



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# **Integración de venta online en una herramienta de gestión para pequeños comercios**

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Hugo Montoto Martínez

**Tutor:** Pietro Manzoni

**Cotutor externo:** Rafael Mora Carrascosa

Julio de 2018



# Resumen

---

Hoy día son muchas las personas que quieren abrir un pequeño comercio, sin embargo, estas ideas se ven en su mayoría truncadas por la barrera que supone toda la gestión que ello conlleva. La herramienta presentada pretende ayudar a la informatización y automatización de esta gestión. Además la herramienta integra un módulo que permite exponer los productos a un gran público mediante una tienda online personalizable sin necesidad de poseer conocimientos técnicos. La unión de estos dos sistemas provoca un ahorro en el tiempo dedicado al negocio y un incremento notable del público alcanzado.

**Palabras clave:** comercio electrónico, gestión comercial, tienda personalizada

# Abstract

---

Nowadays there are many people who want to start a new business. However, these ideas are mostly truncated by the barrier involving the management. The tool displayed aims to support the computerization and automation of this management. In addition, this tool integrates a module that allows to display products to a large audience through a customizable online store without the need of having any technical knowledge. The consolidation of these two systems leads to savings in the time dedicated to all businesses and a remarkable increase in the public reached.

**Keywords :** e-commerce, business management, personalized store

# Resum

---

Hui en dia són moltes les persones que volen obrir un xicotet comerç, no obstant això, estes idees es veuen majoritàriament truncades per la barrera que suposa tota la gestió que això comporta. La ferramenta presentada pretén ajudar a la informatització i automatització d'esta gestió. A més, la ferramenta integra un mòdul que permet exposar els productes a un gran públic mitjançant d'una botiga online personalizable sense necessitat de posseir coneixements tècnics. L'unió d'estos dos sistemes produeix un estalvi en el temps dedicat al negoci i un increment notable del públic aconseguit.

**Paraules clau:** comerç electrònic, gestió comercial, botiga personalitzada





# Índice

---

|   |           |
|---|-----------|
| Índice  | 4         |
| Índice de imágenes                                      | 7         |
| <hr/>   |           |
| <b>1 Introducción</b>                                   | <b>9</b>  |
| 1.1 Objetivos del proyecto                              | 9         |
| 1.2 Estructura seguida                                  | 9         |
| <b>2 Antecedentes</b>                                   | <b>12</b> |
| 2.1 Alternativas actuales                               | 12        |
| 2.1.1 CMS (Content Management System)                   | 12        |
| 2.1.2 Tienda web personalizada                          | 13        |
| 2.1.3 Marketplaces                                      | 13        |
| 2.2 Herramientas necesarias                             | 14        |
| <b>3 Análisis</b>                                       | <b>17</b> |
| 3.1 Objetivo de la aplicación                           | 17        |
| 3.2 Requisitos mínimos                                  | 18        |
| 3.2.1 Artículos   | 18        |
| 3.2.2 Pedidos de cliente, tickets, facturas y albaranes | 19        |
| 3.2.3 TPV (Terminal punto de venta)                     | 19        |
| 3.2.4 Control de stock                                  | 20        |
| 3.2.5 Clientes y proveedores                            | 20        |
| 3.2.6 Flujo de dinero                                   | 20        |
| 3.2.7 Informes  | 20        |
| 3.2.8 Tienda web  | 21        |
| <b>4 Diseño y metodología del desarrollo</b>            | <b>23</b> |
| 4.1 Tecnologías frontend                                | 23        |

|          |  |           |
|----------|--|-----------|
| 4.1.1    | HTML, CSS, Javascript                          | 23        |
| 4.1.2    | jQuery, jQueryUI, jqGrid                       | 24        |
| 4.1.3    | Slim Image Cropper                             | 28        |
| 4.2      | Tecnologías backend                            | 29        |
| 4.2.1    | PHP, MySQL                                     | 29        |
| 4.2.2    | mPDF, PHPSpreadsheet                           | 34        |
| 4.2.3    | Swiftmailer                                    | 34        |
| <b>5</b> | <b>Resultados</b>                              | <b>36</b> |
| 5.1      | Tienda pública                                 | 36        |
| 5.1.1    | Proceso de compra                              | 36        |
| 5.1.2    | Exploración de archivos                        | 40        |
| 5.2      | Herramienta de gestión                         | 42        |
| 5.2.1    | Interfaz general y listados                    | 42        |
| 5.2.2    | Informes                                       | 43        |
| 5.1.1    | Terminal punto de venta                        | 43        |
| <b>6</b> | <b>Conclusiones</b>                            | <b>46</b> |
| <b>7</b> | <b>Bibliografía y documentación consultada</b> | <b>48</b> |
| 8.1      | Referencias                                    | 48        |
| 8.2      | Software empleado                              | 48        |
| 8.3      | Documentación                                  | 49        |
| <b>7</b> | <b>Anexos</b>                                  | <b>51</b> |
| 7.1      | Manual de usuario                              | 51        |



# Índice de imágenes

---

|  |    |
|--|----|
| 1. Visualización de un listado con jqGrid  | 25 |
| 2. JSON usado por jqGrid para rellenar los datos de los listados                 | 25 |
| 3. Inicialización del grid de los artículos                                      | 27 |
| 4. Interfaz de Slim Image Cropper para la subida de fotos                        | 28 |
| 5. Código PHP en archivo HTML  | 30 |
| 6. Manejador del evento “Guardar nueva regla de precio”                          | 31 |
| 7. Tratamiento de la llamada AJAX por el servidor para una nueva regla de precio | 32 |
| 8. Página principal de la tienda   | 36 |
| 9. Ficha del artículo  | 37 |
| 10. Carrito de la sesión del invitado  | 38 |
| 11. Formulario de registro   | 39 |
| 12. Formulario de direcciones  | 39 |
| 13. Finalización de compra   | 40 |
| 14. Árbol de la sección textil   | 41 |
| 15. Resultado de la aplicación de los filtros                                    | 41 |
| 16. Menú principal de la herramienta de gestión                                  | 42 |
| 17. Listado de tickets a cliente   | 43 |
| 18. Informe general de ventas  | 44 |
| 19. Interfaz TPV   | 44 |



# 1. Introducción

---

## 1.1 Objetivos del proyecto

---

En la actualidad son muchas las personas que tienen en mente abrir un pequeño negocio. Sin embargo, la gran mayoría de ellas acaban desistiendo ante esa idea debido a, entre otras razones, la complejidad que conlleva la gestión de un negocio. Partiendo de la administración de clientes hasta la emisión de facturas, son muchos los detalles que hay que tener en cuenta para que todo el proceso transcurra de manera adecuada. Además, es esencial que el tiempo que ocupa toda esta gestión se minimice, de hecho, lo ideal es que el emprendedor se olvide completamente de la gestión y pueda dedicar su tiempo a hacer crecer su idea de negocio. No obstante, la realidad es bien distinta y un gran número de negocios acaban fracasando a causa de que sus autores van mermando gravemente su tiempo en detrimento de las tareas burocráticas.

Para solventar esta problemática, muchas personas optan por externalizar este servicio, sin embargo, ello ocasiona un aumento de costes que no todo el mundo se puede permitir. A decir verdad, aunque antes se ha mencionado el tiempo como un recurso muy valioso para el emprendedor, no hay que olvidar que la financiación no deja de ser un recurso igual o más importante. En este sentido, la problemática de la falta de tiempo o la falta de liquidez provocan que al final el emprendedor pierda motivación y acabe abandonando su proyecto. A raíz de esta problemática nace nuestra idea, con la intención de automatizar e informatizar una gran parte de la gestión del negocio, agilizando y haciendo más asequible las tareas administrativas. En síntesis, el usuario de esta herramienta tiene disponible un administrador de clientes, un gestor de facturas y albaranes, un sistema de control del stock e incluso, en el caso de disponer de una tienda física, el acceso a un punto de venta. A pesar de que en el mercado ya existen multitud de aplicaciones que cumplen la función de informatizar la administración de una empresa, nuestro objetivo no es solo el de agilizar estas labores, si no que se pretende ayudar a la expansión del negocio con la inclusión de una tienda web cohesionada con los datos manejados por el gestor. De esta forma, el pequeño negocio que ofertaba sus productos a un nivel local logra alcanzar promocionarse a nivel global con un gasto mínimo.

