarrollador del módulo.
- *license*: tipo de licencia del módulo. La más común es GPL3.
- *category*: categoría a la que pertenece el módulo. Establece un directorio con la ruta jerárquica.
- *depends*: lista de dependencia con otros módulos. Normalmente entre ellos siempre está el módulo base.
- *init_xml o data*: es una lista de archivos de tipo XML que se cargan al iniciar el servidor OpenERP con la opción *"init=modulo"*. Las rutas definidas deben hacer referencia al directorio donde se encuentra el módulo. Los archivos aquí incluidos son los que definen el flujo de trabajo.
- *update_xml*: es una lista de archivos de tipo XML que se cargan al iniciar el servidor OpenERP con la opción *"update=modulo"*. Las rutas

[2] Algunos siempre aparecen y otros puede que no vengan especificados.

definidas deben hacer referencia al directorio donde se encuentra el módulo. Los archivos aquí incluidos son los relativos a las vistas, acciones, menús, informes y asistentes. Se cargan cuando se instala o actualiza el módulo.

- *demo*: es una lista de archivos de tipo XML que se cargan si el servidor OpenERP está instalado con datos de muestra. Las rutas definidas deben hacer referencia al directorio donde se encuentra el módulo. Los archivos aquí incluidos son los referentes a los datos de demostración.
- *images*: conjunto de imágenes que acompañan al módulo cuando se carga.
- *instalable*: establece si el módulo se puede instalar o no. Como valores sólo tiene *True* o *False.*
- *active*: determina si el módulo será instalado con la creación de la base de datos. Acepta los valores *True* o *False* (predeterminado).
- *application*: será *True* cuando el módulo tenga la condición de aplicación. Para ello tiene que cumplir básicamente dos requisitos: que sea un módulo oficial y certificado y que sea un módulo importante. Ésta es una diferencia con las versiones anteriores de OpenERP, donde no estaba clara la distinción entre módulo y aplicación.

En la Figura 3.1 se puede observar cómo es el código de uno de estos archivos y la declaración de los valores explicados en este apartado.

3.2.3. Archivo <nombre_modulo>.py

Éste es el fichero donde se definen los objetos que componen el módulo, tanto en las vistas como en la base de datos. Son objetos los diferentes recursos de los que se vale OpenERP: los menús, las acciones, los informes, los socios, las facturas, etc.

```
{
    'name' : 'Fleet Management',
    'version' : '0.1',
    'author' : 'OpenERP S.A.',
    'sequence': 110,
    'category': 'Managing vehicles and contracts',
    'website' : 'http://www.openerp.com',
    'summary' : 'Vehicle, leasing, insurances, costs',
    'description' : """
Vehicle, leasing, insurances, cost
==================================


Main Features
-------------
* Add vehicles to your fleet
* Manage contracts for vehicles
* Reminder when a contract reach its expiration date
* Add services, fuel log entry, odometer values for all vehicles
* Show all costs associated to a vehicle or to a type of service
* Analysis graph for costs
""",
    'depends' : [
        'base',
        'mail',
        'board'
    ],
    'data' : [
        'fleet_view.xml',
        'fleet_cars.xml',
        'fleet_data.xml',
        'fleet_board_view.xml',
    ],
    'images': ['images/costs_analysis.jpeg','images/indicative_costs_analysis.jpeg','images/vehicles.jpeg'
    'update_xml' : ['security/fleet_security.xml','security/ir.model.access.csv'],

    'demo': ['fleet_demo.xml'],

    'installable' : True,
    'application' : True,
}
```

Figura 3.1: *Ejemplo del contenido de un archivo __openerp__.py.*

Los objetos que componen y son utilizados por el módulo deben estar definidos en este archivo en una clase de Python de la forma: class nombre_clase (osv.Model). Posteriormente, hay que instanciar dicha clase, es decir, hay que definir los atributos y los métodos que tendrán los objetos de esa clase.

Estos atributos son usados e interpretados por OpenERP. Los obligatorios son:

- *name*: da nombre al objeto. Normalmente, para señalar a qué módulo pertenece. Se escribe al principio el nombre del módulo separado por un punto.
- *columns*: incluye los campos que aparecerán en las tablas de la base de datos y en las vistas.

En los atributos opcionales son especialmente comunes los siguientes:

- *_constraints*: establece ciertas restricciones a algún campo del objeto.
- *_sql_constraints*: establece ciertas restricciones SQL a algún campo del objeto.
- *_defaults*: establece los valores por defecto para un campo.

- *_inherit*: determina la herencia entre objetos.
- *_order*: resultado de la búsqueda y la lectura de métodos.
- *_rec_name*: da el nombre a cada recurso.

Como se ha dicho en el capítulo, los objetos de OpenERP contienen campos, que son utilizados para introducir los datos en la base de datos. La estructura es la de una tabla en la que se van añadiendo columnas para representar distintos valores de los objetos. Por ello, los campos están definidos en el atributo *columns*. Existen dos tipos de campos, los básicos y los relacionales.

3.2.3.1. Campos básicos

En los campos básicos se introducen los datos primarios y se pueden considerar los siguientes tipos de campos básicos:

- *boolean*: solamente puede tener valores *True* o *False*.
  Sintaxis: *fields.boolean ('string', [parámetros opcionales])*
- *integer*: es un número entero.
  Sintaxis: *fields.integer ('string', [parámetros opcionales])*
- *float*: es cualquier número real.
  Sintaxis: *fields.float ('string', [parámetros opcionales])*
- *char*: conjunto de caracteres. Requiere definir su longitud máxima.
  Sintaxis: *fields.char ('string', size=n, [parámetros opcionales])*
- *text*: permite introducir un texto sin límite de espacio.
  Sintaxis: *fields.text ('string', [parámetros opcionales])*
- *date*: para las fechas.
  Sintaxis: *fields.date ('string', [parámetros opcionales])*
- *datetime*: para las fechas y tiempo.
  Sintaxis: *fields.datetime ('string', [parámetros opcionales])*
- *related*: permite extraer un campo de una tabla relacionada.
  Sintaxis: *fields.related ('string', "type" [parámetros opcionales])*
- *function*: es un campo cuyo valor se calcula mediante una función en lugar de almacenarse en la base de datos. Aunque dicho dato también se puede guardar cuando *store=True*, pudiendo incluso almacenarlo en determinada variable. Se puede ver un ejemplo de ello en la Figura 3.2. No tiene por qué ser necesariamente un número.

```
'name': fields.function(_vehicle_name_get_fnc, type="char", string='Name', store=True),
```

Figura 3.2: *Ejemplo sencillo de un campo function en el módulo fleet.*

- *binary*: valor binario.
  Sintaxis: *fields.binary ('string', [parámetros opcionales])*
- *selection*: permite elegir un dato entre varios valores.
  Sintaxis: *fields.selection ((('n', 'Unconfirmed'), ('c','Confirmed')) 'string', [parámetros opcionales])*

Hay atributos que son comunes a estos campos y que actúan como parámetros de la función. Los más importantes son:

- *string*: este es el único atributo que es obligatorio. Es una cadena de texto que nombra la etiqueta del campo.
- *required*: marca la obligatoriedad del campo. Puede ser *True* o *False* (predeterminado).
- *readonly*: señala si el campo es de sólo lectura. Puede ser *True* o *False* (predeterminado).
- *select*: forma un índice para dicho campo si es necesario. El índice generado de PostgreSQL sirve para optimizar las búsquedas de las vistas: un 1 para búsqueda simple y un 2 para búsqueda avanzada. El valor por defecto es 0.
- *change_default*: permite al usuario cambiar los valores por defecto de los campos cuando sea *True*.

Existen algunos tipos de campos con atributos particulares:

- *translate*: puede ser *True* o *False*. En el primer caso indica que el campo puede ser traducido. Se aplica a los campos de tipo *char* y *text*.
- *size*: indica el límite de los caracteres en los campos tipo *char*.
- *digits*: permite seleccionar la precisión y la escala en los campos tipo *float* mediante una tupla.
- *values*: permite seleccionar entre los valores predefinidos en los campos de tipo *selection*. Este atributo está posicionado antes del atributo *string* y está definido por el conjunto de valores. Esa lista debe estar compuesta por tuplas con una clave y su etiqueta.
  Sintaxis: [('key1','Label1), ('key2','Label2'), ...],'string'...

3.2.3.2. Campos relacionales

Los campos relacionales son los que se utilizan para definir las relaciones entre los objetos. Las clases existentes son los siguientes:

- *one2many*: esta es la relación uno a muchos. Indica que un objeto está ligado a varios de otro tipo.
  Sintaxis:
  fields.one2many(
  'other.object.name',
  'field relation id',
  fieldname',
  optional parameter)

- *many2one*: establece una relación de varios objetos con otro que actúa como padre, estableciendo una categoría.
  Sintaxis:
  fields.many2one(
  'other.object.name',
  fieldname',
  optional parameter)

- *many2many*: es la relación de muchos a muchos entre dos clases de objetos.
  Sintaxis:
  fields.many2many (
  'other.object.name',
  'relation _object',
  'actual_object_id',
  'other_object_id',
  'field_relation _id',
  fieldname')

El significado de las partes especificadas en las sintaxis es:

- *other.object.name*: es el otro objeto que forma parte de la relación y es común a todos los tipos.
- *relation_object*: constituye el vínculo.
- *actual_object_id* y *other_object_id*: representan los nombres de los campos que son usados en la tabla de relación.

Para facilitar la legibilidad se ha establecido por convención que los nombres de los campos *many2one* finalicen con *_id* y los nombres de los campos *one2many* y *many2many* en *_ids.* Así es más fácil identificarlos y distinguir a qué tipo de campo hace referencia cada uno.

3.2.4. Archivo <vista_modulo>.xml

Este archivo es el que contiene la declaración de las distintas vistas que se manifestarán. Esto implica definir cómo y dónde aparecerá cada campo diseñado en el objeto. Como ya se ha mencionado, existen diferentes clases de vistas, donde las más importantes son: árbol, formulario, lista, calendario, gráfico y diagrama de Gantt.

Para aplicar las vistas es necesario crear previamente los menús y las acciones asociadas a los mismos. Para ello hay que incluir el nombre del fichero *'menu.xml'* en la lista *update_xml* del fichero *__openerp__.py* del módulo.

Los operadores para la creación de un menú son:

- *menuitem_id*: precisa el identificador del menú en la tabla interna de OpenERP donde están almacenados los distintos menús. Este identificador es un campo obligatorio y tiene que ser exclusivo para cada menú.
- *name*: asigna un nombre para el menú dentro del conjunto de menús. A diferencia del anterior, este nombramiento es opcional, ya que si no se declara, se le asignará automáticamente el nombre de la acción asociada.
- *action*: indica el identificador de acción que tiene que estar definida en la tabla de acción. Se pueden definir elementos de un menú sin asignarles una acción. No es un campo obligatorio.
- *icon*: determina el icono o imagen que se verá en el menú.
- *parent*: declara el identificador de menú padre del que va a depender el menú actual. Por defecto, si no se modifica, dependerá del menú principal o raíz.
- *groups*: restringe qué usuarios tienen posibilidad de ver y abrir el menú. Por ejemplo, si el menú está destinado para el uso del administrador, se especifica así: *groups="admin"*. Si estuviera dirigido a varios grupos sería, por ejemplo: *groups="admin, user".*
- *sequence*: ordena el menú en la jerarquía de menús. Es un número entero que determina su posición; cuanto mayor sea el número, más abajo estará en el árbol de menús. Definirlo es opcional.

Las acciones son las que determinan la manera de proceder del sistema ante los eventos generados por el usuario. Entre las más importantes destacan:

- *window*: abre una nueva ventana o pestaña. Denominada en el sistema *ir.actions.act_window*, es la más común. Permite revelar distintas clases de vistas. La otra opción es que sean etiquetas con el atributo *model="ir.ui.view"*, la diferencia es que estas contienen las

definiciones de las vistas ellas mismas y las primeras relacionan acciones con las vistas.

- *report*: muestra un informe.
- *wizard*: inicia un asistente para efectuar un trabajo o comenzar un proceso.

Como se ha dicho, las más abundantes son las acciones *window* o de ventana. Por ello, requieren un estudio en mayor profundidad. Para elaborar una ejecución de apertura de una ventana se puede seguir el siguiente código general:

```
<record model="ir.actions.act_window" id="action_id_1">
    <field name="name">action.name</field>
    <field name="view_id" ref="view_id_1"/>
<field name="domain"> ["list of 3-tuples (max 250 characters)"] </field>
<field name="context"> {"context dictionary (max 250 characters)"}
</field>
    <field name="res_model">Open.object</field>
    <field name="view_type">form|tree</field>
    <field name="view_mode">form,tree|tree,form|form|tree</field>
</record>
```

El significado de los términos que aparecen en el código es:

- *model*: corresponde con el objeto creado, en el caso de la acción ventana es "*ir.actions.act_window".*
- *id*: es el identificador de la acción. Es distintivo de cada una.
- *name*: es el nombre de la acción. Debe aparecer obligatoriamente.
- *res_model*: señala el objeto sobre el que se aplica la acción.
- *view_type*: distingue entre los dos tipos de vista: *"form"* para vista de tipo formulario y *"tree"* para vista tipo árbol, también denominada tipo lista.
- *view_mode*: solamente se aplica si el tipo de vista es *"form".* Hay cuatro posibilidades:
    - *form,tree*: por defecto la vista que aparece es de tipo formulario, pero se puede cambiar a modo lista en el botón "lista".
    - *tree,form*: por defecto la vista que aparece es de tipo árbol, pero se puede cambiar a modo formulario en el botón "formulario" o dando doble clic sobre un componente de la lista.
    - *form*: la vista es de tipo formulario sin posibilidad de cambiarla.
    - *tree*: la vista es de tipo árbol sin posibilidad de cambiarla.

3.2.4.1. Vista tipo formulario

El tipo de vista formulario permite, a su vez, varios modos: formulario, calendario, gráfico, etc. El código general de XML para este tipo de vista es el que se muestra a continuación:

```
<record model="ir.ui.view" id="view_id">
        <field name="name">view.name</field>
        <field name="model">object_name</field>
        <field name="type">form</field> <!--tree, form, calendar, search,
graph, gantt-->
        <field name="priority" eval="16"/>
        <field name="arch" type="xml">
                <!-- view content: <form>, <tree>, <graph>, ... -->
        </field>
</record>
```

Donde el significado de los términos es el siguiente:

- *id*: es el identificador de la vista.
- *name*: es el nombre de la vista.
- *model*: señala el objeto sobre el que se aplica la vista.
- *type*: tipo de vista: *form, tree, graph, calendar, search* y *gantt.*
- *priority*: establece prioridades entre las vistas mediante un entero, cuanto más grande, menor prioridad.
- *arch*: define la estructura o arquitectura de la vista. Esto se realiza a través de etiquetas XML.

Las vistas de formulario reparten los campos en la ventana siguiendo varias pautas:

- A cada campo le antecede su nombre.
- Los campos se ponen de izquierda a derecha, en el orden en el que se declararon en el archivo XML.
- La pantalla se divide en cuatro columnas. En cada una y por cada fila puede haber una etiqueta de campo o un campo disponible para editar. Como a cada campo le precede su nombre, en cada fila queda espacio para dos campos, ya que las otras dos son ocupadas por las respectivas etiquetas.

Existe la posibilidad de que un campo se muestre en varias columnas a través del atributo *colspan*. También se pueden colocar los campos en distintas pestañas con *notebook* y *page*, como se verá más adelante.

A este tipo de vistas se les puede incluir una serie de elementos:

- *field*: campos adicionales que se pueden añadir. También pueden venir acompañados de un *widget* específico. Puede ser, por ejemplo, un calendario que permita seleccionar el valor de un campo de tipo fecha más fácilmente. Tiene algunos atributos específicos:
    - *string*: es la etiqueta del campo. En las búsquedas remplaza el nombre del campo.
    - *nolabel*: se puede quitar la etiqueta del campo poniendo valor 1 a este elemento.
    - *required*: si su valor es 1, el campo será requerido. El campo aparecerá con fondo azul.
    - *read_only_mode*: cuando su valor es 1, se trata de un campo de sólo lectura.
    - *password*: si se quieren velar los caracteres escritos en el campo, se tiene que establecer en *True*. Se utiliza para las contraseñas.
    - *context*: sirve para declarar un diccionario contextual. Está especificado en código Python.
    - *domain*: sirve para declarar una lista de tuplas. Estas tuplas tienen condiciones que serán utilizadas para restringir valores. Esto se usa en los campos relacionales.
    - *on_change*: cuando se modifica el contenido de un campo se ejecuta este método de Python que calcula de forma automática el valor de los campos relacionados.
    - *groups*: es una enumeración de los usuarios que tienen autorización a ver el campo.
    - *widget*: para alternar el *widget* del campo de forma manual.
- *properties*: es un *widget* dinámico que muestra las diferentes propiedades que se encuentran disponibles. Sirve para elegir entre varios valores según la compañía del usuario.
- *button*: es un *widget* en forma de botón que realiza distintas acciones. Estos son sus atributos particulares:
    - *type*: determina la clase de botón: *"workflow"*, *"object"* o *"action"*.
    - *name*: es la señal del flujo de trabajo, la acción a realizar o el nombre de la función.
    - *confirm*: hace aparecer un mensaje de confirmación con el texto aquí introducido.
    - *states*: es una lista de estados.
    - *icon*: muestra un icono. No es obligatorio.
- *separator*: introduce un espacio en toda una línea horizontal. Sirve para estructurar las vistas. Puede llevar una etiqueta.