## 1.2 Estructura seguida

---

A lo largo de este documento se explicará el funcionamiento y las tecnologías usadas para desarrollar el proyecto. Asimismo, será frecuente su división en dos partes: por un lado, se hará referencia al módulo de gestión, relacionado con la herramienta que cumple esta función. Este

módulo es muy extenso y está compuesto por muchos componentes relacionados entre sí, los cuales han requerido una programación elaborada. Por otro lado, se menciona al módulo público haciendo referencia a la tienda web. Esta tienda, salvo en ciertos detalles, ha requerido un nivel de programación menor, ya que su funcionamiento se basa en exponer los artículos dados de alta en la parte de gestión.

En primer lugar, y para situar al lector, se comienza explicando el panorama actual al que se enfrenta un pequeño comerciante que quiere empezar con una tienda web. La mayoría de las opciones presentadas también cuentan con un pequeño gestor para los artículos publicados, sin embargo, ninguna llega al nivel que propone nuestra aplicación. Paralelamente, se ha omitido exponer sistemas de gestión puros, debido a que: o bien son sistemas muy complejos que además de la gestión administrativa manejan conceptos mucho más amplios como la gestión a nivel contable y escapan al ámbito de nuestra aplicación, o son sistemas que sí llevan una gestión administrativa, pero están realizados de manera específica para los clientes.

Una vez expuesto el panorama actual se procede a entrar más en materia, explicando las bases de la que parte nuestra idea. En esta sección se detalla los objetivos que se quieren cumplir. A su vez se muestran una serie de requisitos mínimos que ha de cumplir el proyecto tras ser finalizado. Estos requisitos ayudan a mantener la filosofía de la aplicación, evitando que con el desarrollo de esta se pierda el motivo por el que ha sido creada.

A partir de esta sección se tratan temas más técnicos y más relacionados con lo que es la elaboración del producto. Se explicará las principales tecnologías usadas y en ocasiones se intentará acercar al lector a ellas mediante la exposición de ejemplos reales de partes de la aplicación. Sin embargo se pretende que la inclusión de código sea mínima para no hacer de esta una lectura densa.

Finalmente se mostrará los resultados obtenidos mediante, esta vez sí, capturas de la aplicación. No obstante en esta sección se promociona más el módulo público (tienda web) que la herramienta de gestión. Esto tiene su explicación en que en los anexos se añade el manual de usuario de la herramienta de gestión, que incluye una gran variedad de casos y capturas.





# 2. Antecedentes

---

## 2.1 Alternativas actuales

---

En la actualidad, los pequeños comercios disponen de varios medios para darse a conocer en la Web. Esta forma de negocio, en la que no hay relación física entre las partes, sino que los pedidos, la información, los pagos, etc., se hacen a través de un canal de distribución electrónico es llamada *e-commerce*[1].

Los inicios del *e-commerce* comienzan en los años 60, cuando se permitieron realizar las primeras transacciones electrónicas.[2] Sin embargo, la imagen del eCommerce en la actualidad dista mucho de aquella idea anterior al nacimiento de Internet, dado que el sector ha crecido tanto que se han desarrollado varias herramientas para ayudar al dueño de un negocio a publicar cualquier producto en Internet.

### 2.1.1 CMS (Content Management System)

Sin duda, una de las herramientas más significativas son los sistemas de gestión de contenidos (en inglés: Content Management System, más conocido por sus siglas CMS). Esta afirmación se ve reflejada en los datos aportados en [3] que muestran que más de un 50% de las páginas de Internet son algún tipo de CMS. Aunque este dato hay que refinarlo, mostrando que dentro de los CMS no todos derivan a una actividad comercial en sí, y aunque dentro de esa estadística se muestra que el CMS más utilizado es *Wordpress* y sea conocido por ser principalmente un creador de contenido para blogs, hay que destacar que este ha ido evolucionando con el tiempo y ahora permite crear contenido dedicado para negocios comerciales.

En segundo lugar, la oferta disponible de CMS es muy variada y completa. Además de una tienda online, estas plataformas suelen contar con una biblioteca de plugins que añaden funcionalidad extra a tu sitio web, proporcionando una visión de stock o un punto de venta virtual. La parte negativa de estos añadidos es que suelen ser contenido de pago, con lo que se produce un aumento en el precio final para el usuario. No obstante, otra ventaja que aportan los creadores de contenido es su fácil manejo, puesto que no son necesarios conocimientos altos de informática para gestionar la tienda. Esto se debe a que cuentan con un panel de gestión muy intuitivo para el usuario medio. Desde este panel, el cliente es el encargado de subir artículos nuevos o actualizar precios de los artículos ya subidos entre otras muchas funciones, sin necesidad de contar con un programador web.

## 2.1.2 TIENDA WEB PERSONALIZADA

Previamente al éxito de los CMS, aquellos que querían mostrar sus productos en la Web tenían la posibilidad de crear una tienda online exclusiva para ellos. A pesar de que esta opción todavía es viable hoy en día y algunos comercios aún se decantan por ella, la creación de tiendas personalizadas ha sufrido una gran caída. Esto se debe a que crear una web con estas características requiere mucho tiempo de análisis y desarrollo. Aunque una tienda parece una idea simple, esta va adquiriendo complejidad a medida que añadimos funciones como el registro de usuarios, la posibilidad de filtrar artículos o la gestión de stock.

A sus desventajas hay que añadir que el mantenimiento de estas webs no suele ser sencillo. Realizar un panel de gestión aumenta aún más el tiempo de desarrollo, por lo que cada vez que el cliente quiera hacer algún cambio seguramente va a necesitar ayuda especializada, ocasionando que el presupuesto del proyecto se vea aumentado y generando una relación de dependencia con el equipo de desarrollo de la web, privando así al cliente de la libertad de publicar lo que desee cuando quiera.

Por otro lado, hay que destacar que ninguna otra opción permite tanta personalización como esta. Hoy en día, para llevar a la realidad cualquier idea, lo único necesario es tiempo y es el dueño del negocio quien ha de sopesar si el sobre coste de un desarrollador o un equipo dedicado a él es factible o no.

## 2.1.3 MARKETPLACES

Un marketplace es un concepto desconocido por mucha gente, sin embargo este desconocimiento no se debe a no conocer ninguno, si no a no saber asociar el término “Marketplace”. Si nombramos a Amazon o Ebay, es bastante posible que las dudas anteriores se despejen.

Una definición sencilla sería la de un mercado web donde están invitados tanto compradores como vendedores y donde lo más usual es que el vendedor tenga que pagar una cierta comisión a la plataforma por hacer uso de sus servicios. Tomando a Amazon como el mayor exponente de este sector en el mundo occidental[4], un vendedor tiene que pagar una mensualidad para exponer sus productos. Amazon publicará sus artículos junto los de otros vendedores e incluso con los suyos propios. A pesar de la gran cantidad de negocios y, por lo tanto, la competencia que existe en estas plataformas, no hay que olvidar que una de las grandes ventajas de los marketplaces es la facilidad con la que se accede a un tráfico enorme de potenciales clientes. Amazon recibió cerca de 1.373 millones de visitas anuales en su dominio .es [5], cifra que escapa a cualquier otro método mencionado anteriormente, y a pesar de que la competencia en estos medios puede ser muy alta, las cifras destacan que por término medio los vendedores aumentan sus ventas en un 50%[6].