- *newline*: inserta una nueva línea.
- *label*: muestra un título con libertad en el texto o para las leyendas del formulario.
- *group*: con este comando se puede agrupar campos bajo una misma etiqueta. Se mostrará con un marco alrededor de dicho grupo.
- *notebook, page*: los elementos *notebook* contienen las pestañas para los elementos *page* y éstos especifican cada una de las pestañas. Tienen estos atributos:
    - *name*: etiqueta de la pestaña.
    - *position*: lugar en el que están ordenadas las pestañas en el *notebook*. Puede ser: *inside*, *bottom*, *top*, *right* o *left*

Es de especial interés el tipo de vista calendario. Es muy práctica, ya que permite mostrar resumidos, y en la posición conveniente, los distintos eventos trascendentes guardados por el usuario.

Los atributos con los que funciona el calendario son los siguientes:

- *string*: es la etiqueta con la que se identifica cada campo.
- *date_delay*: es un campo de tipo numérico que especifica el tiempo en horas que dura el registro o evento. Tiene prioridad sobre los dos siguientes.
- *date_start*: es la fecha en la que comienza un determinado evento. Es un campo *datetime*.
- *date_stop*: es la fecha en la que está planeado el fin de un determinado evento. Es un campo *datetime*. No se tiene en cuenta si el atributo *date_delay* está especificado.
- *day_length*: es un campo para introducir un entero que representa la duración de la jornada laboral. Por defecto es 8 horas.
- *color*: es un campo que sirve para señalar con determinado color en el calendario. Normalmente es un campo *many2one*.

El resultado de la elaboración de una vista de formulario se puede observar en la figura 3.3:

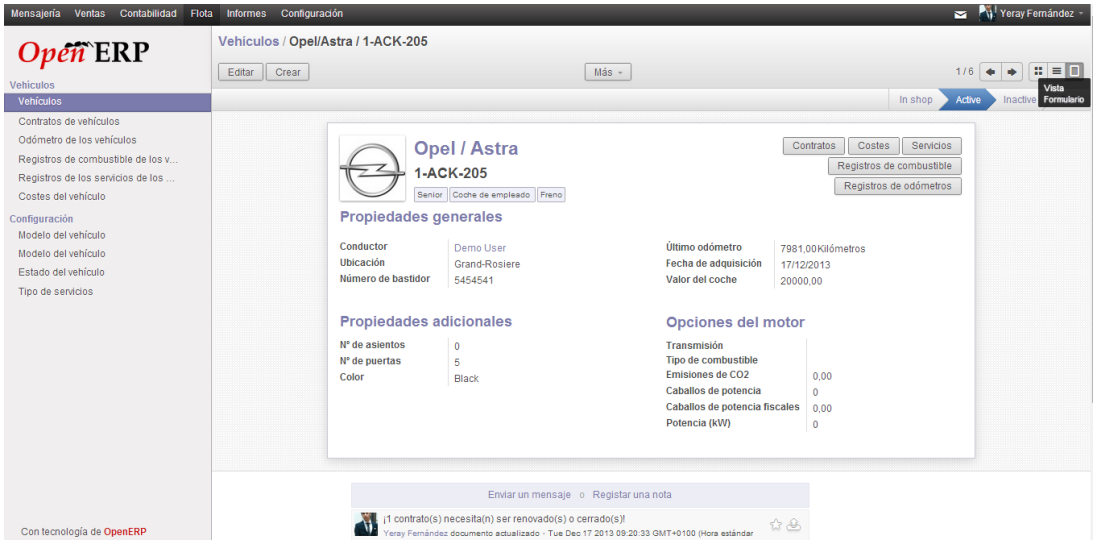


Figura 3.3: *Ejemplo de una vista tipo formulario de un vehículo existente.*

3.2.4.2. Vista tipo árbol

Las vistas tipo árbol suelen ser listas, adaptadas para facilitar las búsquedas. Son más simples que las vistas tipo formulario, ya que tienen menos opciones, y tienen un elemento padre, que es de tipo *tree*.

Los principales atributos que presentan son:

- *colors*: son los colores establecidos en una lista en el diccionario Python. Sirven para establecer diferentes colores en las líneas de la vista árbol.
- *editable*: permite la edición inmediata de la lista. Se puede establecer a *top* para añadir nuevos registros al principio de la lista o *bottom* para hacerlo al final.
- *toolbar*: si se establece a *True* se puede utilizar la categoría superior de la jerarquía de objetos a modo de barra de herramientas. Esto es lo que se utiliza en los menús.

Los elementos que puede incluir son: *field*, *group*, *separator*, *tree*, *button*, *filter* y *newline*.

Se puede ver un ejemplo de una vista de modo lista en la figura 3.4:

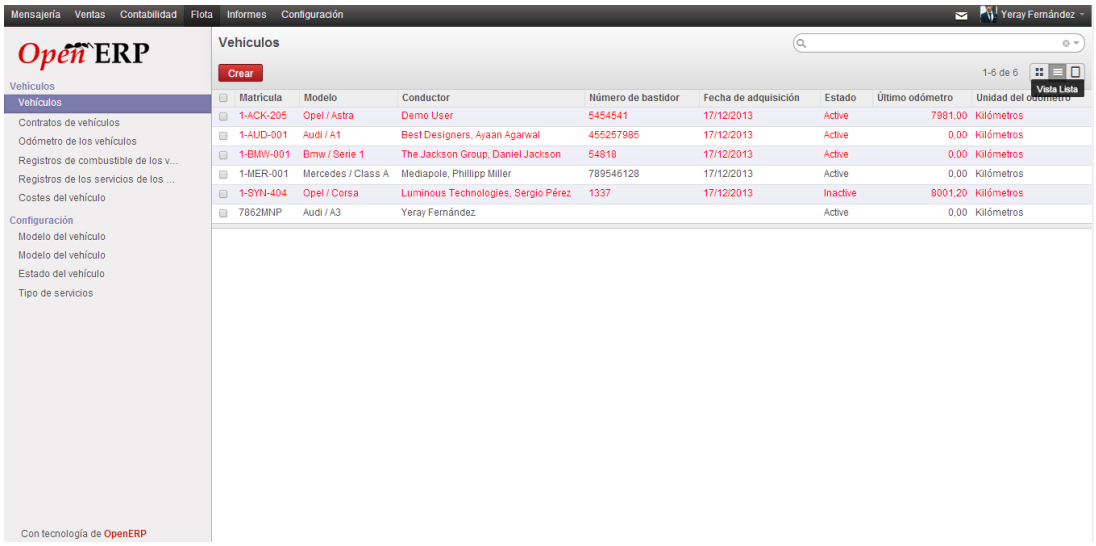


Figura 3.4: *Vista tipo lista de la flota de vehículos.*

3.2.4.3. Vista kanban

Una de la novedades de la versión 7 de OpenERP es la inclusión de un nuevo tipo de vista, las vistas tipo kanban. Esta nueva opción permite una visión globalizada del estado de documentos y procesos de negocio. Se declaran de una manera similar a las dos anteriores vistas y el resultado se puede comprobar en la figura 3.5:

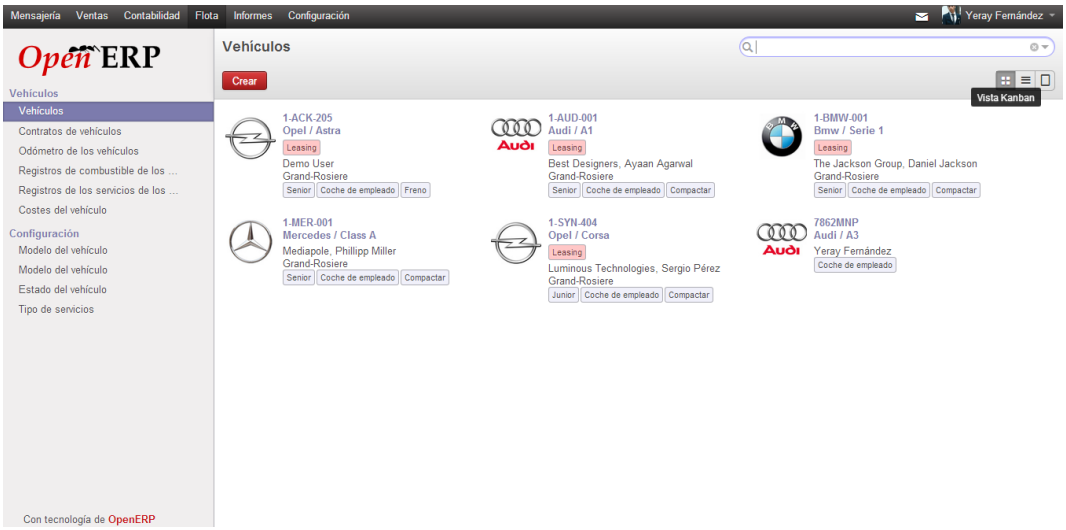


Figura 3.5: *Vista tipo kanban de la flota de vehículos.*

CAPÍTULO IV

PERSONALIZACIÓN DE UN MÓDULO
PARTE I

Una vez explicado el funcionamiento y composición de los módulos en OpenERP, en este capítulo se elaborará una guía para afrontar la modificación de un módulo dedicado a la gestión de una flota de vehículos. Para ello, partiremos del módulo preinstalado *fleet management,* que se puede descargar de la tienda de aplicaciones: reutilizar un módulo ya creado permite optimizar tiempo y dinero al no tener que partir de cero en su elaboración.

El objetivo del capítulo es sentar las bases para que cualquier pequeña o mediana empresa pueda implementar los cambios que desee en su sistema, en aquellos módulos que sea necesario. Esto es posible a través de la modificación del código Python y XML que compone los archivos del módulo aunque, posteriormente, habrá que aplicar esos cambios a la versión web del ERP.

Existen dos maneras de llevar a cabo la personalización: mediante programación o a través de la misma interfaz de OpenERP. En los puntos siguientes se compararán ambos métodos y se discutirá de forma razonable cuál es la elección más adecuada desde el punto de vista del alcance del proyecto.

4.1. Métodos de modificación

Como se ha comentado, existen dos maneras de modificar un módulo en OpenERP: a través de la programación de los archivos mediante código Python y XML, y a través de la interfaz web. Cada una de ellas presenta una serie de características, ventajas e inconvenientes que analizaremos a continuación.

4.1.1. Programación de los archivos mediante código Python y XML

Este método consiste en la alteración del código fuente de los archivos correspondientes al módulo o a los módulos que se quieran cambiar. En el capítulo anterior ya se expusieron los distintos archivos de los módulos de OpenERP y para qué servía cada uno. También se presentaron los componentes de los distintos archivos y qué función tenían en el software.

Para efectuar cambios en el programa basta con realizar las modificaciones pertinentes en el código del módulo para cargar después la actualización vía web. Se eliminan, añaden y/o modifican los elementos

existentes en el archivo correspondiente del módulo y se realizan los cambios pertinentes en los archivos XML para que quede reflejado en las vistas. Estas alteraciones se realizan manualmente, como por ejemplo, con la herramienta ya mencionada en capítulos anteriores IDLE Python (Figura 2.2).

El software reconoce los archivos alterados y aparecen en el apartado "Actualizar lista de módulos", que está en la sección "Módulos" del menú "Configuración". Para cargar la actualización hay que hacer clic en "Actualizar".

A través del código se pueden añadir nuevas variables, nuevos menús, nuevos campos, etc. Todos esos cambios deben ser consistentes y ser declarados por igual en los distintos archivos para que el programa los reconozca y se puedan implementar en la solución final.

Entre las ventajas de realizar el cambio mediante la programación se encuentran las siguientes:

- Las posibilidades de modificación son enormes. Se pueden añadir los elementos que se consideren necesarios, declarar su tipo y función de una sola vez.
- Se pueden trasformar los campos bases, que de otra manera permanecen inmutables.
- Se da la posibilidad de aplicar numerosos cambios al programa simultáneamente.
- Ofrece un alto grado de versatilidad.

Sin embargo, también presenta varios inconvenientes o desventajas:

- Es más lento para descubrir errores o disconformidades.
- La ejecución es más compleja.
- Requiere de los conocimientos técnicos y experiencia en el lenguaje de programación Python.
- Es fácil confundirse entre todos los archivos existentes.
- Es poco intuitivo.
- Requiere del uso de otros programas ajenos para la lectura y cambio del código, aunque es cierto que la herramienta se puede descargar conjuntamente con el mismo ERP.
- La carga de los archivos al software vía web presenta bastantes problemas, que en muchos casos son difíciles de solucionar o directamente no se pueden resolver.

4.1.2. A través de la interfaz

La versión de OpenERP con la que se ha trabajado en este Trabajo Fin de Máster es la 7.0. En ella se han realizado numerosos cambios respecto a las versiones anteriores: en concreto, uno de los más trascendentales y de especial importancia para este trabajo es que se ha añadido la posibilidad de realizar cambios en el software a través de unas opciones ofrecidas en sus menús; en otras palabras, con esta versión de OpenERP se pueden realizar modificaciones desde la misma interfaz del programa, sin necesidad de recurrir a herramientas externas.

El proceso consiste básicamente en definir nuevas variables o campos y coordinarlo con las vistas para que se puedan utilizar de manera funcional. De la misma manera se pueden modificar los menús y las acciones ligadas a estos.

Las pros que presenta esta forma de modificación son:

- Es un método intuitivo y rápido.
- No es necesario salir del ERP.
- No da errores de compilación.
- Se puede ir observando el resultado mientras se van realizando los cambios.
- Los campos se ven de manera compacta, de forma que en una misma ventana se pueden observar sus propiedades y modificarlas, asignar grupos, añadir menús,… Con la modificación mediante código todo esto es mucho más complejo.

Por el contrario, presenta los siguientes inconvenientes:

- Hay que coordinar la coherencia con las vistas para evitar errores y fallos.
- No se pueden cambiar los campos base. En este sentido, la personalización es menos profunda, pero se compensa con la calidad, completitud y adecuación de estos campos que vienen de origen. Son campos fundamentales de los que rara vez se prescindirá de ellos y, además, no supone un problema dejarlos.

4.2. Elección del método

Con lo visto hasta ahora, se está en la disposición de seleccionar cuál es el método más conveniente sin perder de vista el objetivo del presente trabajo.

Se comenzó intentando modificar el módulo desde el código Python y los archivos XML. Tras varias pruebas e intentos, en la actualización del ERP con el nuevo módulo, es decir, en la carga de los archivos modificados a la web, el sistema no fue capaz de integrar el nuevo módulo y se produjeron múltiples errores relacionados con los permisos.

A modo de prueba se creó un módulo desde cero, para comprobar que los errores mostrados no se trataban de fallo del código. El resultado acabó por ser el mismo.

El procedimiento que se siguió fue el siguiente: añadir la carpeta con los archivos del nuevo módulo a la carpeta donde se encuentran los demás (carpeta *addons*), después se abre la cuenta administrador con las características técnicas activadas, luego hay que entrar en "Configuración" "Actualizar lista de módulos" y "Actualizar". Posteriormente en la barra de búsqueda se introduce el nombre del nuevo módulo y se hace clic en "*Install*" para tener disponible dicho módulo. Es en ese momento es cuando sale el error comentado más arriba.

El módulo de prueba se trataba de un ejemplo sencillo en el que se creaba una nueva entrada en el menú general y en la base de datos una tabla con el nombre, la edad y la fecha de nacimiento de los empleados. Los archivos de código generados se pueden observar en las Figuras 4.1, 4.2, 4.3 y 4.4.

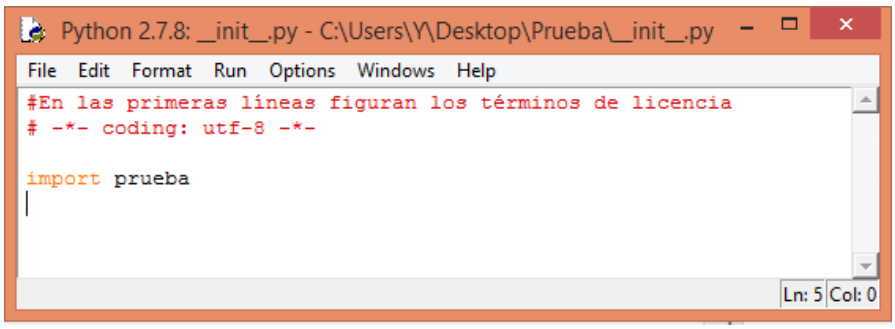


Figura 4.1: *Archivo __init__.py del módulo de prueba.*

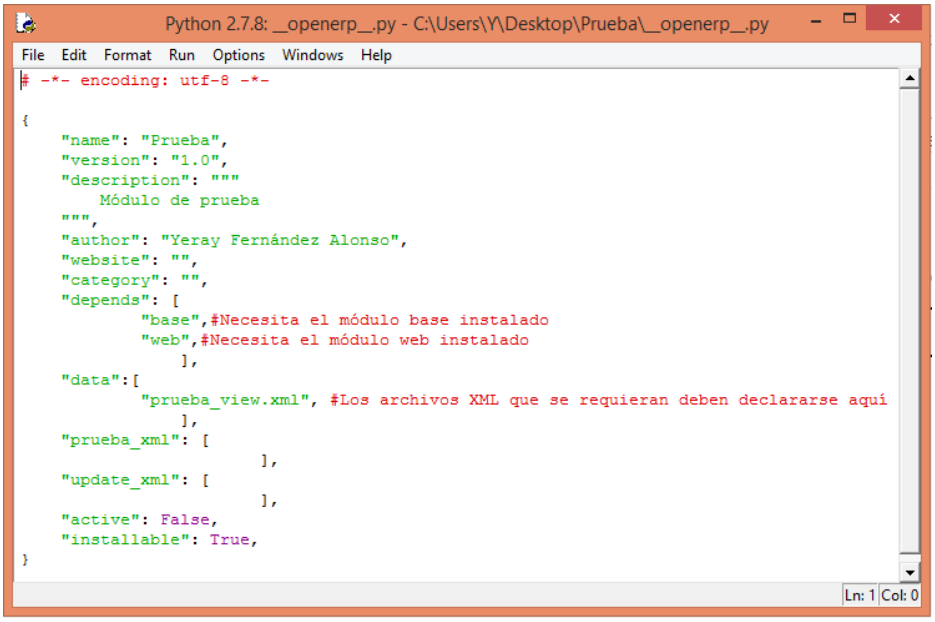


Figura 4.2: Archivo __openerp__.py del módulo de prueba.

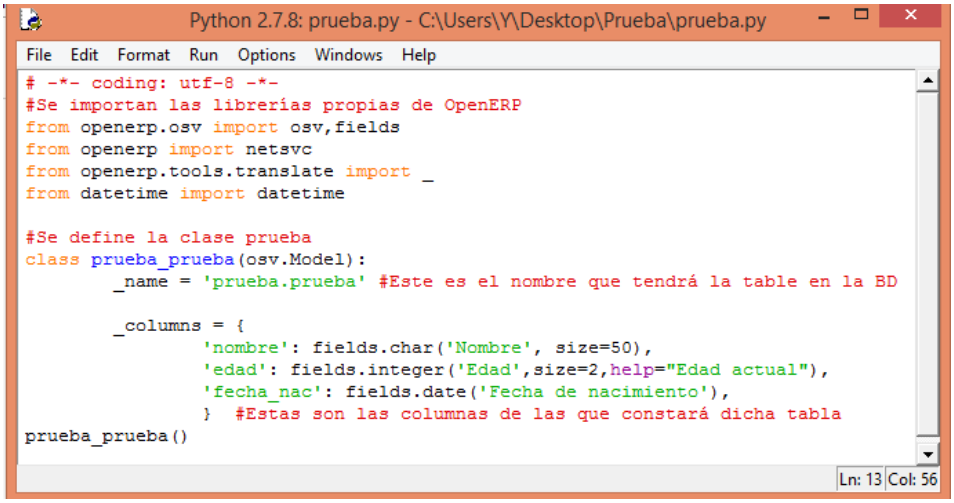


Figura 4.3: Archivo prueba.py del módulo de prueba.

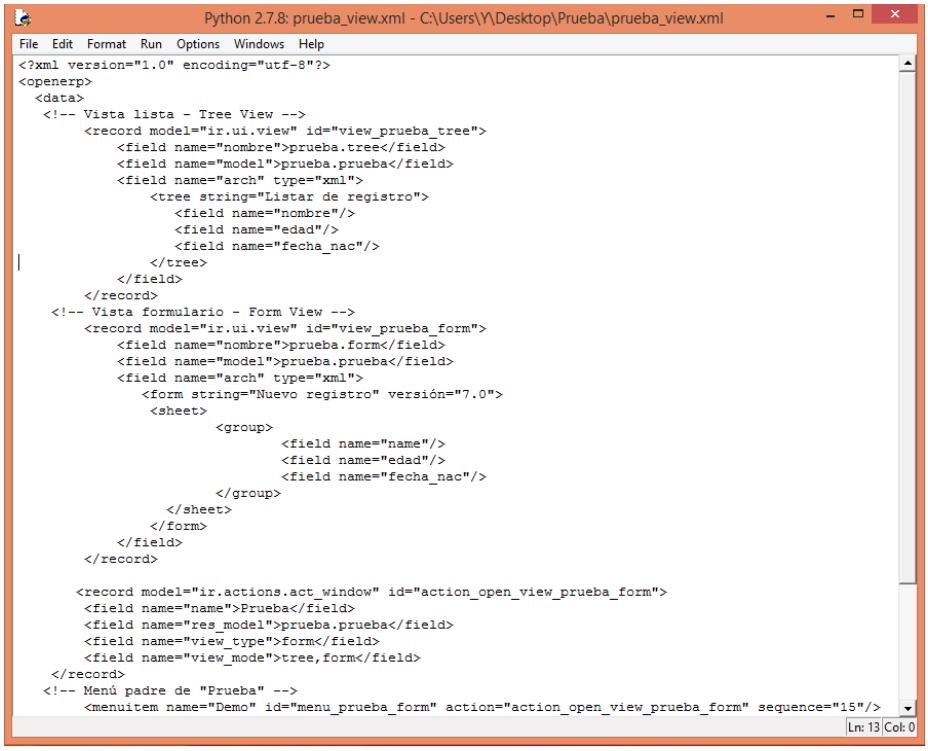


Figura 4.4: *Archivo prueba_view.xml del módulo de prueba.*

Tras múltiples búsquedas en internet, las soluciones que se encontraron sólo eran válidas para para entornos Linux, y más concretamente Ubuntu. Sin embargo, en la actualidad el sistema operativo predominante en las empresas Españolas (aunque también a nivel mundial) sigue siendo Windows, por lo que esta forma de proceder mediante el código puede traer más problemas que soluciones y no se recomienda en entornos Windows. Cabe decir que este trabajo se realiza con el sistema operativo Windows 8.1.

Se puede concluir que la versión 7.0 de OpenERP, frente a lo que ocurría en las 6.X, está orientada para trabajar de manera continua desde la interfaz, incluyendo la modificación de los módulos obtenidos de la tienda de aplicaciones. La posibilidad de modificar el código se deja para aspectos muy concretos que no se pueden realizar de otra forma, aunque en contadas ocasiones sí es posible utilizarlo.

Teniendo presente que este proyecto se dirige hacia pequeñas o medianas empresas que no tienen por qué tener mucha capacidad técnica, pero sí las mismas necesidades de personalización de sus herramientas como cualquier otra, el método finalmente seleccionado es el de modificación de módulo desde la interfaz de OpenERP.

A partir de este momento se elaborará una guía para introducir los cambios mediante este método y se dará una visión general del módulo del que se parte y qué elementos fundamentales se pueden incorporar.

4.3. Adición de campos

Como se ha justificado en el apartado anterior, a lo largo de este trabajo se llevará a cabo la modificación del módulo mediante la interfaz de usuario web. Para comenzar a trabajar en la modificación del módulo, es necesario activar el modo desarrollador. Esta opción se encuentra al hacer clic en "Acerca de OpenERP", donde aparecerá la ventana que se muestra en la Figura 4.5:

Figura 4.5: *Ventana "Acerca de OpenERP".*

Tras activar este modo desarrollador, en cada pantalla de OpenERP aparecerá un menú desplegable en la esquina superior, a la izquierda del nombre de la sección, tal y como se puede apreciar en la Figura 4.6.

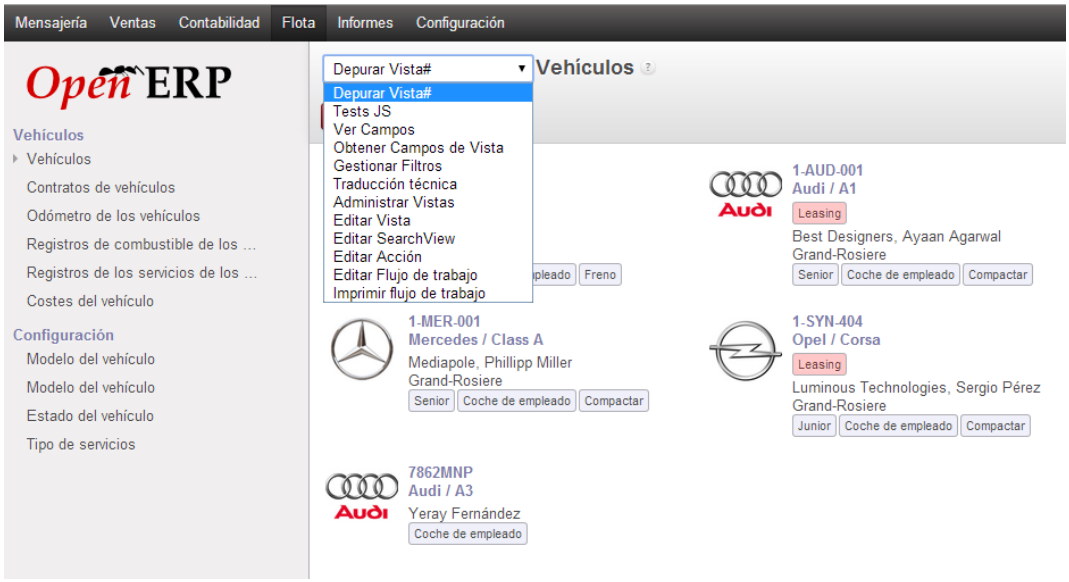


Figura 4.6: *Menú desplegable en el apartado "Vehículos".*

Lo más importante que se puede llevar a cabo desde él es la *edición de las vistas*, donde se puede añadir y quitar campos, introducir nuevos tipos de vistas, etc. Más adelante se explicará qué hay que hacer en este menú. Además, la cuenta de usuario desde la que se vaya a editar el software tiene que tener habilitadas las características técnicas que permitan realizar cambios en el programa.

Para añadir nuevos campos hay que ir a "Configuración" dentro del menú general en la barra superior. Una vez ahí, en la sección "Técnico" hay que pinchar en "Estructura de la base de datos". En esta parte, hay que introducirse en el submenú "Modelos", donde se encuentra una lista con todos los modelos que integran los módulos instalados. Lo más cómodo es utilizar la precisa barra de búsqueda en la esquina superior derecha y teclear el apartado al que se quiere tener acceso (los nombres están en inglés y así hay que buscarlos).

En este caso se buscan los modelos del módulo de la flota, como se ve en la Figura 4.7 (se busca la palabra *fleet*). Ahí se hallan los modelos relativos al estado de los contratos, los servicios disponibles en los vehículos, la información de los propios vehículos, su coste, la información de sus contratos, los registros del combustible, los servicios para los vehículos, modelo y marca, registro del odómetro, estado del vehículo y etiqueta del vehículo.

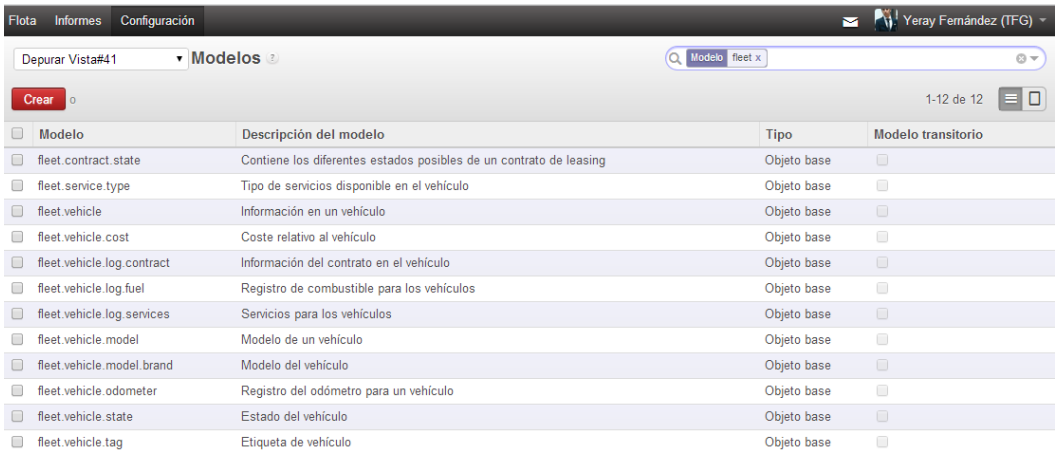


Figura 4.7: *Búsqueda de modelos con la palabra fleet.*

Se pueden añadir más modelos mediante el botón rojo “Crear”. Simplemente hay que rellenar el formulario que aparece en la Figura 4.8, completando los campos obligatorios (los que tienen el fondo azul) y opcionalmente los demás. Cuando se haya finalizado se da a “Guardar” para conservar los cambios. Los nuevos modelos creados se irán añadiendo a la lista de la Figura anterior.

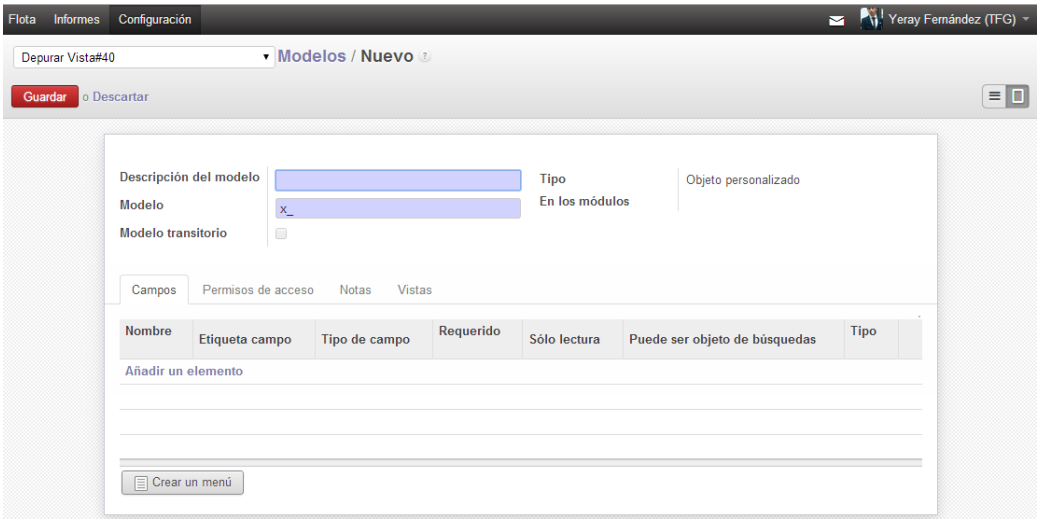


Figura 4.8: *Plantilla para la creación de un modelo.*

También se puede crear un menú para dicho modelo en el botón “Crear un menú”: sólo hay que darle un nombre e indicar la dependencia de menús (Figura 4.9).

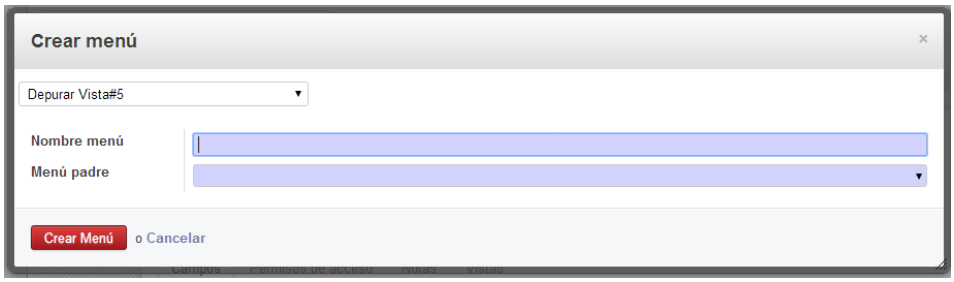


Figura 4.9: *Creación de un menú para un nuevo modelo.*

Tanto en los modelos creados por el usuario como en los que vienen con el módulo se pueden añadir elementos, es decir, nuevos campos para completar la información contenida en ellos.