Con estos datos, los marketplaces parecen la mejor opción y, de hecho, en muchos casos lo son; sin embargo, también tienen ciertos inconvenientes que pueden hacer que no sea la mejor opción: por ejemplo, comercios que se benefician de tener una gran imagen de marca se ven afectados al no vender en una tienda propia, dado que la imagen y la presencia de esa marca en

internet no será tan contundente como tener una tienda propia. Asimismo, debemos tener en cuenta que ello implica utilizar una plataforma de terceros que también vende sus propios productos, es decir, esta plataforma va a tener la posibilidad de promocionar sus propios productos por encima de los tuyos de distintas maneras, como puede ser el posicionamiento de los artículos al retornar resultados de una búsqueda.

## 2.2 Herramientas necesarias

---

El desarrollo web no necesita de grandes herramientas. De hecho es posible la realización de proyectos con solamente un bloc de notas y una terminal para subir archivos al servidor. Sin embargo esta opción además de ser muy rudimentaria no es eficiente, existen herramientas hoy en día que ayudan mucho en la labor del desarrollo web.

Se pueden dividir las herramientas en dos sectores: por un lado aquellas que te ayudan con la creación y edición de código, correspondiente al bloc de notas y por otro la encargada de subir los archivos al servidor, correspondiente a la terminal.

Hay una gran variedad de herramientas pertenecientes al primer sector. El bloc de notas por defecto que viene en los sistemas operativos es una de ellas, pero como se ha dicho antes no es la herramienta más recomendada debido a su sencillez.

En la edición y creación de código se busca un editor de código que proporcione varias características que ayuden a ser más productivo. Entre estas características están por ejemplo: el sangrado automático de las líneas, el autocompletar variables y funciones, el uso de diferentes colores para proporcionar una visión clara del código, etc.

Estas características y muchas más son ofrecidas por diversos programas como SublimeText, Notepad++, Atom...

Concretamente en este proyecto se ha usado SublimeText. Esta es una herramienta que proporciona las anteriores características. Además está acompañada de un gestor de paquetes que permite aumentar más sus funcionalidades. La elección de esta herramienta frente al resto ha sido debido a que ya se ha trabajado con ella en anteriores ocasiones.

Por otro lado debemos de disponer de herramientas para subir los ficheros al servidor. La manera más rudimentaria es mediante comandos de terminal pero es una opción que a día de hoy no es usada debido a la cantidad de programas mucho más amigables, rápidos y visuales que hay. Se ha optado por usar un servidor FTP (File Transfer Protocol), programas que mediante este protocolo hacen transferencias de archivos entre el servidor y el cliente. Las ventajas de FTP como protocolo son varias, pero las más destacables son la rapidez, la seguridad y la cohesión que dispone con la mayoría de servicios web.

Dentro de los servidores FTP se ha escogido FileZilla por ser de los más populares, gratuito y al igual que ocurría con los editores de texto por ya haber sido usado en otros proyectos.

Cabe mencionar que otro método de trabajo es el uso de IDEs (Integrated Development Environment), que son un conjunto de programas más extensos que trabajan entre sí para proporcionar un entorno de trabajo conjunto desde el cual realizar cualquier acción deseada. Los IDEs entre otras funciones engloban los dos apartados anteriores: la creación de código y la

subida de ficheros. Asimismo pueden ofrecer otras funcionalidades como la edición directa en el servidor o proporcionar acceso directo a la base de datos.

Ejemplos de estos programas son Coda o el conjunto de programas de JetBrains como PhpStorm. Sin embargo estos programas además de añadir una barrera de aprendizaje no son de libre uso y requieren la adquisición de una licencia para su utilización.



# 3. Análisis

---

## 3.1 Objetivo de la aplicación

---

Se parte de la idea de la necesidad de una aplicación dedicada específicamente para la gestión y venta de productos a través de Internet. Esta aplicación web está formada por dos módulos principales: el primero de ellos tiene por meta proporcionar una ayuda a la gestión comercial y financiera del negocio. Este módulo tiene que ser accesible desde cualquier localidad. Para ello, utilizará Internet como requisito de conexión a la aplicación y se accederá a esta con unas credenciales, otorgando así un espacio privado y seguro en el que almacenar información del estado del negocio. En lo referente al sector comercial, este debe brindar algún medio para almacenar una cartera de clientes en el que se guarde la información de los contactos, pudiendo hacer ofertas especiales para clientes específicos o para grupos de estos. En relación con esto, otra de las características que ha de cumplir la aplicación es la correcta gestión del stock, la cual podrá llevar una fácil trazabilidad de los artículos, permitiendo saber la cantidad de artículos de los que disponemos en el almacén o qué cliente dispone de qué artículo y cuántos.

Por otro lado, la aplicación ha de permitir llevar un control de las facturas y albaranes correspondientes, así como apoyar a estos con un sistema de devoluciones y pedidos. Se mostrarán listados de todos ellos y se permitirán operaciones CRUD (Create, Read, Update and Delete) sobre estos. Además, se llevará de manera automatizada la numeración de las facturas y se planteará como indispensable que se ofrezca una vista resumen de la facturación, que se podrá mostrar como un informe de ventas. Por último, otra funcionalidad que debe ofrecer la aplicación web es un punto de venta, el cual dispondrá de mucha de la información mencionada anteriormente. Con los datos de los descuentos de los clientes, junto al stock y la generación automática de facturas, esta terminal ha de ser lo suficientemente autónoma como para generar tickets de compra válidos (descuentos aplicados, imposibilidad de vender a falta de stock, numeración correcta de facturas...).

Por otra parte, el segundo módulo ofrece una tienda web que pone a disposición de todo el mundo con acceso a Internet los productos gestionados por el primer módulo. Esta tienda, al igual que el punto de venta, hace uso de los datos almacenados para ofrecer una vista adecuada al cliente. Para que sea personal de cara a él, ha de proporcionar un área privada por cada usuario que se desee registrar. Este mismo usuario dispondrá de un perfil donde guardar sus datos personales para poder realizar pedidos de manera más dinámica y desde su área privada tendrá la posibilidad de ver todas las compras realizadas y el estado de estas, entre otros elementos.

La estructura de la tienda seguirá una jerarquía de categorías basada en Secciones, Familias y Subfamilias en las que se categorizan los artículos. Estas categorías han de poder ser personalizables desde el área privada de gestión de la aplicación, desde donde también se asignará la categoría a la que pertenece cada artículo, conformando así la estructura de la tienda. El procedimiento de búsqueda de los artículos será dual. Por un lado, se ha de permitir una búsqueda directa, ofreciendo como resultado un listado de artículos que coincidan con el término de la búsqueda ya sea por nombre, categoría..., etc. Por otro, se permitirá ir explorando el árbol jerárquico en el que se estructuran los artículos. El listado de artículos devueltos se podrá especificar aún más con la aplicación de filtros, los cuales, como las categorías, son definidos en la parte de gestión y definirán atributos de los artículos a vender como podrían ser la talla o el color de una prenda de ropa.

## 3.2 Requisitos mínimos

---

Dentro de las funcionalidades de la aplicación serán requisitos mínimos y esenciales los apartados siguientes.