Para ello, con el modo desarrollador activado y desde una cuenta con las características técnicas habilitadas, hay que introducirse en un modelo y hacer clic en el botón “Editar” en la esquina superior izquierda (Figura 4.10).

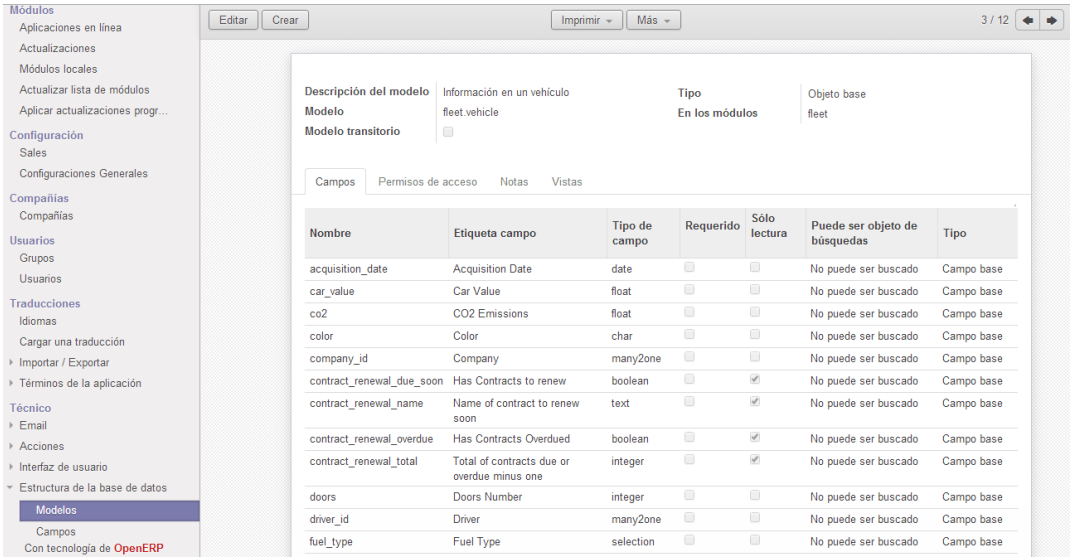


Figura 4.10: *Lista de campos en el modelo fleet.vehicle.*

Una vez que se haya dado a editar, justo debajo de la última fila de la tabla de campos del modelo aparecerá la opción “Añadir un elemento”. Seleccionándola aparecerá la ventana de la Figura 4.11.

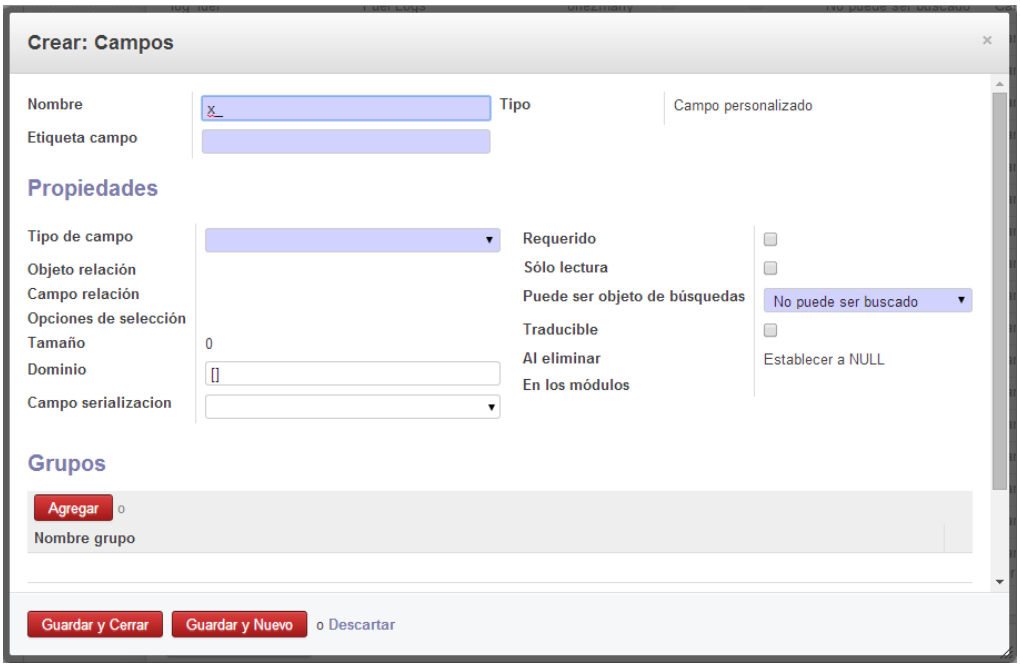


Figura 4.11: *Creación de un nuevo campo.*

Como en otros apartados, los campos que aparecen en azul son los obligatorios. Hay que definir el nombre con el que se va a reconocer el nuevo campo en la edición del programa y su etiqueta, que es el identificador que aparecerá junto a él en las vistas. En cuanto al “Tipo de campo” hay que seleccionar una de las opciones del menú desplegable indicado en la Figura 4.12, los tipos de variables más importantes están ya comentadas en el capítulo anterior.

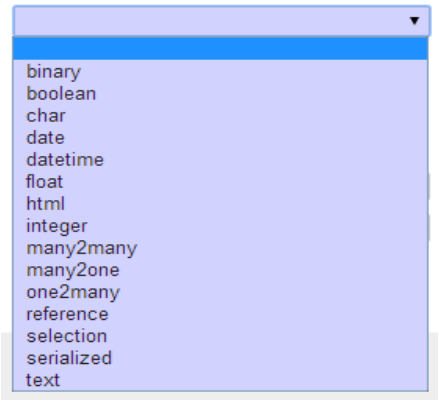


Figura 4.12: *Opciones de elección de "Tipo de campo".*

Los valores de "Objeto relación", "Campo relación", "Opciones de selección" y "Tamaño" dependen del tipo de campo en cuestión, solamente para algunos de los tipos será necesario definir alguno de esos valores.

Los apartados opcionales son: "Dominio", "Campo de serialización" (para codificar y transmitir el objeto en formato XML), "Requerido" (para hacer el campo obligatorio), "Sólo lectura" (para que no se pueda modificar), "Puede ser objeto de búsqueda" (hay que elegir entre "No puede ser buscado", "Siempre puede ser buscado" y "Búsqueda avanzada") y "Traducible" (en caso de que no sea un campo numérico o binario).

Por último, se pueden seleccionar los grupos con los que el campo estará relacionado. Hay que dar a "Agregar" y luego seleccionar el grupo o grupos requeridos. Para concluir, se guardan los cambios: "Guardar y Cerrar" para salir o "Guardar y Nuevo" para crear un nuevo campo en el mismo modelo.

4.4. Adaptación de las vistas

Una vez se hayan creado los campos que se consideren oportunos, es necesario modificar las vistas en las distintas ventanas de los módulos, a fin de adaptarlos a los nuevos campos introducidos.

Para hacer esto, se necesita el menú que se habilitó con el modo desarrollador. Cuando se esté en la ventana del modelo del módulo donde se han añadido los campos, hay que seleccionar "Administrar vistas" y saldrá una ventana con los tipos de vistas disponibles para ese modelo, tal y como se puede apreciar en la Figura 4.13.

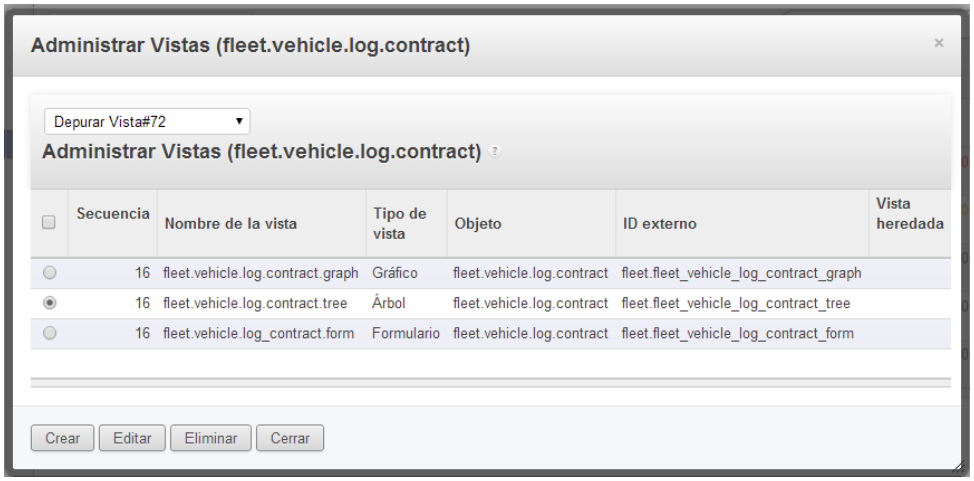


Figura 4.13: *Ventana desde la que se administran las vistas.*

Mediante el botón "Editar" se puede acceder al esquema de comandos XML que definen la vista seleccionada (Figura 4.14). Coincidiendo con la filosofía de esta versión de OpenERP, la 7.0, la edición de la vista desde aquí también se ha simplificado y se pueden añadir sin gran dificultad.

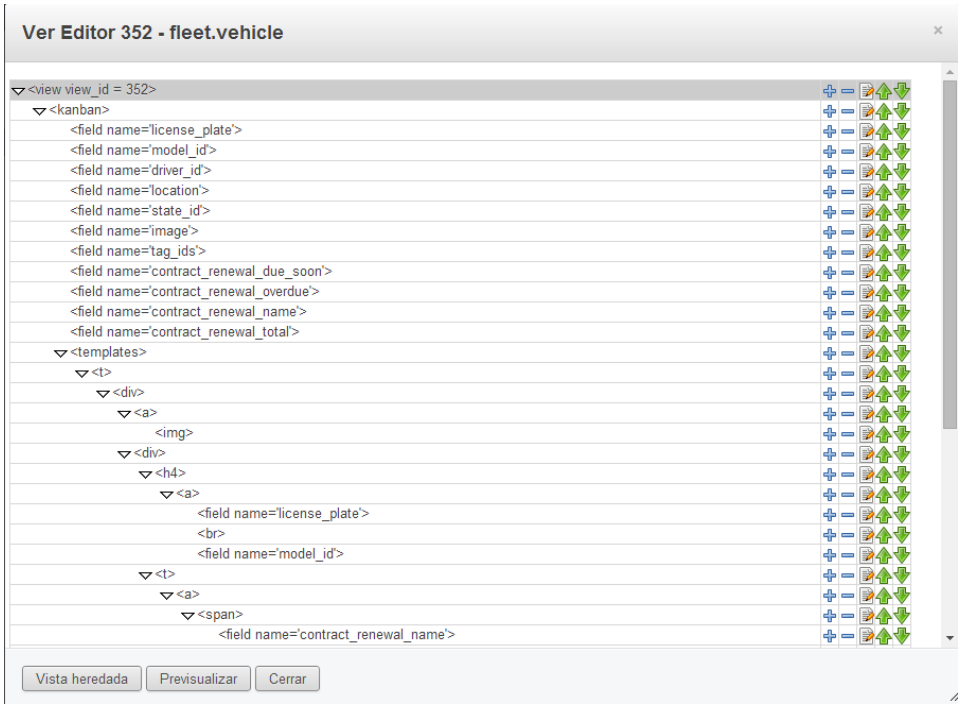


Figura 4.14: *Estructura del código XML.*

Las flechas verdes permiten subir o bajar la línea de posición, sirve para ordenar y estructurar el resultado final de la vista. Con el botón "+" se abre un cuadro de diálogo con el que añadir los campos previamente creados que se precisen. Con "-" se elimina esa línea y el campo al que se refiere ya no aparecerá en la ventana resultante. El botón central abre una ventana en la que se pueden modificar las propiedades del campo en cuestión (Figura 4.15).

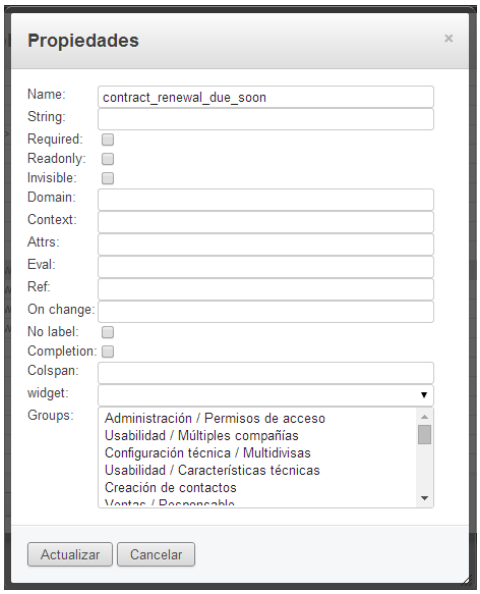


Figura 4.15: *Cuadro de propiedades del campo.*

Cuando se hayan cambiado los parámetros necesarios, se da a "Actualizar" para salir.

Además, se ofrece la posibilidad de realizar los cambios en el código XML de forma manual (Figura 4.16). En el mismo menú donde se administran las vistas existe una opción de editar únicamente la vista que en ese momento esté abierta. En esa misma ventana se pueden editar el campo hijo dependiente del actual y la vista heredada.

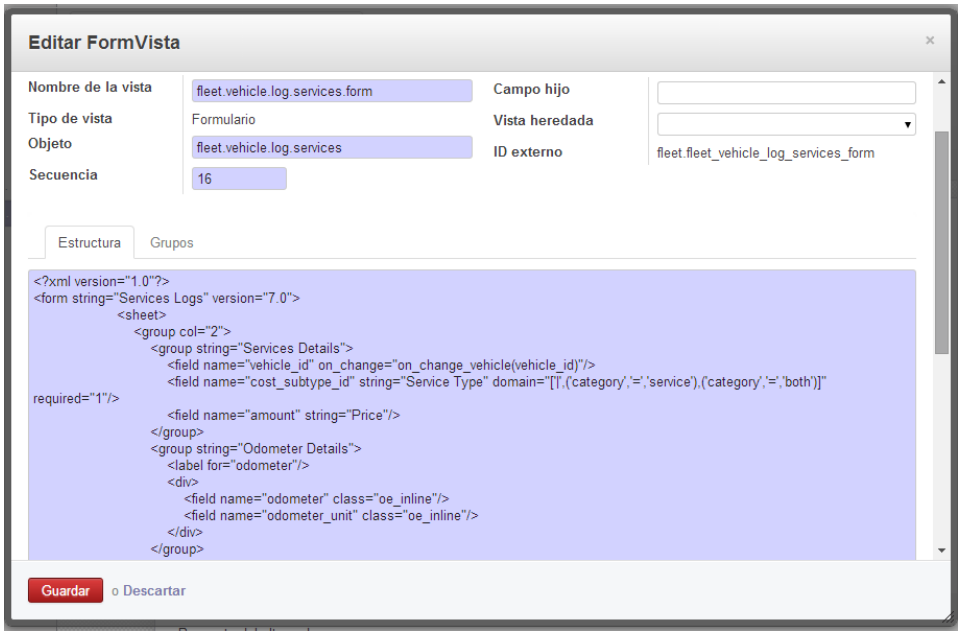


Figura 4.16: *Ventana de edición de la vista manualmente.*

Con cualquiera de estas dos opciones los campos y las vistas estarían sincronizados y se obtendría un módulo totalmente funcional.

A su vez, también se pueden ir editando los cuadros de diálogo en la creación de los objetos y menús. Para hacerlo, hay que editar las características de las ventanas mediante el menú que aparece al lado de los títulos cuando se activa el modo desarrollador. Esto puede ser útil si se necesita que el programa pida al usuario completar la información de un determinado campo para crear o editar algún elemento.

Una vez descrita la manera de proceder a la hora de editar un módulo, se está en condiciones de afrontar la modificación del módulo encargado de la gestión de la flota. Esto se realizará a lo largo del próximo capítulo.

CAPÍTULO V

PERSONALIZACIÓN DE UN MÓDULO
PARTE II

En el capítulo anterior se expuso la manera de proceder para editar, de forma general, los campos de un módulo. En este capítulo se van a aplicar dichos cambios a un módulo concreto, el *fleet management*, diseñado para gestionar una flota de vehículos.

Por este motivo, en primer lugar se va a analizar brevemente cómo es este módulo original de OpenERP y posteriormente se detallarán algunos de los cambios realizados, de forma que sirvan de ejemplo para realizar cualquier tipo de personalización.

5.1. Módulo original

El módulo que se va a modificar es el que ya se ha comentado en varias ocasiones, el de la flota de vehículos (*fleet management*). Se parte del módulo base descargado de la tienda de aplicaciones. Para instalarlo, hay que ir a "Configuración" y en el menú "Módulos" entrar en el apartado "Tienda de aplicaciones". Una vez ahí se busca la aplicación "*Fleet management*" y se selecciona "*Install*" (Figura 5.1). Cuando finalice la instalación, estará disponible dicha aplicación en el menú general.

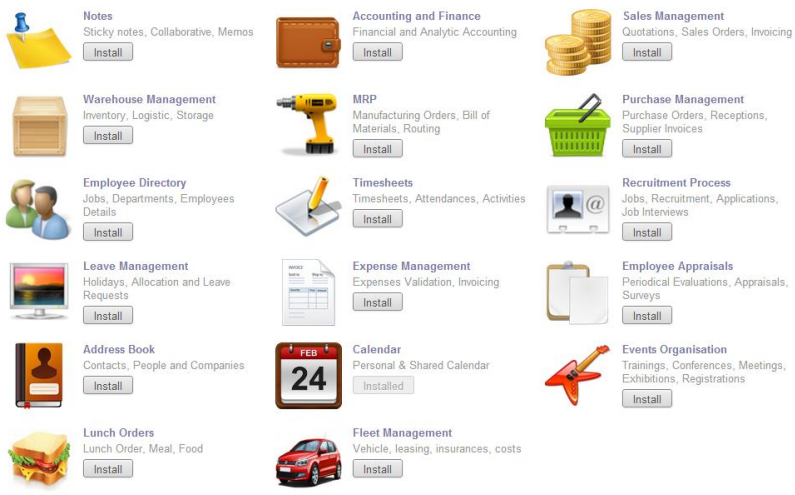


Figura 5.1: *Tienda de aplicaciones.*

El módulo de gestión de flota puede servir para empresas de reparto o logísticas, pero también para cualquier empresa que tenga vehículos de ámbito profesional y quiera gestionar y mantener integrada esa información con el resto de su sistema.

En la instalación de OpenERP se seleccionó la opción que permite incluir datos de demostración, por lo que al instalar este módulo aparecen ya unos vehículos con sus propiedades de ejemplo. Estos mismos datos sirven para ejemplificar el proceso a seguir para personalizar el módulo.

Los diferentes menús que integra se pueden observar en la Figura 5.2:

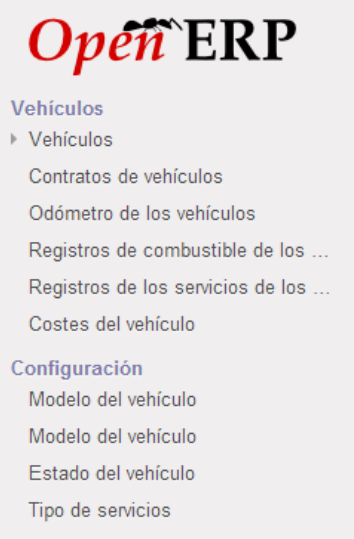


Figura 5.2: *Menú principal del módulo "Flota".*

A continuación se muestra con más detalle los aspectos principales de cada submenú del apartado Vehículos:

- Vehículos: incluye las vistas kanban, lista y formulario. En la primera de ellas, la vista kanban, se muestran los vehículos que están creados en la empresa en forma de pequeñas tarjetas para ver sus características generales. En la vista tipo lista se muestra una tabla en la que cada fila es uno de los vehículos y en las columnas están los campos Matrícula, Modelo, Conductor, Número de bastidor, Fecha de adquisición, Estado, Último odómetro y Unidad del odómetro. Por último, en la vista tipo formulario se encuentra la información almacenada de cada automóvil, es decir, las características asociadas a los vehículos. Además de las recogidas en otras vistas, se encuentran en el apartado "Propiedades generales", el valor del coche y la fecha de adquisición; en "Propiedades adicionales" se encuentran los campos correspondientes al número de asientos, número de puertas y el color; por último, en "Opciones de motor" aparecen la transmisión, el

tipo de combustible, las emisiones de CO2, los caballos de potencia, los caballos de potencia fiscales y la potencia (en kW).

- Contratos de vehículos: las vistas que vienen predefinidas son lista (Figura 5.3), formulario (Figura 5.4) y gráfico, que construye gráficos de distinto tipo y varias posibilidades de visualización.

Fecha de inicio del contrato	Fecha de expiración del contrato	Vehículo	Tipo	Proveedor	Coste de activación	Importe del coste recurrente	Frecuencia de los costes recurrentes	Estado
01/01/2013	16/12/2013	Opel/Astra / 1-ACK-205	Leasing	Axelor	0,00	150,00	Semanalmente	Para cerrar
01/01/2013	22/12/2013	Opel/Corsa / 1-SYN-404	Leasing	Axelor	0,00	20,00	Diariamente	Para cerrar
01/01/2013	31/12/2013	Bmw/Serie 1 / 1-BMW-001	Leasing	Axelor	0,00	400,00	Mensual	Para cerrar
01/01/2013	31/12/2013	Audi/A1 / 1-AUD-001	Leasing	Axelor	0,00	4000,00	Anualmente	Para cerrar
01/01/2012	31/12/2016	Mercedes/Class A / 1-MER-001	Leasing	Axelor	17000,00	0,00	No	En proceso

Figura 5.3: *Vista tipo lista de los contratos de vehículos.*

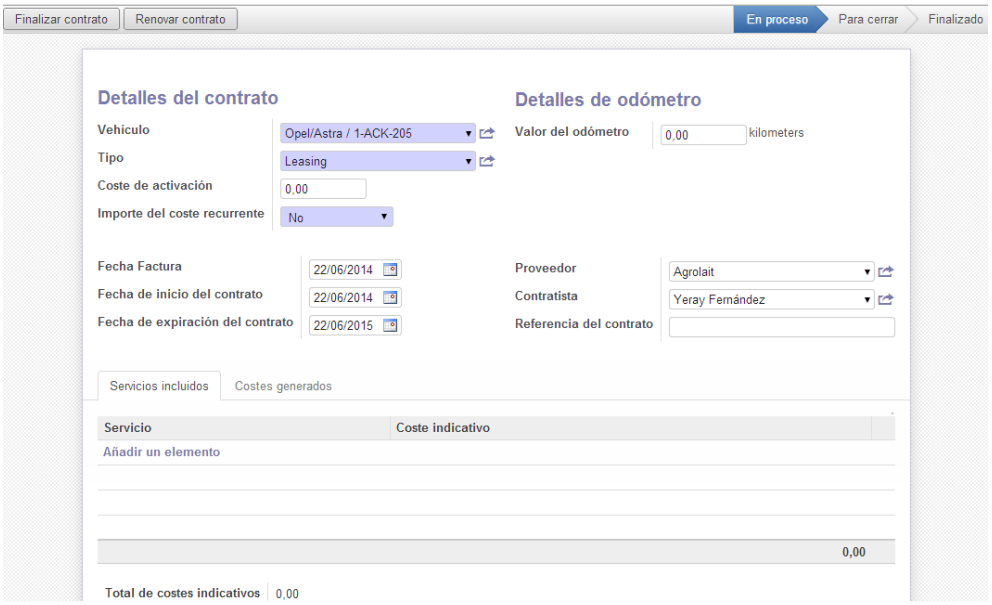


Figura 5.4: *Vista tipo formulario de los contratos de vehículos.*

- Odómetro de los vehículos: desagrega por fecha los kilómetros recorridos por cada vehículo. Viene con los tipos de vista lista y gráfico.

- Registros de combustible de los vehículos: detalla los gastos de combustible de cada vehículo. Las vistas que incluye originalmente son tipo lista, formulario (Figura 5.5) y gráfico (Figura 5.6).

Figura 5.5: *Vista tipo formulario del registro de combustible de un vehículo.*

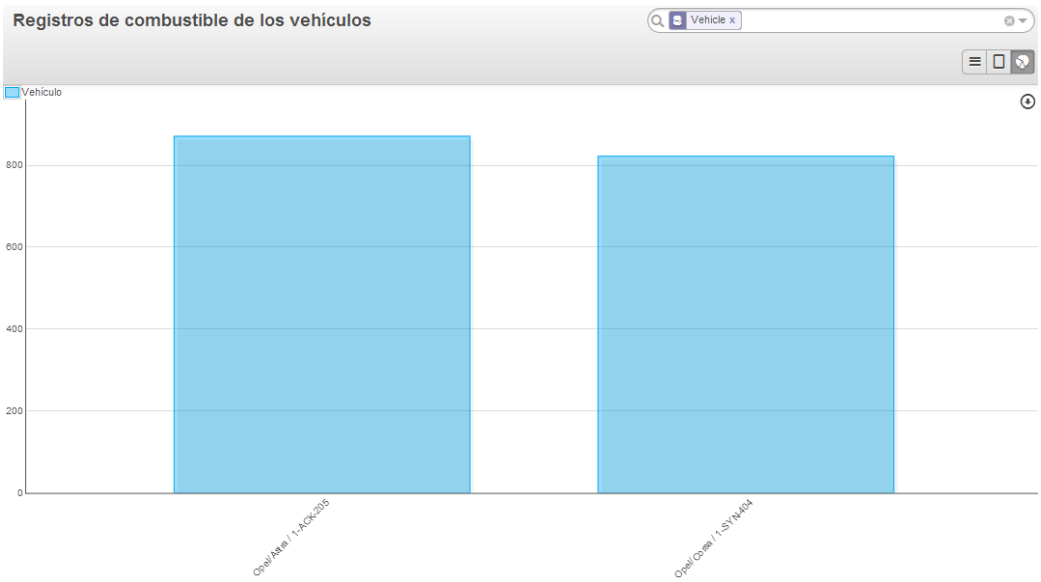


Figura 5.6: *Vista tipo gráfico de los registros de combustible de dos vehículos.*

- Registros de los servicios de los vehículos: sirve para realizar un seguimiento de los distintos servicios de los que son objeto los

vehículos. Se puede ver qué servicio recibe cada uno y otros datos de interés en la vista tipo lista (Figura 5.7). En la vista formulario se centra en un servicio determinado que recibe uno de los vehículos en concreto (Figura 5.8). Mediante un gráfico se pueden comparar entre coches y periodos (Figura 5.9).

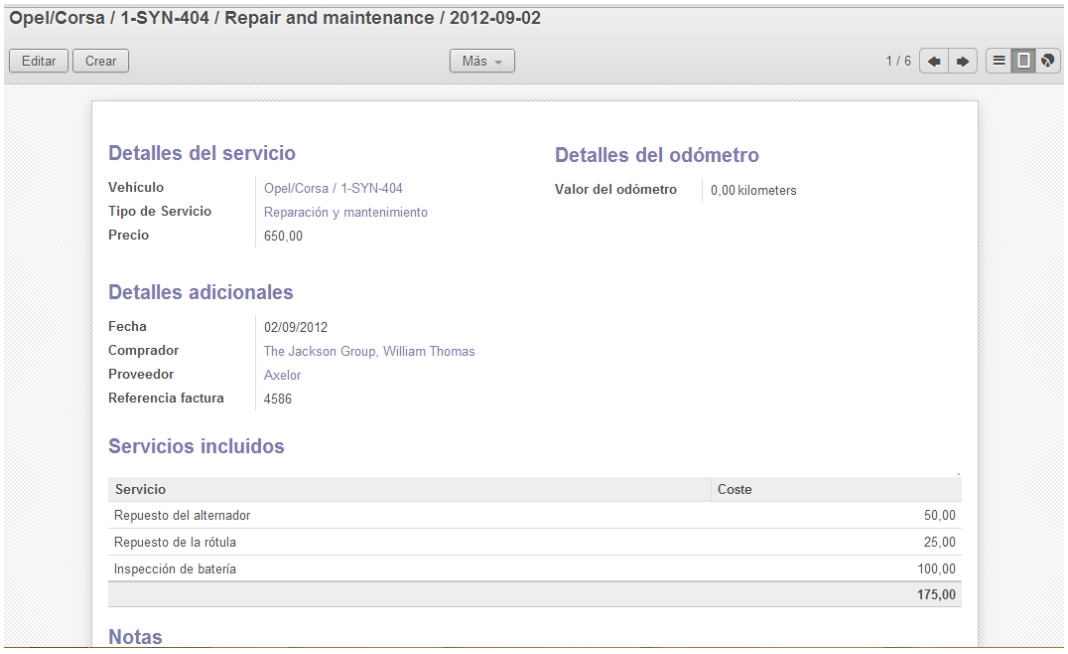
Registros de los servicios de los vehículos

Crear

1-6 de 6

Fecha	Vehículo	Tipo	Comprador	Proveedor	Referencia factura	Notas	Precio total
02/09/2012	Opel/Corsa / 1-SYN-404	Reparación y mantenimiento	The Jackson Group, William Thomas	Axelor	4586	Usual vehicle repairing	650,00
02/11/2012	Opel/Corsa / 1-SYN-404	Reparación y mantenimiento	The Jackson Group, William Thomas	Axelor	4814	After crash repairing	350,00
15/10/2012	Opel/Astra / 1-ACK-205	Reparación y mantenimiento	The Jackson Group, William Thomas	Axelor	124	Maintenance	513,00
08/10/2012	Bmw/Serie 1 / 1-BMW-001	Reparación y mantenimiento	The Jackson Group, William Thomas	Axelor	20984	Maintenance	412,00
25/09/2012	Audi/A1 / 1-AUD-001	Reparación y mantenimiento	The Jackson Group, William Thomas	Axelor	241	Maintenance	275,00
15/09/2012	Mercedes/Class A / 1-MER-001	Reparación y mantenimiento	The Jackson Group, William Thomas	Axelor	22513	Maintenance	302,00
							2502,00

Figura 5.7: *Vista tipo lista de los servicios de los vehículos.*



Opel/Corsa / 1-SYN-404 / Repair and maintenance / 2012-09-02

Editar Crear Más 1 / 6

Detalles del servicio

Vehículo	Opel/Corsa / 1-SYN-404
Tipo de Servicio	Reparación y mantenimiento
Precio	650,00

Detalles del odómetro

Valor del odómetro	0,00 kilometers

Detalles adicionales

Fecha	02/09/2012
Comprador	The Jackson Group, William Thomas
Proveedor	Axelor
Referencia factura	4586

Servicios incluidos

Servicio	Coste
Repuesto del alternador	50,00
Repuesto de la rótula	25,00
Inspección de batería	100,00
	175,00

Notas

Figura 5.8: *Vista tipo formulario del servicio de un vehículo.*

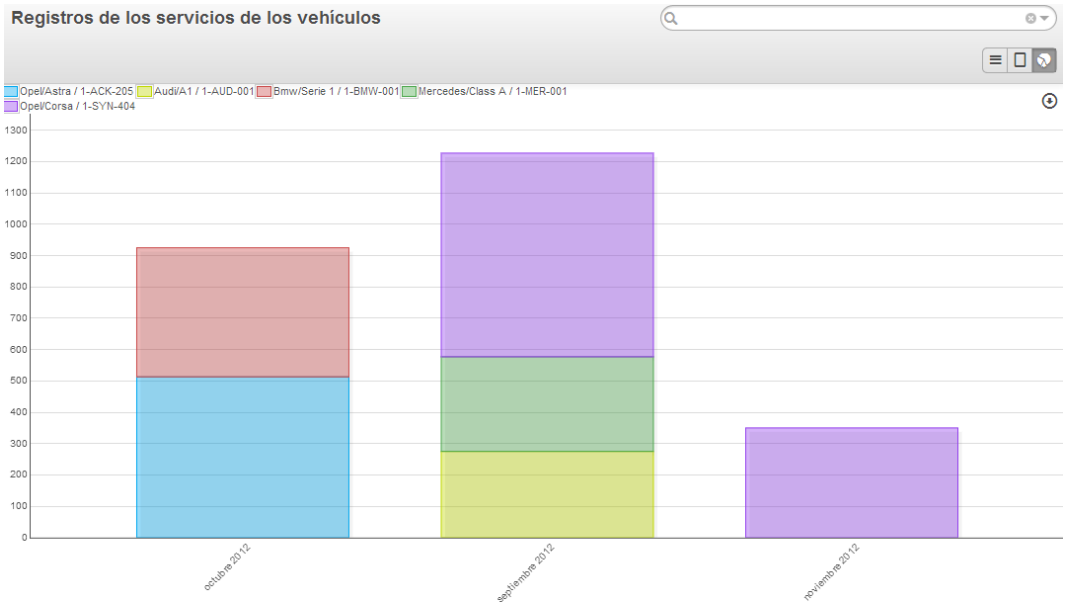


Figura 5.9: *Vista tipo gráfico de los registros de servicios de los vehículos.*

- Costes del vehículo: especifica todos los costes en los que incurre cada vehículo (por contrato, de combustible, etc.). Nuevamente, este apartado se puede visualizar como lista, formulario o gráficos.