### 3.2.1 ARTÍCULOS

Sección donde se podrá dar de alta un artículo nuevo. Un artículo se definirá a través de sus detalles. La ficha del artículo cuenta con varios niveles de información:

- Información general: nombre, categoría a la que pertenece, detalle del artículo...
- Fotos: fotos del producto que luego aparecerán en la tienda web.
- Ficheros: ficheros con documentación relevante al artículo (manuales de usuario, folletos de la marca...).
- Modelos: cuando un atributo recibe un valor (ej. rojo) puede ser asociado a un modelo. Un modelo es una especificación de un artículo, esta definición puede llevar a equívocos y, puesto que es un concepto muy importante de la aplicación, se intentará aclarar con el siguiente ejemplo: En nuestra base de datos tenemos un artículo con id:'003' de nombre 'Camiseta de fútbol' con dos atributos: 'Equipo' y 'Talla'. Los modelos de este artículo pasan a estar definido con la especificación de sus atributos. En nuestro almacén podríamos tener un modelo que fuera {Nombre: 'Camiseta de fútbol', Equipo: 'Valencia', Talla: 'S' }.
- Reglas de precios: lugar donde se añadirán reglas de precios para clientes específicos o para grupos de clientes. Las reglas se han de poder definir tanto sobre artículos como sobre modelos específicos.
- Stock: stock disponible de ese artículo en nuestros almacenes.
- Pedidos a proveedor: espacio con la información de los pedidos a proveedores del artículo.
- Clientes: espacio con información de quién y cuándo ha comprado ese artículo.

·Atributos: los atributos son características que definen un artículo (ej. talla, color...) .

### **3.2.2 PEDIDOS DE CLIENTE, TICKETS, FACTURAS Y ALBARANES**

La aplicación contará con una vista para cada tipo de documentos. Dentro de esta vista serán posibles operaciones CRUD sobre los documentos, pero siempre respetando la numeración. En el caso concreto de las facturas no será posible su eliminación, excepto si es la última creada, es decir: si se desea borrar alguna factura ha de ser mediante la creación de una factura rectificativa y si aparecen marcada como pagada no será posible la adición de más artículos.

Además de las operaciones de actualización, creación y eliminación, desde cada vista se ha de poder imprimir el tipo de documento, y si se trata de un pedido, marcarlo como pagado. También es necesaria la posibilidad de filtrar entre fechas y/o entre tiendas.

Asimismo, en los listados se ha de poder acceder al detalle del tipo de documento, mostrando la información relevante a estos, como puede ser: la numeración, el cliente, la forma de pago, la tienda donde se efectúa el movimiento, la fecha de creación, el listado de artículos y las observaciones pertinentes.

### **3.2.3 TPV**

Esta sección será una de las más relevantes de la aplicación, puesto que será el punto de entrada principal de los beneficios de las ventas. La información mostrada en el punto de venta dependerá en función del usuario que la use, ya que cada terminal usará los datos de la tienda a la que el usuario pertenezca. Esto será así para que las ventas que se hagan sean correctas: impedimento de venta de artículos sin stock, numeración asociada a la tienda, datos de la tienda mostrados en el ticket correctos, etc.

El punto de venta contará con dos formas de introducir artículos al ticket, una de ellas mediante un panel de búsqueda, el cual devuelve los artículos coincidentes con las palabras clave, indicando también el stock disponible en esa tienda. La otra forma de añadir productos al ticket será de manera directa, por medio de la lectura del código de barras del artículo. Por lo tanto, se ha de añadir soporte para escáneres de este tipo.

La TPV ha de estar pensada, tanto para clientes de nuestra cartera ya registrados como para personas ajenas a esta, y ofrecerá de manera automática los descuentos definidos en la ficha del cliente.

A la hora de añadir un artículo al ticket se ha de poder seleccionar el número de unidades que se quieran, para evitar tener que introducir varias veces el mismo artículo/código de barras. Para artículos únicos, con número de serie y de los que se quiera seguir una trazabilidad más detallada, se ha de ofrecer la posibilidad de, una vez introducido el artículo en el ticket, poder asociar un número de serie.



El operador de caja, además de las opciones de modificar cantidades y eliminar líneas de ticket, ha de poder ofrecer al cliente un descuento. En este sentido, el máximo descuento aplicado estará limitado por lo indicado en la ficha del usuario en la parte de gestión.

Por último, desde la terminal será posible realizar tanto cobros como albaranes. Cuando se realice una venta se ha de ofrecer la posibilidad de pagar con tarjeta, efectivo o mediante una mezcla de estos dos, dividiendo el importe del ticket entre estas dos formas de pago. Además, por cada pago se elegirá si el cliente quiere que se le haga factura o no.

### **3.2.4 CONTROL DE STOCK**

La aplicación ha de incluir una sección que permita al usuario tener una visión clara del stock disponible. Se ha de poder controlar todo el material que gestiona el negocio, esto incluye el propio almacén, los pedidos a proveedores y los envíos a clientes. Este control de stock será uno de los datos a utilizar cuando se generen informes, por tanto, es muy interesante tener la posibilidad de saber el importe total en bienes disponibles en el almacén. Otra vista que nos proporcionará será la del stock disponible en los clientes, permitiendo un mayor control de los productos enviados, pudiendo discernir entre los productos ya facturados o aquellos que solamente han sido enviados.

### **3.2.5 CLIENTES Y PROVEEDORES**

Se requiere que la aplicación contenga un ligero gestor de clientes y proveedores. No es necesario ofrecer todas las funcionalidades que ofertan los CRMs (siglas) puros, pero sí que se requiere un cierto control de los clientes. Este control se refiere a poder almacenar ciertos datos personales como su nombre, teléfono, email, forma de pago usual, una o varias direcciones postales y un campo de texto libre en el que poder realizar anotaciones referentes a ese cliente. Además de estos datos personales, esta sección ha de proporcionar una vista con todos los pedidos y facturas que este cliente ha realizado.

### **3.2.6 FLUJO DE DINERO**

Desde esta sección se visualizará cada movimiento de capital efectuado en la tienda o en la TPV virtual y será en la medida de lo posible automatizada. Cada vez que se realice una venta, se generará una nueva entrada en esta tabla con el origen de la venta, su fecha, el usuario que la realizó, el método de pago usado y el importe total de la venta. Entradas parecidas a estas han de introducirse también con las devoluciones. En lo concerniente a los movimientos manuales de efectivo, como la retirada de este de las TPVs, se ha de dar la opción de crear entradas también de forma manual, así como editar las ya creadas o eliminarlas.

### **3.2.7 INFORMES**

Todos los datos generados por el uso de la aplicación han de estar disponibles de una manera cómoda para poder proceder a su análisis. Para lograr esto, se ha de proporcionar una serie de informes basados en estos datos. Los informes necesarios serán:

- Un informe completo de todas las ventas hechas. Antes de generar este informe se ha de proporcionar una serie de filtros para poder acotar la búsqueda, permitiendo elegir el rango de fechas en la cual se ha realizado la venta, el empleado que la ha realizado o el tipo de artículo que se ha vendido. En este informe figurarán los filtros empleados, el número de ventas y los detalles de cada venta (importe, fecha, cliente, vendedor, número de factura, artículo...).

- Un parte del día. Una vez seleccionada la tienda y el día, se visualizará un informe esquemático con el desglose de las ventas del día por secciones, así como el total vendido. También en este informe se proporcionará un lugar donde anotar a mano tareas pendientes o sugerencias y quejas de los clientes.

- Ventas por tienda. A partir de la selección de la tienda y un rango de fechas se generará un informe con las ventas realizadas en ese establecimiento. El informe debe incluir forma de pago, concepto, cantidad, fecha, precio, descuento, IVA..., de los productos vendidos. Además, este informe ha de estar disponible para su descarga en dos formatos, uno de ellos que permita la edición de los datos y otro que solo permita la visualización de estos. Ejemplos de estos dos formatos serían una hoja de cálculo y un archivo PDF.

- Informe de stock. Generará un informe con el stock disponible en el almacén.

### **3.2.8 TIENDA WEB**

El módulo de gestión contendrá un apartado desde el cual se controlarán ciertos aspectos de la tienda web pública. En este apartado se permite modificar el logotipo que se muestra en la web, los enlaces a las redes sociales que aparecen en la página inicial de la tienda, elegir la publicidad, imágenes y enlaces que se muestran en esta, y las fotos que se visualizarán en el slider de la página inicial de la tienda.



# 4. Diseño y metodología del desarrollo

---

En este apartado se intentará explicar de la forma más sencilla posible los aspectos técnicos de la aplicación y el porqué del uso de una tecnología específica frente a las otras disponibles. No todos los métodos utilizados son los más adecuados sobre el papel, sin embargo, debido a que la aplicación es un proyecto que definitivamente se ha llevado a su ejecución, hay que entrar a valorar las limitaciones que conlleva, como por ejemplos las librerías propietarias o recursos que no permiten su uso comercial.

Esta sección se estructura en dos apartados principales, los cuales dividen la aplicación desde un punto de vista arquitectónico: por un lado se expondrán las tecnologías que corren de lado del navegador web, conocidas como tecnologías frontend, que tienen la función de mostrar de la manera más adecuada posible los datos al usuario final. A continuación se muestran aquellas tecnologías que nuestra aplicación usa del lado del servidor, denominadas como tecnologías backend, las encargadas del tratamiento real de los datos. Aquí se incluye, por ejemplo, las tecnologías usadas para el procesamiento de los datos y su almacenamiento en la base de datos.

## 4.1 Tecnologías frontend

---

Como se ha explicado anteriormente, el frontend es la parte del software que interactúa con el usuario final. El frontend hace uso de las tecnologías, lenguajes de estilo o programación del lado del cliente para la estructuración, el maquetado y la animación de las aplicaciones web. Con esta definición se puede apreciar que la parte visual de la aplicación y con la que más se familiariza el cliente es con estas tecnologías.

### 4.1.1 HTML, CSS, JAVASCRIPT

Actualmente la tecnología más usada en este sector es la tríada de HTML (HyperText Markup Language), CSS (Cascading Stylesheets) y Javascript. Estos tres lenguajes dan la posibilidad de hacer prácticamente cualquier diseño, efecto o interfaz necesaria.

Como explicación breve se puede definir al HTML como el lenguaje usado para dotar de una estructura a la aplicación. Se basa en etiquetas y es posible su uso en solitario, sin embargo, su uso colaborativo con el CSS y el Javascript permite hacer webs más dinámicas y atractivas visualmente.

El CSS es el encargado de aplicar estilos a las estructuras creadas con el HTML. Se puede definir como la capa de personalización de la aplicación. Esta personalización se aplica

vinculando el HTML con el CSS y el abanico de personalización que ofrece es tan amplio que el mismo fichero HTML vinculado con dos CSSs diferentes puede generar un resultado completamente distinto. Ambos lenguajes son lenguajes de etiquetas que parten de una base muy simple pero que pueden llegar a tener estructuras muy complejas.

Mientras que los dos lenguajes anteriormente mencionados se encargan de dotar una estructura y un estilo a la aplicación, Javascript dota de dinamismo a esta. Es un lenguaje mucho más potente y complejo que los anteriores, tanto que incluso se está abriendo camino fuera del mundo Web[7]. Sin embargo, debido a este crecimiento, rara es la vez en que se ve el uso de Javascript en solitario en una aplicación web. Al ser un lenguaje tan extenso, se suele optar por el uso de alguna librería más enfocada al mundo web. Estas librerías o frameworks utilizan Javascript como base pero hacen una mejor aproximación al mundo web, es decir, las funcionalidades que ofrecen estas librerías son posibles también desde JavaScript, pero se consiguen de una manera más sencilla y rápida. Ejemplos de estos son jQuery o AngularJS. Concretamente en este proyecto se usa Javascript con jQuery.

#### **4.1.2 jQuery, jQueryUI, jqGrid**

jQuery es una biblioteca de JavaScript que ayuda mucho en el momento de interactuar con archivos HTML, manejo de eventos, manipulación del DOM (Document Object Model) y es la librería de Javascript más usada actualmente[8].

La funcionalidad del proyecto no se habría visto muy afectada sin el uso de jQuery, sin embargo, los tiempos de desarrollo han sido enormemente mermados gracias a la simplicidad en la sintaxis de esta. La posibilidad de acceder a cualquier elemento del DOM con solamente saber su identificador y desde ahí poder modificar parámetros o llamar a eventos directamente, hace que esta librería sea esencial en términos de productividad. Agregar que esta biblioteca se puede complementar con jQueryUI, una librería desarrollada para dar estilos a los componentes de jQuery y que también es usada en el proyecto.

Todas las tecnologías nombradas anteriormente son comúnmente usadas en el mundo web, quizás el uso de jQuery está empezando a resentirse un poco debido a la aparición de bibliotecas más modernas que mejoran el diseño adaptativo para vistas móviles, como puede ser el caso de Ionic[9].

No obstante, la siguiente librería sí está más relacionada con los objetivos que se intentan alcanzar con nuestra aplicación, y es que, analizando los requerimientos de esta y el uso que se le va a dar, se observa que la mayoría de vistas son listados con una serie de elementos sobre los que hacer operaciones de inserción, eliminación y edición: es el caso de los albaranes, clientes, facturas, artículos... Esta librería es jqGrid, una biblioteca de jQuery enfocada principalmente al tratamiento y visualización de los datos en tablas.

JqGrid ofrece varias formas de funcionamiento y obtención de los datos, sin embargo en este documento sólo se recogerán los modos usados en el proyecto, puesto que la explicación de esta librería no es el objetivo principal de este escrito.

| Tienda   | Fecha               | Usuario | Tipo               | Cantidad | Observaciones |
|----------|---------------------|---------|--------------------|----------|---------------|
| CENTRAL  | 02-11-2017 14:00:19 | hugo    | Venta TARJETA VISA | 24.20€   | TICKET TPV    |
| CENTRAL  | 22-03-2018 10:47:35 | hugo    | Venta CONTADO      | 7.26€    | TICKET TPV    |
| CENTRAL  | 20-03-2018 08:48:32 | hugo    | Venta CONTADO      | 302.50€  | TICKET TPV    |
| CENTRAL  | 20-03-2018 08:45:35 | hugo    | Venta CONTADO      | 302.50€  | TICKET TPV    |
| VALENCIA | 20-03-2018 08:33:22 | hugo    | Venta TARJETA VISA | 72.00€   | TICKET TPV    |
| CENTRAL  | 01-03-2018 08:45:19 | hugo    | Venta CONTADO      | 24.20€   | TICKET TPV    |

Imagen 1. Visualización de un listado con jqGrid

Mostrar una tabla en pantalla consta de dos pasos, cargarla en memoria y rellenarla con los datos.

El medio usado para obtener los datos del servidor es mediante una llamada a un PHP que genera un JSON (JavaScript Object Notation). Este JSON es un array multidimensional de los elementos que contiene la tabla como se puede ver en la imagen 2.

```

{page: "1", total: 1, records: "7", rows: [{id: 0,...},...]}
  page: "1"
  records: "7"
  rows: [{id: 0,...},...]
    0: {id: 0,...}
      cell: ["35739", "39", "BOLAS", "213", "CALLAWAY", "0", null, "BOLAS CALLAWAY HEX ", "41", "1",...]
        0: "35739"
        1: "39"
        2: "BOLAS"
        3: "213"
        4: "CALLAWAY"
        5: "0"
        6: null
        7: "BOLAS CALLAWAY HEX "
        8: "41"
        9: "1"
        10: "049caafd99d047894e269bdb43039b48"
      id: 0
    1: {id: 1, cell: ["36285", "3", "CALZADO", "206", "HOMBRE", "336", "NIKE", "ZAPATOS NIKE", 0, "1",...]}
    2: {id: 2, cell: ["36291", "44", "SERVICIOS", "0", null, "0", null, "Clases Golf", 0, "1",...]}