Los elementos que están en la categoría "Configuración" ofrecen la posibilidad de crear y modificar los modelos y las marcas de los vehículos, así como su estado (*In shop*, *Active*, *Inactive* y *Sold*) y el tipo de servicios y contratos relacionados con los vehículos.

5.2. Algunos ejemplos de los cambios realizados en el módulo

Una vez presentadas las características del módulo del que se parte, es el momento de plantearse qué cambios se pueden hacer para lograr una mayor adaptación de dicho módulo a una PYME, conscientes de que la personalización real debe ser efectuada por cada una en base a sus necesidades.

Si bien es cierto que el módulo de la flota ofrece una amplia variedad de posibilidades, también posee una serie de carencias que se pueden suplir teniendo en cuenta todo lo descrito previamente en esta memoria.

Por este motivo, se establecerán una serie de pautas para la modificación del módulo de la flota, de modo que sirva para ejemplificar el proceso y que cualquier empresa lo pueda aplicar en la adaptación de lo que particularmente sea necesario para ella.

Para empezar, se añade una variable en el modelo de vehículos de tipo binario. Lo que permite esta variable es adjuntar un archivo en el lugar donde esté definida. Se muestra la utilización de esta variable para poder cargar un contrato de compra del vehículo, por ejemplo. Aunque muchos de los detalles importantes del contrato, como la fecha y el tipo, vienen definidos directamente, tener la posibilidad de acceder al contrato completo en formato PDF, por ejemplo, puede ser de gran utilidad.

Por tanto, como ya se ha comentado, dentro del modelo "Información del contrato en el vehículo", pulsar "Editar" y seleccionar "Añadir un elemento". El nombre que se les va a dar a todos los elementos nuevos empieza por "x_" para diferenciarlo de los demás. Este nombre no admite espacios ni caracteres como las tildes.

El nombre definido de la variable binaria es "x_contrato_completo", con la etiqueta que identifica al campo "Contrato completo". Como ya se ha dicho, es una variable de tipo binario (seleccionar en "Tipo de campo" el valor *binary*), cuyo aspecto final en el modelo queda tal y como se muestra en las Figuras 5.10 y 5.11:

Fecha de inicio del contrato	Fecha de expiración del contrato	Vehículo	Tipo	Proveedor	Coste de activación	Importe del coste recurrente	Frecuencia de los costes recurrentes	Estado	Contrato completo
01/01/2014	24/06/2014	Opel/Astra / 1-ACK-205	Leasing	Axelor	0,00	150,00	Semanalmente	Para cerrar	Descargar (15.06 Kb)
01/01/2014	30/06/2014	Opel/Corsa / 1-SYN-404	Leasing	Axelor	0,00	20,00	Diariamente	Para cerrar	Descargar (0.00 Bytes)
01/01/2014	31/12/2014	Bmw/Serie 1 / 1-BMW-001	Leasing	Axelor	0,00	400,00	Mensual	En proceso	Descargar (0.00 Bytes)
01/01/2014	31/12/2014	Audi/A1 / 1-AUD-001	Leasing	Axelor	0,00	4000,00	Anualmente	En proceso	Descargar (0.00 Bytes)

Figura 5.10: *Vista tipo lista donde se ve la opción de descargar el contrato.*

Figura 5.11: *Vista tipo formulario del contrato de un vehículo.*

Lo siguiente que se puede añadir es una variable de selección que permita distinguir entre los distintos tipos de vehículos que pueda tener la empresa. Este dato puede ser importante y no viene especificado en el módulo original, por lo que añadiremos un elemento de nombre "x_tipo_vehiculo", la etiqueta de campo "Tipo de vehículo" y el valor de "Tipo de campo" *selection*. Las posibles opciones de selección que ofrecerá la variable hay que introducirlas al estilo Python, como se muestra en la Figura 5.12 en "Opciones de selección". La primera palabra dentro de cada paréntesis es la etiqueta de la opción y la segunda, el nombre con el que se mostrará.

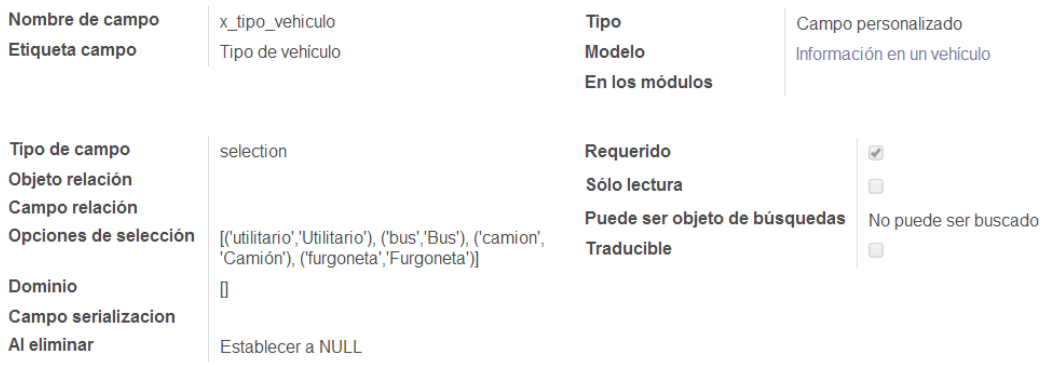


Figura 5.12: *Edición de un campo tipo selection.*

Otro campo que realiza la misma función es *reference*. La diferencia es que en éste último se puede limitar el tamaño de la selección. La declaración de opciones se realiza de la misma manera.

El resultado final en ambos casos es un menú desplegable que aparecerá en el lugar que se indique en la ventana de la vista (Figura 5.13).

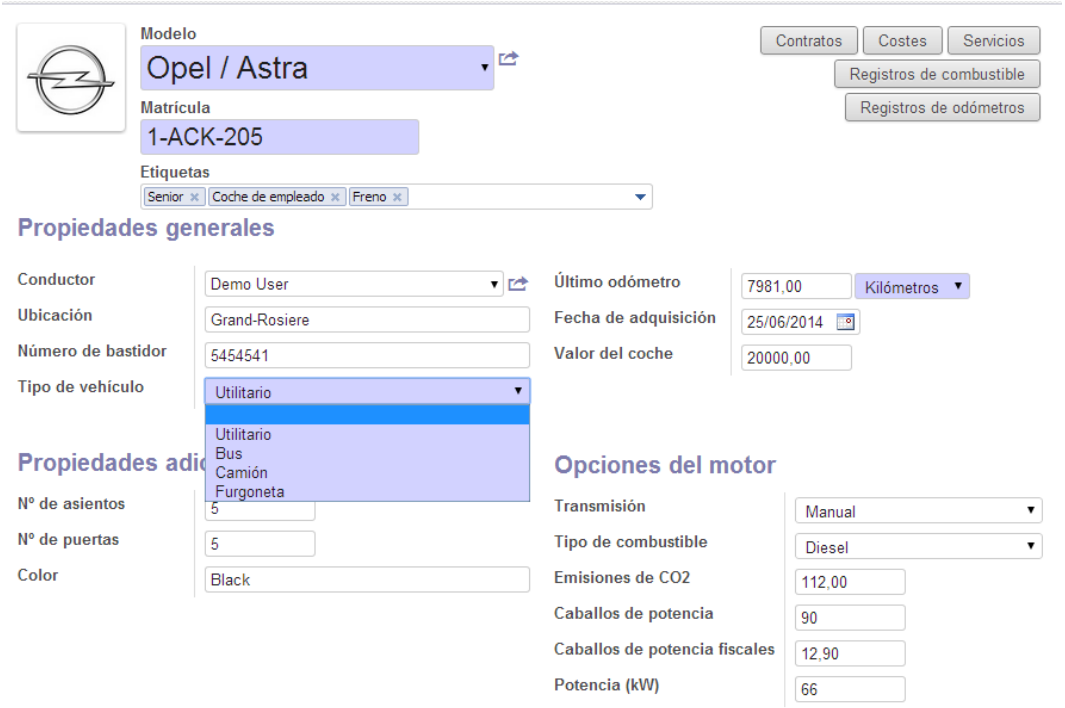


Figura 5.13: *Menú desplegable en "Tipo de vehículo".*

Se añade ahora un campo que permita hacer alguna descripción del vehículo o añadir un comentario. Con este fin existen dos posibilidades: un campo tipo *char* o uno tipo *text*. La diferencia entre uno y otro es que en los de tipo *text* no hay limitación de caracteres, mientras que en los de tipo char es el usuario el que impone una limitación. Por ejemplo, en la información de los vehículos se puede habilitar un espacio para realizar comentarios o hacer alguna aclaración. El aspecto final sería el de la Figura 5.14.

Figura 5.14: *Campo de texto en "Comentarios".*

Otro campo que se utiliza para dejar constancia de un texto, pero con más opciones visuales, es el llamado "html" (Figura 5.15). Funciona de la misma manera que los otros dos.

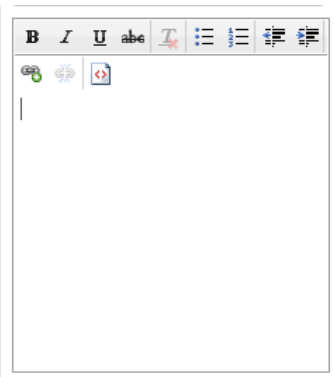


Figura 5.15: *Campo de tipo html.*

También se puede añadir una variable booleana (indica si el valor es cierto o es falso) para señalar la presencia o no de un complemento en concreto del vehículo. Por ejemplo, se puede utilizar para informar que un coche dispone de aire acondicionado (Figura 5.16).

Figura 5.16: *Variable booleana en "Aire acondicionado".*

A continuación se incluye un campo que permita conocer a partir de qué fecha el vehículo en cuestión se encuentra disponible. Para ello se crea un nuevo elemento de tipo *date*. Mediante este campo se integra el calendario con los vehículos. Esta variable puede ser útil si con dicho automóvil se van a realizar trayectos con la vuelta determinada. Cada vez que un empleado utilice el medio deberá indicar cuándo terminará su uso. Así, el resto de personas que utilizan ese vehículo podrán saber en qué momento vuelve a estar a su disposición. La apariencia de este campo en la vista formulario se puede apreciar en la Figura 5.17.

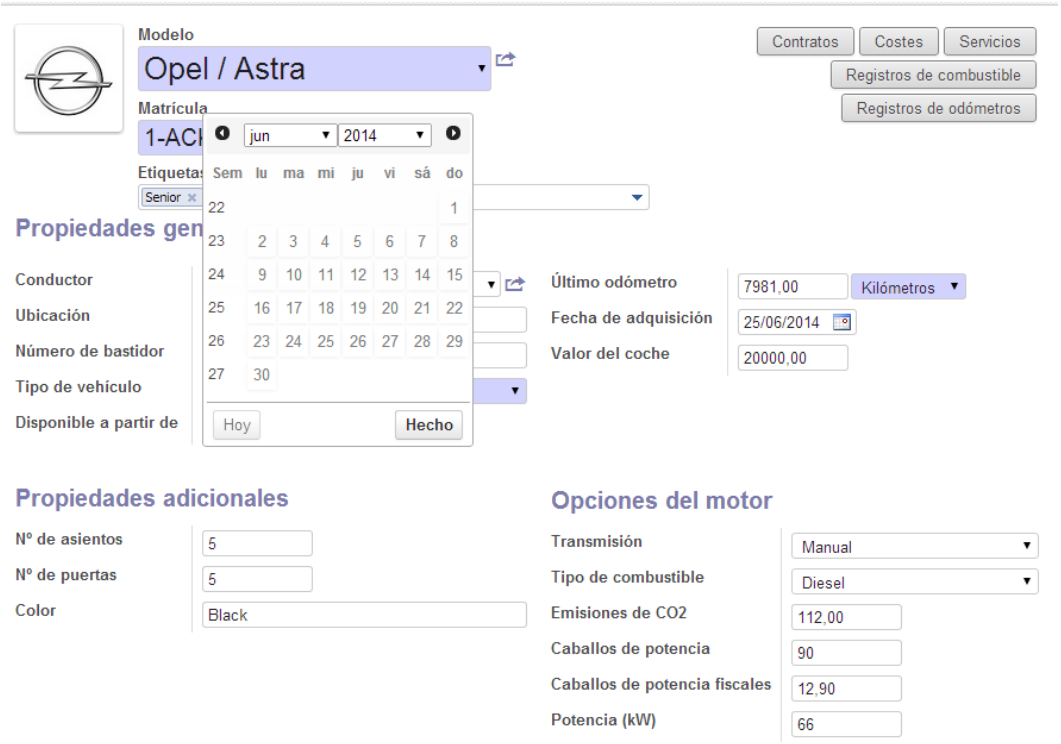


Figura 5.17: *Calendario en un campo tipo date.*

La otra alternativa para hacer esto mismo consiste en añadir la variable, pero que sea de tipo *datetime*. La diferencia radica en que con este tipo de campo se puede ampliar la información de la fecha añadiendo la hora (Figura 5.18).

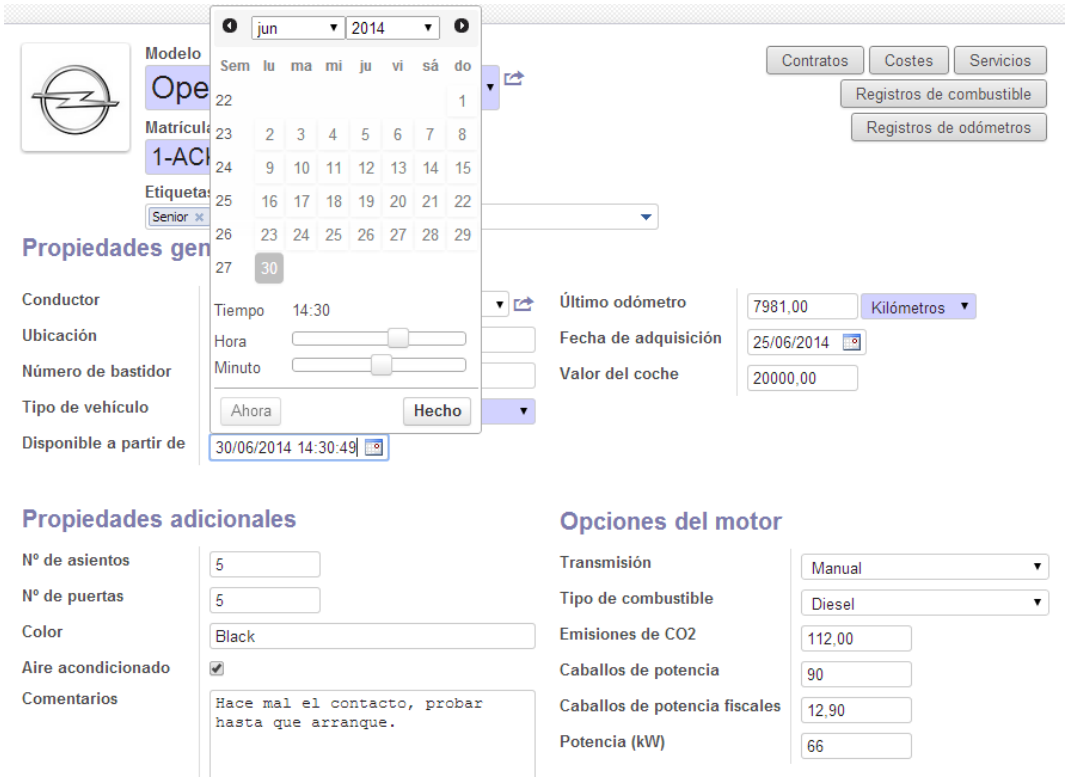


Figura 5.18: *Calendario y hora en un campo tipo datetime.*

También se pueden realizar modificaciones que afecten a otros módulos de OpenERP. Para ejemplificarlo, ha sido necesario descargar el módulo de *recursos humanos* desde la tienda de aplicaciones y aprovechar los datos de prueba para mostrar el resultado.

Éste es el caso de la variable añadida "Piloto". El módulo original incluye un campo de "Conductor" que solamente permite asignar a los *partner*, pero puede ser conveniente que exista un apartado donde se muestren los empleados que utilizan cada vehículo. Por ello se crea un campo relacional "muchos a muchos" (el "Tipo de campo" es *many2many*) ya que un empleado puede conducir varios vehículos y un mismo vehículo puede ser, a su vez, utilizado por varios pilotos. Para indicar con qué objetos está relacionado hay que señalar el identificador en el apartado "Objeto relación"; de ahí es de donde se seleccionarán los pilotos de los automóviles (del apartado "*hr.employee*"), tal y como aparece en la Figura 5.19.

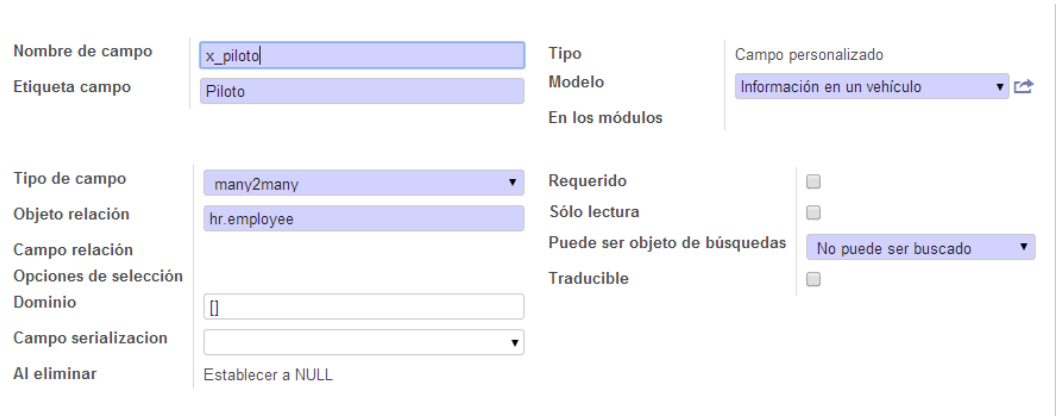


Figura 5.19: *Edición de un campo many2many.*

El aspecto de este cambio en la vista formulario de los vehículos se muestra en la Figura 5.20. Como se puede apreciar, se ha añadido una pequeña tabla con la información más relevante del empleado.

Figura 5.20: *Variable many2many en "Piloto".*

Otra opción interesante que no se encuentra en el módulo básico de la flota es la de determinar el estado del vehículo. Originalmente existe una lista con los posibles estados del vehículo, pero no existe la posibilidad de asociar en qué estado se encuentra el mismo.

El modelo que define los diferentes estados en los que se pueden encontrar los vehículos ya está creado en el módulo original; es "Estado del vehículo". Los estados que abarca son, como se ha comentado en alguna ocasión: *In shop*, *Active*, *Inactive* y *Sold*. Esta lista se puede completar con los valores que se crean oportunos. En nuestro caso, la lista de estados se ha traducido al castellano y se ha incorporado la opción "En trayecto".

A continuación, es necesario crear un campo que relacione el vehículo con los posibles estados. El tipo de variable que satisface esta condición es *many2one*, ya que es posible que varios vehículos se encuentren en un mismo estado, por ejemplo, estando muchos activos.

El resultado de añadir esta variable a la vista tipo formulario de los vehículos se puede observar en la Figura 5.21:

Figura 5.21: *Variable many2one en "Estado del vehículo".*

Algo parecido se puede hacer, por ejemplo, con los seguros. Se definen en un nuevo modelo los tipos de seguro con los que trabaja la empresa y, por otra parte, se añade el campo pertinente a los vehículos. De la misma manera que antes, varios vehículos pueden tener un mismo tipo de seguro, luego la variable también es de tipo *many2one*.

Para realizar esto se crea una lista con las clases de seguro: "A terceros", "A todo riesgo" y "A todo riesgo con franquicia". En la Figura 5.22 se muestra la vista formulario donde aparece dicho campo.

Figura 5.22: *Variable many2one en "Clase de seguro".*

Ahora se va a incluir un campo que relacione cada vehículo con los permisos de los que dispone. Por lo tanto, se tiene que crear una variable de tipo *one2many*, ya que un mismo vehículo puede tener más de un permiso simultáneamente.

El resultado, como se puede ver en la Figura 5.23, consiste en un campo en forma de tabla, donde se pueden ir añadiendo los permisos de los

que el vehículo disponga. Además, se incluyen dos columnas que indican el teléfono y el correo electrónico del centro donde se validó cada permiso.

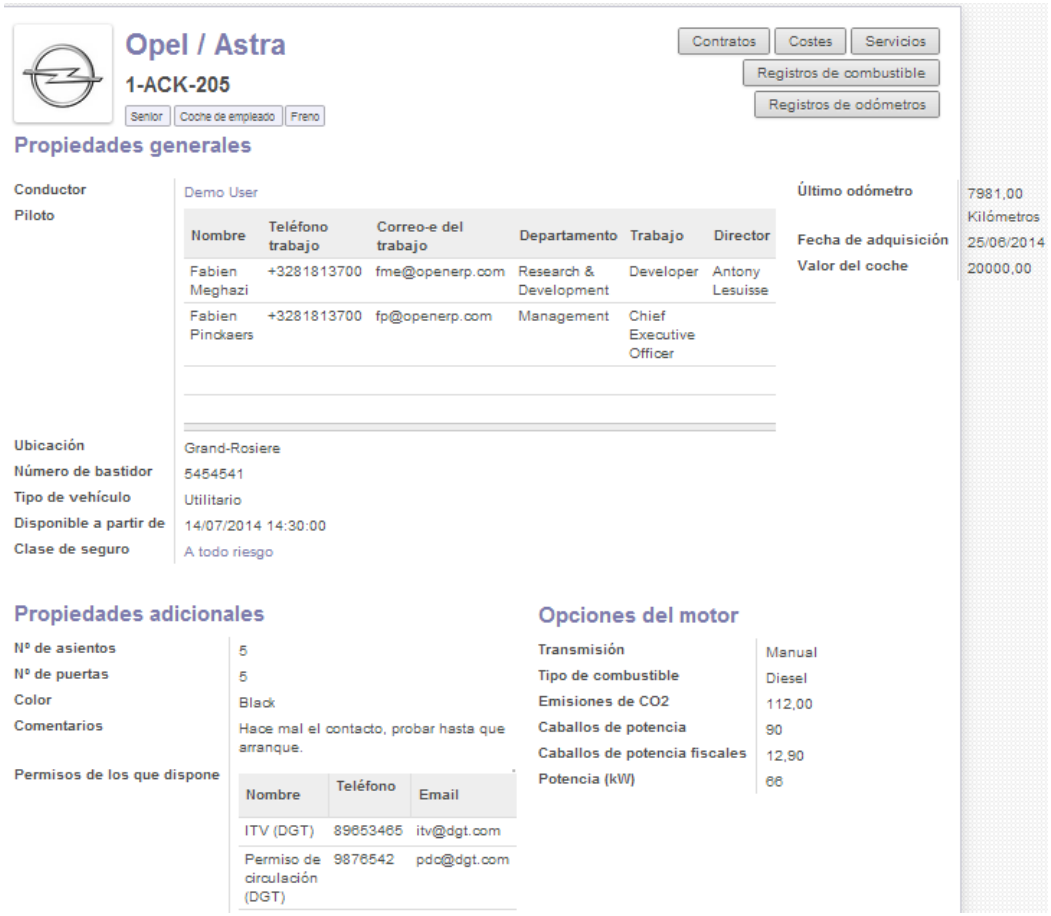


Figura 5.23: *Variable one2many en "Permisos de los que dispone".*

Con todos estos cambios, se puede dar por concluida la personalización del módulo. Como se ha dicho en otras ocasiones, esto no es más que una versión ilustrativa acerca de lo que se puede hacer con el módulo de la flota.

A continuación, cada empresa tiene que estudiar cuáles son los cambios que le convienen, implementarlos y posteriormente validarlos. Pero no basta con quedarse ahí, sino que es necesario aplicar un proceso de mejora continua. Según se vaya utilizando el ERP con las nuevas funcionalidades se irán descubriendo otras carencias y/o nuevas posibilidades no contempladas al principio.

La idea es que este prototipo del módulo de la flota de vehículos pueda ser de utilidad a una pequeña o mediana empresa, ya sea porque requiera los cambios ejemplificados en este capítulo o porque se pueda valer del procedimiento para poder cubrir sus propias necesidades.

Además, no sólo hay que quedarse ahí, sino que hay que extender estas posibilidades de personalización a los demás módulos que utilice la empresa. El fin último es que la organización consiga un sistema gestor de la información integrado y totalmente personalizado a las características de su negocio.

CAPÍTULO VI

EVALUACIÓN ECONÓMICA

En el presente capítulo se va a efectuar el análisis económico del trabajo asociado al proyecto realizado, para así poder determinar cuantitativamente los costes que supone su elaboración. En este tipo de proyectos, el grueso de los costes es debido a la mano de obra directa encargada de trabajar con el software, que es la que desarrolla las actividades encaminadas a la consecución de los objetivos. Se habla, por tanto, de un trabajo de ingeniería.

Normalmente, en la contabilidad de costes de un proyecto se distinguen entre costes directos e indirectos. Los costes directos son los imputables a las actividades concretas, como por ejemplo los costes de personal, del hardware y software necesarios, fungibles o del material de oficina. En los costes indirectos se contabilizan aquellos que no se pueden asignar directamente y son de carácter general, como los gastos de electricidad, teléfono, etc.

6.1. Planificación temporal

Se requiere conocer con detalle cómo ha sido la distribución temporal de las actividades del proyecto para poder determinar los costes, por lo tanto, se hace una distinción de las etapas de las que ha constado el proyecto y se analizan mediante una conocida herramienta como es el diagrama de Gantt.

Para empezar hay que conocer cómo se utiliza dicho diagrama para efectuar un análisis correcto. El objetivo del diagrama es resumir gráficamente las distintas actividades del proyecto (su duración y sus dependencias). Para estimar la duración de las actividades es necesario conocer los recursos disponibles para la elaboración del trabajo, en este caso ha intervenido un graduado en Ingeniería en Organización Industrial (capital humano) y un ordenador portátil (recursos técnicos y materiales).

Ahora hay que saber las fases en las que se ha estructurado el trabajo y son las que siguen:

- Análisis del trabajo y viabilidad de su ejecución.
- Selección del ERP, teniendo en cuenta la adecuación al proyecto.
- Esquema del trabajo o estructuración de las etapas, especialmente de la etapa posterior.
- Desarrollo del trabajo o ejecución de las actividades planificadas en el paso anterior. Sus actividades comprenden la búsqueda de información del ERP, aprendizaje de su funcionamiento y ejecución del trabajo.
- Seguimiento del proyecto y comprobación del cumplimento de los plazos y objetivos.
- Elaboración de la memoria.

Existe una realimentación entre la planificación del trabajo y el desarrollo del mismo, ya que hay cosas que se han tenido que modificar y otras que no aparecían en dicho plan ha sido necesario incorporarlas.

El diagrama de Gantt al que se ajusta este proyecto aparece en la Figura 6.1. Dicho diagrama ha sido elaborado con la herramienta Microsoft Project.

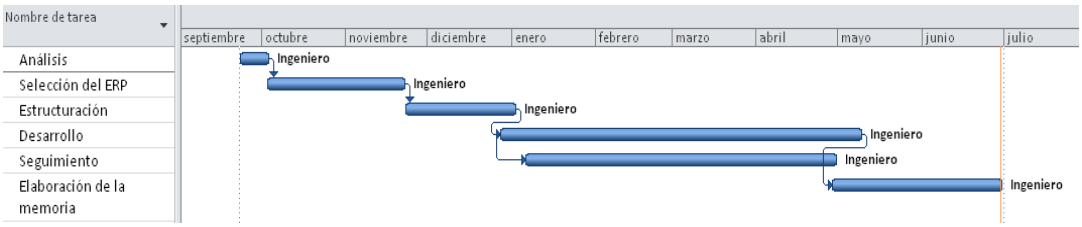


Figura 6.1: *Diagrama de Gantt del proyecto.*

6.2. Costes del TFG

La asignación de los recursos a las distintas actividades de un proyecto incurre en una serie de costes. Hay que llevar a cabo un control sobre los mismos e intentar mantenerse en los límites fijados en el presupuesto inicial.

La estimación de los costes se realiza una vez definidas las actividades, sabiendo su duración y sus precedencias. Para poder realizar un estudio apropiado de los costes, se pueden seguir diversas clasificaciones. En este caso se hace la distinción entre costes directos e indirectos.

6.2.1. Costes directos

Como ya se ha dicho al principio del capítulo, los costes directos son los que se pueden imputar directamente a las actividades. En el presente trabajo entran dentro de esta clasificación los costes de personal y los de material.

6.2.1.1. Costes de personal

La distribución de horas del graduado en Ingeniería en Organización Industrial según las etapas vistas en el apartado anterior es la siguiente:

- Análisis: 15 horas.

- Selección del ERP: 62 horas (en un principio el trabajo iba a ser realizado en Tryton).
- Estructuración: 45 horas.
- Desarrollo: 180 horas.
- Seguimiento: 50 horas.
- Elaboración de la memoria: 70 horas.

El trabajo se realizó durante el curso 2013/2014 del alumno, por lo que las horas disponibles para la realización del trabajo se reducen a 16 horas semanales

Esas horas corresponden con el trabajo del ingeniero, que de media, están retribuidas con 30 €/h. Con esta estimación la distribución de costes del personal por etapas es la que se muestra en la Figura 6.2:

	Duración (h)	Coste de personal (€)
Análisis	15	450
Selección del ERP	62	1860
Estructuración	45	1350
Desarrollo	180	5400
Seguimiento	50	1500
Elaboración de la memoria	70	2100
TOTAL	**422**	**12660**

Figura 6.2: *Tabla-resumen de los costes de personal.*

6.2.1.2. Costes de material

En cuanto a los costes de material, se tienen en cuenta los correspondientes al equipo informático, software y demás material fungible.

Para el equipamiento técnico es necesario calcular la amortización lineal del ordenador portátil teniendo en cuenta su vida útil. Esta es la herramienta fundamental del trabajo, ya que se utiliza para todas las actividades y de forma continuada.

La inversión inicial corresponde con el precio de compra del ordenador, 519 €. La vida útil de este tipo de dispositivos es de tres años aproximadamente, pasados los cuales su valor residual se aproxima a cero. La amortización correspondiente es:

$$\frac{519\ €}{36\ meses} * 9\ meses = 129{,}75\ €$$

La parte de los costes debida al software es gratuita, ya que se trata de un software libre por el que no hay que pagar para su uso.

Por otro lado, los costes fungibles de oficina no son amortizados e incluyen las impresiones, copias, CD, encuadernación, etc. Son, aproximadamente, 80 €.

Los costes de material se resumen en la Figura 6.3:

	Coste de material (€)
Equipo informático	129,75
Software	0
Fungibles	80
TOTAL	209,75

Figura 6.3: *Tabla-resumen de los costes de material.*

6.2.2. Costes indirectos

En esta clasificación, se agrupan los costes no imputables directamente a las actividades del proyecto.

Los que se deben contabilizar en este apartado son los distintos gastos de explotación, de electricidad, teléfono, conexión a internet, entre otros. Las estimaciones de estos costes aparecen en la Figura 6.4:

	Coste (€)
Climatización	250
Electricidad	410
Internet	240
Otros	80
TOTAL	**980**

Figura 6.4: *Tabla-resumen de los costes indirectos.*

6.3. Coste total y resultado económico

Una vez vistos los costes directos e indirectos en los que incurre el proyecto, se puede conocer el coste total del mismo sumando ambas cantidades. En la Figura 6.5 se resume:

	Costes totales (€)
Costes directos	12869,75
Costes indirectos	980
TOTAL	**13849,75**

Figura 6.5: *Tabla-resumen de los costes totales.*

En el caso de querer realizar la venta de este trabajo, habría que aplicar un margen comercial de beneficios adecuado. Un valor normal del beneficio respecto de los costes sería un 30%. Por lo tanto, el precio de venta podría ser de 18004,67 €.

Como el proyecto no está orientado a una empresa en particular, sino que se puede considerar de carácter general y puede ser de utilidad para muchas empresas, sería conveniente hacer llegar el resultado al mayor número posible de ellas.

Una estimación realista para el número de empresas que pudieran estar interesadas en comprarlo sería de 15 empresas. Esto supondría que habría que vender el trabajo a cada una a un precio de 1200,31 € para cumplir las expectativas de beneficio.

No quiere decir que la empresa sólo tuviera que pagar este precio, sino que podría incluirse a este valor el coste asociado a servicios auxiliares como el de instalación o personalización del software.