```

Imagen 2. JSON usado por jqGrid para rellenar los datos de los listados

Los datos devueltos por el JSON no son solo los datos a mostrar en la tabla, si no que también contienen otros campos con información adicional como puede ser la cantidad de registros devueltos, datos utilizados para completar la información mostrada al usuario. Como se ha detallado anteriormente, esta no es la única forma de obtener la información del servidor: otros formatos como XML son aceptados y la carga local de datos también está permitida. Para cargar los datos con del modo mostrado se especifican una serie de parámetros a la hora de definir la tabla, como son la URL en la que se encuentra el fichero PHP encargado de hacer la

consulta a la base de datos o el formato de datos devuelto (JSON). Además, en nuestro caso, para simplificar la consulta establecemos el parámetro *loadonce* a *TRUE* y así solo se accede al servidor una única vez ocasionando que los filtros de búsqueda y ordenación se apliquen a los datos locales evitando hacer llamadas al servidor cada vez que se manipulen los datos. Esto ocasiona también que el fichero PHP, encargado de devolver el JSON, se simplifique, puesto que no se le envía al servidor los filtros usados ni el método de ordenación. Sin embargo, sí que podemos añadir al fichero parámetros propios para realizar una búsqueda más certera: es el caso, por ejemplo, de querer obtener líneas de un albarán. El archivo utilizado para recuperar las líneas de los distintos albaranes es el mismo pero al que se le añade a la dirección el id del albarán deseado. El PHP recoge esta variable mediante el array *\$\_POST* y realiza la consulta pertinente.

Cada elemento que se desea mostrar en la tabla se empaqueta en otro array, el cual contiene tantos índices como columnas la tabla. La diferencia entre número de índices y número de columnas puede ocasionar errores de ejecución, por eso es una buena práctica añadir campos vacíos antes que devolver un array de menor tamaño.

Una vez definido el archivo que nos proveerá de los datos para rellenar la tabla podemos proceder a la definición de esta. La definición de la tabla se hace con una llamada dentro del *\$(document).ready()* establecido por jQuery. Lo primero que hace la función es vaciar la tabla para evitar solapar datos cuando se vuelva a llamar a la función. Tras despejar el contenido de la tabla, ya se llama al constructor de esta. La forma de construir la tabla viene definida por los siguientes parámetros:

- url*: el archivo que nos devuelve los datos. En este caso se puede ver como a la url se le concatena el identificador de la tienda para que la consulta solamente devuelva los artículos asociados a esta.

- datatype*: el formato en el que se devolverán los datos

- colNames*: los nombres que se mostrarán en la cabecera de las columnas

- colModel*: array que contiene diversa información sobre cada columna. Cada elemento del array corresponde a una columna y han de ser tantos como elementos haya en *colNames*. Para cada columna se definen una serie de atributos como su identificador, estilo asociado a los datos alojados, ancho de columna... Un atributo muy recurrente es el atributo *hidden* con valor *TRUE*, que no muestra la columna de manera visual pero sigue permitiendo asociar datos. Es muy usado para identificadores, por ejemplo en el caso de la imagen 3 no nos interesa mostrar el identificador de la sección pero sí su nombre, sin embargo es muy útil tener este identificador para poder hacer manipulaciones directamente sin tener que volver a rehacer la consulta.

- rowNum*, *rowTotal*: número de registros por página y número de registros totales respectivamente.

- loadonce*: como se ha explicado anteriormente dependiendo de su valor hace las modificaciones en local o vuelve a llamar al servidor.

- sortname*, *sortorder*: nombre de la columna por la que se ordena la tabla por defecto y su modo de ordenación

- caption*: título que se asocia a la tabla

- width*, *height*: ancho y altura de la tabla

```

// #####
// CARGAR EL GRID ...
// #####
function CargaGRID(){
    jQuery("#tbl").jqGrid("GridUnload"); //Realizo el Unload para poder volver a llamar a la carga con loadonce...

    jQuery("#tbl").jqGrid({
        url:'inc/articulos.json.php?IdTienda=' + $("#txtIdTienda").val(),
        datatype: "json",
        colNames:['Id', 'IdSeccion', $("#txtSeccionGrid").val(), 'IdFamilia', $("#txtFamiliaGrid").val(), 'IdSubFamilia',
            $("#txtSubFamiliaGrid").val(), "txtNombreGrid").val(), $("#txtStockGrid").val(), $("#txtActivoGrid").val(), 'CRC'],
        colModel:[{name:'IdArticulo',index:'IdArticulo', width:50},
            {name:'IdSeccion',index:'IdSeccion', hidden:true},
            {name:'Seccion',index:'Seccion', width:160, align:"left"},
            {name:'IdFamilia',index:'IdFamilia', hidden:true},
            {name:'Familia',index:'Familia', width:160, align:"left"},
            {name:'IdSubFamilia',index:'IdSubFamilia', hidden:true},
            {name:'SubFamilia',index:'SubFamilia', width:160, align:"left"},
            {name:'Nombre',index:'Nombre', width:200, align:"left"},
            {name:'Stock',index:'Stock', width:60, align:"left"},
            {name:'Activo',index:'Activo', width:40, align:"center", formatter:"checkbox"},
            {name:'CRC',index:'CRC', hidden:true}
        ],

        rowNum:100,
        rowTotal: 2000,
        rowList:[],
        loadonce:true,
        mtype: "GET",
        gridView: true,
        pager: '#pager',
        sortname: 'Nombre',
        viewrecords: true,
        sortOrder: "asc",
        caption:"",
        ignoreCase:true,
        width:980,
        shrinkToFit:false,
        height:h,
        beforeRequest:function(){
            var searchFiler = JSON.parse(sessionStorage.getItem("filtrosArticulos"));
            if(searchFiler != null){
                var grid = $("#tbl"), f;
                if (searchFiler.length === 0) {
                    grid[0].p.search = false;
                    $.extend(grid[0].p.postData,{filters:""});
                }
                f = {groupOp:"AND",rules:[]};
                f.rules = searchFiler;
                grid[0].p.search = true;
                $.extend(grid[0].p.postData,{filters:JSON.stringify(f)});
                $.extend(grid[0].p.postData,{_search:true});
            }
        },
        loadComplete:function(){
            var $this =$("#tbl");
            postfilt = $this.jqGrid('getGridParam','postData').filters;
            if ($this.jqGrid("getGridParam", "datatype") === "json") {
                setTimeout(function () {
                    $this.jqGrid("setGridParam", {
                        datatype: "local",
                        postData: {filters: postfilt},
                        search: true
                    });
                    $this.trigger("reloadGrid");
                }, 50);
            }
            $("#tr.jqgrow:odd").css("background", "rgba(0, 0, 0, 0.03)");
        },
        ondblClickRow:function(rowid){$("#btnEditar").click();}
    })

    jQuery("#tbl").jqGrid('navGrid','#pager',{edit:false,add:false,del:false,search:false,refresh:false});
    jQuery("#tbl").jqGrid('filterToolbar',{stringResult:true, searchOnEnter:false, defaultSearch:'cn',
        beforeSearch:function(){
            var filters = JSON.parse($("#tbl").getGridParam('postData').filters);
            var rules = filters.rules;
            sessionStorage.setItem("filtrosArticulos",JSON.stringify(rules));
        }});
    //OCULTO EL BOTON DE PLEGAR DESPLEGAR EL GRID ...
    $("#a.ui-jqgrid-titlebar-close.HeaderButton").hide()
}

```

Imagen 3. Inicialización del grid de los artículos



Estos parámetros definen la estructura de la tabla, pero no su comportamiento, por eso también se facilitan una serie de eventos a los que asociar acciones. En este caso se puede apreciar cómo se asocian acciones a diferentes eventos. Nombrar los diferentes tipos de eventos como:

- beforeRequest*: ejecutado antes de llamar al fichero que nos devuelve los datos
- loadComplete*: ejecutado cuando la tabla termina la carga de datos
- ondblClickRow*: ejecutado cuando se hace doble click en una fila y normalmente asociado al evento del botón editar en nuestra aplicación.