En resumen, el resultado económico del proyecto se muestra en la Figura 6.6:

	Costes totales (€)
Costes directos	12869,75
Costes indirectos	980
TOTAL	**13849,75**

Precio total del proyecto	18004,67 €.
Precio de venta	1200,31 €
Beneficio	**4154,92 €**

Figura 6.6: *Tabla-resumen del estudio económico.*

Por último, el resultado en forma gráfica aparece en la Figura 6.7:

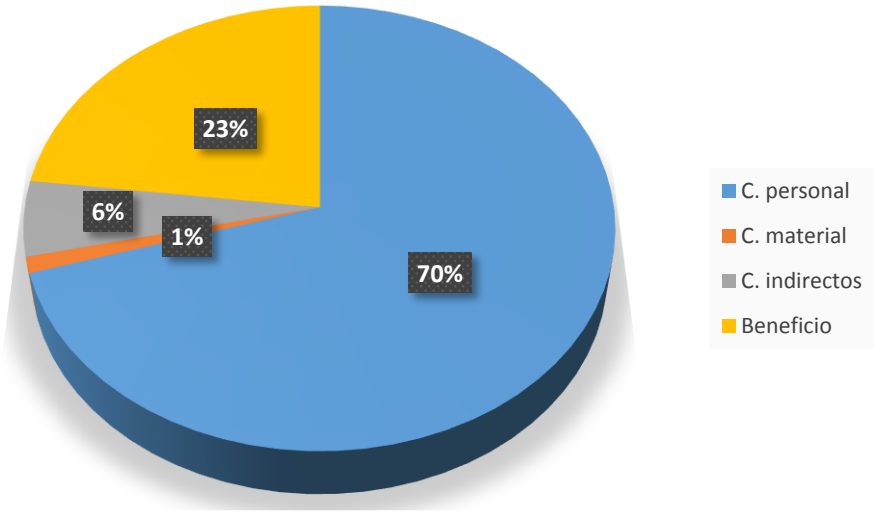


Figura 6.7: *Gráfico del resultado económico.*

CAPÍTULO VII

CONCLUSIONES Y LÍNEAS FUTURAS

Una vez finalizado el desarrollo técnico del trabajo, es el momento de evaluar en qué grado se han satisfecho los objetivos planteados al principio del mismo y de extraer las conclusiones más relevantes del trabajo realizado.

También se presentan algunas posibilidades o alternativas de evolución en el ámbito que se está tratando.

7.1. Conclusiones

Las grandes compañías generalmente cuentan con sistemas ERP, en su mayoría propietarios, para apoyar la planificación de los recursos empresariales, además del personal necesario para su gestión y personalización. Sin embargo, no suele ocurrir lo mismo con las PYMES, donde los recursos son más limitados. Estas empresas constituyen el motor de crecimiento clave de nuestra economía, por lo que es importante establecer mecanismos que faciliten su labor.

Partiendo de esta idea, el presente proyecto surgió con el objetivo de estudiar los mecanismos de personalización de módulos ERP a fin de lograr la adaptación y personalización de un módulo orientado a la logística, de modo que sirviese de guía y apoyo a cualquier PYME que quisiera personalizar sus módulos de forma fácil e intuitiva con el menor consumo de recursos posible.

Para lograr este objetivo, en primer lugar se realizó una investigación sobre los ERP existentes hoy en día y se seleccionó el más apropiado para el trabajo, resultando ser este OpenERP. Cabe señalar que, aunque en un principio se planteó utilizar Tryton, los problemas que surgieron relacionados con los permisos en Linux nos hicieron cambiar la dirección del proyecto hacia OpenERP.

Con el ERP seleccionado surgía una nueva dificultad, la de cómo llevar a cabo el objetivo principal. Para dilucidarlo, se exploraron las distintas posibilidades existentes y tras un análisis comparativo, se decidió modificar el módulo a través de la interfaz.

Después de realizar todas estas tareas, se está en disposición de extraer una serie de conclusiones acerca del mundo del ERP, de los tipos, de las marcas comerciales y del procedimiento, utilidad y resultado de modificación de un módulo.

La necesidad de poseer un sistema de gestión integrado queda fuera de toda duda tal y como se encuentra el contexto empresarial actual. La rápida evolución de los procesos, productos e incluso de las formas de negocio

empuja a las organizaciones a controlar su información de forma segura e íntegra. La mejor herramienta para lograr esa meta y permitir que la empresa avance, adaptándose con ella, es el ERP.

Cuando ya se conoce la necesidad de estos instrumentos para gestionar la organización y simplificar los procesos, hay que analizar cuál de las clases de ERP es más conveniente implantar. Está claro que un estudio es exhaustivo es importante, porque determina cuál será la herramienta básica de gestión. Adquirir un sistema que no sea el adecuado acarreará consecuencias negativas para la organización.

Una lectura que se puede extraer tras analizar las tres clases de ERP (sistemas propietarios, de software libre o SaaS) es que las soluciones *opensource* proporcionan un grado de satisfacción alto en comparación con los otros tipos. Están consiguiendo una alta calidad en la herramienta y además consiguen una adaptación mayor que los otros tipos a las empresas que interesan en este caso, las PYMES.

Una vez seleccionado el tipo de ERP más adecuado para el caso que se trate, hay que elegir entre las diferentes marcas comerciales atendiendo a los criterios que se consideren oportunos.

Otra experiencia que se deriva de la realización del trabajo es que hay que tener en cuenta la rápida evolución de estos instrumentos. Cuando se inició este proyecto se comenzó a utilizar la última versión de existente hasta el momento de OpenERP, la 7.0. En su conclusión, ya está disponible una nueva versión y hasta ha habido un cambio tan radical como cambiar el nombre de la marca a Odoo, del que se habla más adelante.

La última lectura que se puede extraer tras la experiencia del trabajo es que realmente es necesaria la personalización del sistema ERP para cada empresa. Aunque la herramienta ofrezca muchas ventajas en la gestión, si no se encuentra adaptada a la forma de funcionar del negocio en concreto surgirán ineficiencias. Por lo tanto, hay que exprimir la capacidad y las posibilidades que brindan estos instrumentos y optimizar sus funcionalidades.

Considerando lo visto hasta ahora, el contenido de los capítulos y el resultado final en el ERP, se puede determinar que sí se ha cumplido el objetivo prioritario.

Las posibilidades que ofrece OpenERP son sobradamente adecuadas para la unificación y ordenación de los distintos departamentos de la empresa en un lugar común y totalmente funcional. La versatilidad de este software hace posible una adaptación a las características individuales de cada empresa de forma muy satisfactoria.

Esta adaptabilidad es lo que se ha pretendido mostrar ejemplificándolo con el módulo de la flota. Se han añadido funcionalidades como la de asignar conductores a los vehículos de entre los empleados contratados y que aparezcan en la base de datos, adjuntar documentos en la información de los vehículos como puedan ser contratos y seguros o habilitar espacios para la adición de notas, entre otras.

Aunque lo realizado en este trabajo puede ser de gran ayuda para una PYME, siempre será necesaria la elaboración de un estudio previo para conocer la complejidad y las particularidades de cada una de ellas, que es lo que conduce a una serie de necesidades individuales y propias.

Implantar un ERP en la empresa como sistema de gestión puede parecer en un primer momento una decisión traumática, pero sin duda aportará muchas ventajas al negocio. Particularmente, una herramienta *opensource* ofrece muchas ventajas económicas, unido a unas funcionalidades de gestión sorprendentemente buenas y unas posibilidades de personalización prácticamente ilimitadas.

Al principio el manejo de la herramienta requiere tiempo, ya que hay que familiarizarse con el software. Es necesario compromiso y dedicación, pero el resultado técnico merece sin duda la pena. La finalización y resultado del trabajo han sido satisfactorios, pero siempre es necesario continuar con el ciclo de mejora continua.

7.2. Líneas futuras

El dominio de la herramienta no es suficiente, las empresas no se pueden permanecer en esta fase si no quieren quedarse rezagadas en un entorno empresarial que evoluciona tan rápido y es tan competitivo.

El cumplimiento del objetivo de la personalización del módulo es el primer paso hacia la optimización en el sistema de gestión empresarial. OpenERP es un instrumento en constante evolución, por lo que resulta complicado seguir al tanto de las actualizaciones y mejoras que se van promoviendo.

Una alternativa es el desarrollo por cuenta propia de aplicaciones que aporten nuevas funcionalidades o buscar nuevas opciones útiles para este proyecto entre los módulos descargables. Para ello, hay que mantenerse alerta para no perder ninguna novedad.

Otra posibilidad es probar la herramienta desde otra base, es decir, utilizar otro sistema operativo como Linux. Está comprobado que el programa se comporta de manera distinta dependiendo del sistema desde el que se ejecute. El funcionamiento desde aquí sería diferente y también las opciones de modificación ofrecerían más variantes, al menos habría soluciones para afrontar la modificación del módulo por medio de los archivos de programación.

Una vez que se hubiera abierto la posibilidad de integrar cambios por medio del código, se podría realizar un estudio para abrir nuevas variantes a partir de las posibilidades que aporta este método. Un conocimiento profundo del lenguaje de programación abriría otras puertas a la hora de implementar cambios que no se podrían aplicar de otra manera.

Continuando con la evolución de OpenERP, es necesario comentar en este apartado de visión de futuro la nueva versión, el nuevo concepto que se quiere instaurar desde OpenERP. Se trata de Odoo (Figura 7.1), su nueva imagen de marca.

Figura 7.1: *Logotipo de Odoo.*

Con el cambio hacia Odoo, se pretende ir más allá de un ERP, pero manteniendo siempre el modelo *opensource*. Es una herramienta que aspira a mejorar la empresa en más de un sentido. Para ello, aglutina muchas soluciones desde una misma aplicación, es decir, es un sistema unificado.

Aporta recursos empresariales en forma de aplicaciones para:

- Crear y diseñar la página web de la empresa (también de comercio electrónico y blogs).
- Incrementar las ventas (los conocidos CRM, presupuestos y puntos de venta).

- Gestionar el negocio (las aplicaciones que ya están maduras desde OpenERP como los de facturación, contabilidad o gestión de almacén).
- Potenciar el marketing (vía email, chat, eventos, creación de comunidades…).
- Satisfacer a los empleados (redes sociales de empleados, reclutamiento, evaluación…).
- Potenciar la productividad (*business intelligence*, grupos de discusión, notas, etc.).

Con Odoo, se ha intentado simplificar aún más varios aspectos: el tiempo que se requiere para manejar esta herramienta se ha reducido drásticamente; la integración con Office es aún mejor, facilitando la importación o exportación de datos; sigue trabajando vía web, por lo que no requiere la instalación del cliente y, además, se puede ejecutar desde dispositivos móviles y tabletas; la flexibilidad seguirá siendo una de las señas de identidad del sistema, para la personalización y la solución de necesidades con los módulos disponibles.

Esta herramienta será muy poderosa en el futuro e incorporarla cuanto antes puede ser una gran decisión. Aun así, se pueden explorar otras alternativas en cuanto a la marca que se quiera implantar, como pueden ser Openbravo o Tryton.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

Fayol, H. (1916). *Administration industrielle et générale; prévoyance, organisation, commandement, coordination, controle.* París: H. Dunod et E. Pinat.

Ferrero, R. L. (2012). *Desarrollo de un módulo en OpenERP para la gestión de la flota de vehículos.* Valladolid.

Payo, A. U. (2010). *Análisis de mercado de software ERP para la gestión empresarial: un enfoque hacia el código abierto.* Valladolid.

Sola, Ó. S. (2012). *Análisis y desarrollo de un sistema de gestión empresarial basado en software ERP para una empresa logística.* Valladolid .

RECURSOS WEB

Álvarez, M. A. (19 de Noviembre de 2003). *Qué es Python*. Obtenido de desarrolloweb.com: http://www.desarrolloweb.com/articulos/1325.php

Areny, A. C. (7 de Octubre de 2008). *OpenERP Spanish Localization Project*. Obtenido de https://launchpad.net/openerp-spain

Bulchand, J. (11 de Noviembre de 2009). *ERPs de Software Libre*. Obtenido de Gaceta tecnológica: http://www.gacetatecnologica.com/opinion/tribuna/1191-erps-de-software-libre-.html

Carlos, J. S. (24 de Noviembre de 2013). *Agregar campos a formularios de OPENERP 7 mediante modulos, usando su framework OPENOBJECT – Parte I*. Obtenido de Odoo ERP - Perú: http://huber.salazarcarlos.com/?p=396

Carlos, J. S. (10 de Junio de 2013). *Creación de módulos en OPENERP 7 en modo gráfico*. Obtenido de Odoo ERP - Perú: http://huber.salazarcarlos.com/?p=339

Carlos, J. S. (8 de Marzo de 2014). *Creación de módulos en OPENERP 7 usando su propio framework OPENOBJECT – Parte II*. Obtenido de Odoo ERP - Perú: http://huber.salazarcarlos.com/?p=445

Domínguez, G. P. (6 de Enero de 2014). *Programando con OpenERP y Python*. Obtenido de http://poncesoft.blogspot.com.es/2014/01/desarrollo-de-modulos-para-openerp-61-70.html

Edwars, J. (2011). *Oracle Aplicaciones*. Obtenido de ERP (Enterprise Resource Planning): http://oracle.abast.es/oracle_erp.shtml

Frank, J. (Diciembre de 2010). *blog estudiante de ingenieria industrial*. Obtenido de http://collazosflores.blogspot.com.es/2010_04_01_archive.html

García, A. (22 de Febrero de 2013). *TRESCloud*. Obtenido de Creacion Módulo OpenERP TRESCloud: http://blog.trescloud.com/2013/02/creacion-modulo-openerp-trescloud.html

Gestiweb. (2012). *Historia del ERP*. Obtenido de gestiweb: http://gestiweb.com/?q=content/212-historia-del-erp

González, G. (10 de Julio de 2012). *Software OpenERP & Metodología ERP*. Obtenido de http://www.slideshare.net/guszkoh/software-openerp-metodologa-erp

Ibáñez, S. (17 de Enero de 2013). *¿Qué proveedor de ERP elegir?* Obtenido de Blog de Silvia Ibáñez: http://sibamas.blogspot.com.es/2013/01/que-proveedor-de-erp-elegir.html

Informática Hoy. (2008). *Características fundamentales del ERP*. Obtenido de http://www.informatica-hoy.com.ar/software-erp/Caracteristicas-fundamentales-del-ERP.php

Informática Hoy. (2011). *Sistemas ERP de software libre*. Obtenido de http://www.informatica-hoy.com.ar/software-erp/Sistemas-ERP-de-software-libre.php

Informática Hoy. (s.f.). *Comparativa entre ERP de software libre y propietario*. Obtenido de http://www.informatica-hoy.com.ar/software-erp/Comparativa-entre-ERP-de-software-libre-y-propietario.php

Olalde, A. J. (13 de Abril de 2009). *Archivo etiqueta manual programación openerp*. Obtenido de http://www.openerpsite.com/tag/manual-programacion-openerp/

OpenERP. (3 de Noviembre de 2010). *File:OpenERP Client server caption.png*. Obtenido de Wikimedia Commons: http://commons.wikimedia.org/wiki/File:OpenERP_Client_server_caption.png

OpenERP. (2011). *OpennERP Licensing*. Obtenido de openerp.com: http://v6.openerp.com/legal

OpenERP. (s.f.). *Module structure*. Obtenido de https://doc.openerp.com/trunk/server/03_module_dev_01/

OpenERP. (s.f.). *OpenERP Server Developers Documentation*. Obtenido de https://doc.openerp.com/trunk/server/

OpenERP. (s.f.). *OpenERP Spain*. Obtenido de http://openerpspain.com/

OpenERP. (s.f.). *Type of Fields*. Obtenido de https://doc.openerp.com/v6.0/developer/2_5_Objects_Fields_Methods/field_type.html/#functional-fields

openerpsite. (s.f.). *Módulos*. Obtenido de http://www.openerpsite.com/erp-openerp-modulos/

Sánchez, G. S. (2011). *SISTEMAS ERP – ENTERPRISE RESOURCE PLANNING – HISTORIA Y EVOLUCIÓN (I)*. Obtenido de Stratic: http://stratic.es/erp-i-historia-y-evolucion/

Stojanoski, N. (25 de Octubre de 2012). *OpenERP 7.0 compared to OpenERP 6.1*. Obtenido de VION Technology Blog: http://www.vionblog.com/openerp-7-0-compared-to-openerp-6-1/

Wikifab. (2013). *El ABC del ERP*. Obtenido de Wikifab: http://wikifab.dimf.etsii.upm.es/wikifab/index.php/El_ABC_del_ERP

Wikipedia. (Junio de 2014). *Wikipedia*. Obtenido de http://es.wikipedia.org/wiki/PostgreSQL

Wikipedia The Free Encyclopedia. (2014). *Comparison of Tryton and Open ERP*. Obtenido de Wikipedia: http://en.wikipedia.org/wiki/Comparison_of_Tryton_and_Open_ERP

Your ERP software. (s.f.). *Tipos de sistemas ERP*. Obtenido de yourerpsoftware.com: http://www.yourerpsoftware.com/content/28-tipos-de-sistemas-erp