Esto son solo unos pocos ejemplos de los ofrecidos en la API de jqGrid.

Posteriormente a la inicialización de la tabla se procede a inicializar tres complementos de esta: el pie de la tabla (footer), el elemento que nos permite desplazarnos por las distintas páginas (navGrid) y la barra de filtros(filterToolbar). Todos ellos con un amplio nivel de personalización.

### 4.1.3 Slim Image Cropper

Slim Image Cropper, de aquí en adelante simplemente Slim, es un plugin de subida de fotos. A pesar de que actualmente HTML5 ofrece opciones bastante sencillas para la subida de fotos, este plugin permite múltiples opciones como rotar la imagen o recortarla. Además ofrece una interfaz muy clara y visualmente atractiva para el usuario final.



Imagen 4. Interfaz de Slim Image Cropper para la subida de fotos.

Desde el punto de vista del desarrollador web la subida de archivos, con su posterior tratamiento en el servidor y su vinculación en la base de datos puede ser algo ardua, sin embargo, Slim intenta simplificar este proceso en la medida de lo posible.

El proceso completo se compone de tres pasos: definir en qué lugar de la aplicación el usuario sube las imágenes, crear las acciones a ejecutar en el cliente y finalmente hacer las acciones en el servidor para que los cambios sean persistentes.

El primer paso se realiza en el archivo HTML desde donde se desea realizar la subida creando un `<div>` con atributos personalizados `data`. Los más importantes son el `data-service` en el que se incluirá la ruta donde se encuentra alojada la librería y el atributo `data-did-upload` que referencia en nuestro caso a una función en Javascript que se ejecuta una vez se ha subido la imagen. Como ocurre con jqGrid esta librería también permite pasar parámetros al lado del servidor para que pueda tomar las acciones oportunas. En nuestro caso vuelve a ser el identificador de la tienda y el identificador del artículo, necesarios para crear la estructura de directorios correctamente en el servidor, las fotos de cada artículo se guardan en un subdirectorío que lleva por nombre el identificador del artículo.

Esta configuración se determina en el archivo definido en `data-service`. Además de nombrar correctamente los directorios se encarga de que tenga los permisos de escritura y lectura correctos.

Con estas acciones se consigue subir las fotos al servidor, sin embargo el sistema sería muy dependiente de la jerarquía de ficheros para saber si un artículo dispone de imágenes o no, por eso esta información también es almacenada en la base de datos. La función referenciada desde `data-did-upload` es la que se encarga de actualizar la base de datos mediante una llamada AJAX (Asynchronous JavaScript And XML) a un PHP. Finalmente, cuando se recibe una respuesta exitosa se actualiza visualmente la foto para el usuario.

## 4.2 Tecnologías backend

---

El backend es la sección que se encarga de hacer que todo lo que está detrás de la aplicación funcione correctamente y aunque en la actualidad el frontend también se ve algo implicado en esta función, el peso que representa es mucho menor y viene justificado principalmente para evitar cargas en el servidor o excesivo consumo del ancho de banda.

Por lo tanto, las tecnologías backend son las encargadas del procesamiento de la información, las consultas a las bases de datos o la conexión con el servidor entre otras tareas.

### 4.2.1 PHP, MySQL

PHP es un lenguaje de programación que actúa del lado del servidor mientras que MySQL es un sistema de gestión de bases de datos relacional. Esta pareja trabaja conjuntamente para hacer que los datos de la aplicación sean persistentes y accesibles para el usuario.

La aplicación hace uso del lenguaje PHP principalmente de tres formas diferentes.

La primera de ellas es en la carga inicial de las páginas, se combina con el HTML para, por ejemplo, rellenar etiquetas `<select>` con la información adecuada. Esta forma de funcionamiento ayuda a crear una página con datos personalizados ya guardados en la base de datos. Como se ve en la imagen 5 la inclusión de código PHP se hace entre los caracteres “`<?php`” y “`?>`”. En el fragmento de código se ven dos usos diferentes, en la línea 397 se utiliza para llamar a la función `dicc()`. Esta función es muy recurrente a lo largo de la aplicación puesto que es la encargada de traducir todas la etiquetas que aparecen al lenguaje definido por el usuario.

El texto a traducir se le pasa en los parámetros, hace una llamada a la base de datos y retorna la etiqueta en el idioma correspondiente. Para lograr este funcionamiento en la base de datos hay una tabla en la que cada fila es el término que deseamos traducir y sus columnas son sus valores en diferentes idiomas. El idioma escogido está predefinido en las variables de sesión del usuario, que a su vez están definidas por los valores dados en el perfil del usuario dentro de la aplicación.

```
397 <td><label><?php echo dicc("IVA"); ?></label></td>
398 <td class="tdSi">
399     <select id="cmbIVA" name="cmbIVA">
400         <?php
401             include('inc/conexion.php');
402             $sql = "SELECT CodIVA, IVA, Predet FROM IVA WHERE IdTienda='".$_SESSION['IdTienda']."' ";
403             foreach ($conn->query($sql) as $aux){
404                 $sel = ""; if($row["CodIVA"]==$aux["CodIVA"]){$sel = ' selected="selected"';}
405                 echo '<option value="'.$aux["CodIVA"]."' '.$sel.'>'.$aux["IVA"].'</option>';
406             }
407             $conn=null;
408             ?>
409     </select>
410 </td>
```

Imagen 5. Código PHP en archivo HTML

En el mismo fragmento de código se ve otro uso más usual del PHP en ficheros HTML, que es la inicialización de elementos. Lo primero que se hace en estos casos es realizar la conexión a la base de datos mediante la inclusión del fichero de la conexión. Una vez la conexión está definida se realiza la consulta, en este caso se le asocia una variable para que nos devuelva el IVA que pertenece a esa tienda en concreto. Posteriormente se itera sobre las filas devueltas y se vuelve en el HTML el texto deseado con la función `echo` propia de PHP. La línea 404 hace uso de una estructura de control para decidir qué opción es la que se enseñará por defecto al usuario. En este caso se compara con el valor correspondiente al artículo que se está editando obtenido anteriormente.

El uso del lenguaje PHP explicado anteriormente es correcto a la hora de la carga inicial, pero no nos permite guardar la información de manera persistente, sólo nos permite acceder a ella. La siguiente forma de uso de este lenguaje sí lo permite. Además en la aplicación queremos que se guarden los cambios y se muestren directamente al usuario. Para alcanzar este objetivo vamos hacer uso conjunto del PHP con llamadas AJAX. Lo que nos permite las llamadas AJAX es mostrar al usuario los cambios sin que este tenga que refrescar la página. Este método es el más usado dentro de la aplicación, todos los listados de facturas, clientes, artículos... hacen uso de este método para dar altas nuevas, modificaciones o eliminaciones, y todos siguen la misma estructura.

Lo primero que se ha de realizar es localizar el manejador del evento para insertar la llamada AJAX. Después realizamos las comprobaciones que consideremos necesarias, como que esté un descuento aplicado o un cliente seleccionado y si no se cumplen paramos la ejecución del manejador de evento y elevamos un error llamando a la función *Error()* propia. Esta función lleva asociado el texto de error que se le muestra al usuario y un número que corresponde con la etiqueta de error que se mostrará. Posteriormente creamos una cadena de texto que contiene la información que se va a almacenar en la base de datos. En la imagen 6 se puede ver como se desea crear una nueva regla de precio y la cadena de texto se compone por los campos “Op” , “IdArticulo” y “Token” entre otros con un valor asociado. El campo “Op” se refiere a operación y sirve como punto de entrada en el PHP para saber qué función ejecutar. El campo “Token” también es un campo especial y añade una capa de seguridad. Este campo es una cadena de texto autogenerada con el id de sesión del usuario, la fecha actual y una contraseña. El PHP genera este valor y si ambos coinciden permite su ejecución. El objetivo de esta comprobación es que no se pueda llamar al servidor desde fuentes desconocidas.

```
// #####
// Guardar NUEVA REGLA DE PRECIO ...
// #####
$("#aGuardar1").click(function(){
    //Hay que seleccionar un cliente o un grupo de clientes
    if($("#txtIdCliente1").val()=="" && $("#cmbGrupos1").val()=="0"){
        Error("1", $("#txtIntroCliente").val());    return false;
    }

    //No se puede seleccionar ambas cosas
    if($("#txtIdCliente1").val()!="" && $("#cmbGrupos1").val()!="0"){
        Error("1", $("#txtSoloUnTipo").val());    return false;
    }

    var Descuento = parseFloat($("#txtDescuento1").val());

    //Hay que indicar un descuento o un precio
    if(Descuento==0){
        Error("1", $("#txtIntroDesc").val());    return false;
    }

    // DATASTRING
    var dataString = "Op=NUEVA REGLA&IdArticulo=" + $("#txtIdArticulo").val();
    dataString = dataString + "&IdCliente=" + $("#txtIdCliente1").val();
    dataString = dataString + "&IdGrupo=" + $("#cmbGrupos1").val();
    dataString = dataString + "&Descuento=" + $("#txtDescuento1").val();
    dataString = dataString + "&CB=" + $("#txtCodBarrasRegla").val();
    dataString = dataString + "&Token=" + $("#txtToken").val().replace(",",".");

    $.ajax({
        async:"false", type:"POST", url: "inc/articulos.ficha.reglas.bd.php", data: dataString,
        success: function(data){
            data=$.trim(data);
            if(data=="TOKEN"){Error("1", $("#txtProblemaSeguridad").val());}
            if(data=="EXISTE"){Error("1", $("#txtExisteRegla").val());}
            if(data=="OK"){$("#dialog_Nueva_Regla").dialog("close"); CargaGRID_Precios();}
        } //FIN DEL SUCCESS
    }); // FIN DEL AJAX
}); //aGuardar1
```

Imagen 6. Manejador del evento “Guardar nueva regla de precio”

Una vez generada la cadena de texto se procede a la llamada AJAX, que será servida al fichero especificado en el parámetro “url” acompañado de la cadena mencionada. Cuando el servidor responda se actúa en consecuencia. En el caso ejemplo el servidor puede responder de tres formas: “TOKEN”, “EXISTE” o “OK” las cuales manejaremos de la manera adecuada.

Las operaciones que ocurren del lado del servidor las trata el fichero llamado y se observan en la imagen 7. Estos ficheros siempre llevan una estructura parecida. Primero se inicia la sesión, después se comprueba que la llamada contenga el “token” correcto y entonces se procede a evaluar el campo “Op” para saber qué operación ejecutar. Las sentencias a ejecutar dentro de cada operación suelen estar siempre relacionadas con consultas a la base de datos, ya sea de selección, modificación o eliminación. En la imagen 7 se puede ver las sentencias realizadas para crear una nueva regla de precio. Destacar que los datos incluidos en la cadena de texto en la llamada AJAX son accesibles mediante el array \$\_POST.

```
// #####
// NUEVA_REGLA
// #####
if($_POST["Op"]=="NUEVA_REGLA"){

    // Hay que comprobar que no existe ya en la Tabla
    $sql = "SELECT COUNT(*) AS Cuenta FROM ARTICULOS_PRECIOS ";
    $sql = $sql . "WHERE IdArticulo=".$_POST["IdArticulo"]." ";
    if($_POST["IdCliente"]!=""){ $sql = $sql . "AND IdCliente=".$_POST["IdCliente"]." ";}
    if($_POST["IdGrupo"]!=""){ $sql = $sql . "AND IdGrupo=".$_POST["IdGrupo"]." ";}
    if($_POST["CB"]!=""){ $sql = $sql . "AND CB=".$_POST["CB"]." ";}
    $row = $conn->query($sql)->fetch();

    if($row["Cuenta"]>0){echo "EXISTE"; exit;}

    // INSERT
    $sql = "INSERT INTO ARTICULOS_PRECIOS(IdArticulo, CB, IdCliente, IdGrupo, Descuento) ";
    $sql = $sql . "VALUES(:IdArticulo, :CB, :IdCliente, :IdGrupo, :Descuento) ";
    $q1 = $conn->prepare($sql);
    $q1 -> execute(array('IdArticulo'=>$_POST["IdArticulo"],
        'CB'=>$_POST['CB'],
        'IdCliente'=>$_POST["IdCliente"],
        'IdGrupo'=>$_POST["IdGrupo"],
        'Descuento'=>$_POST["Descuento"]));

    $Respuesta = "OK";
} // NUEVA_REGLA
```

Imagen 7. Tratamiento de la llamada AJAX por el servidor para una nueva regla de precio

Para añadir seguridad y un control de acceso a la aplicación hacemos uso del lenguaje PHP de un último modo. Para acceder a cualquier página privada, ya sea dentro de la parte de gestión, como en el panel de usuario de la tienda pública hacemos uso de las variables de sesión de PHP.

La primera vez que se accede se obliga al usuario a identificarse, esa identificación para el usuario solo requiere introducir su usuario y contraseña pero a nivel de servidor se establecen más variables como el lenguaje deseado o la tienda a la que corresponde. Cada página tiene en

su cabecera un control de sesión que te redirecciona a la página inicial si estas variables no están presentes.

A pesar de que todas las acciones con la base de datos se realizan desde los ficheros PHP este no es el encargado de mantener y gestionar la base de datos, simplemente es la puerta de acceso a ella. Como sistema gestor de base de datos se utiliza MySQL y aunque no sea usado de un modo directo cada vez que se realiza una llamada es este sistema quien la procesa. Además para la creación de las tablas y sus atributos sí ha sido este sistema el usado.

Uno de los detalles más importantes del proyecto es que los dos módulos (la tienda pública y la aplicación de gestión) comparten la misma base de datos y atacan sobre las mismas tablas. Esto se refiere a que los artículos mostrados en la tienda y los listados en la parte de gestión son los mismos, solo que, dependiendo de su valor activo, se muestran o no en la tienda. Se ha seguido esta estructura con el fin de evitar duplicaciones o tener guardados dobles, además de que garantiza que los datos vistos desde un módulo se correspondan fielmente con el otro.

El sistema en su conjunto está formado por más de 60 tablas, debido a que la aplicación maneja mucha información persistente. Sin embargo, muchas de ellas siguen una estructura muy parecida, ya que como se lleva diciendo a lo largo del texto, hay diversas secciones que aunque no están directamente relacionadas sus datos se organizan de una manera parecida, es el caso de todos los listados. Con los listados nos referimos, por ejemplo, a los pedidos que le aparecen en el perfil a un usuario de la tienda o los albaranes y facturas que aparecen en la parte de gestión. La estructura de todas estas secciones se basa en una tabla principal y otra vinculada con esta que almacena la información de cada línea. Para ejemplificar esto se muestra la estructura que sigue la sección de facturas a cliente. Su tabla principal es *FACTURAS\_CLI* y como campos relevantes tiene:

·*IdFactura*: campo autonumérico, es la clave primaria de esta tabla

·*NumFactura*: es el número de la factura, formado por el identificador de la tienda dado en la parte de gestión concatenado con el número de factura que le corresponde.

·*IdCliente*: clave primaria del cliente que ha realizado la compra.

·*IdAlbaran*: clave primaria del albarán al que se le asocia la factura.

·*Base, IVA*: campos referentes a los costos de la factura.

·*Pagado*: estado de la factura, pagada o pendiente.

·*Rectificada*: la factura ha sido devuelta o no.

Estos datos almacenan la información básica de la factura, sin embargo no es posible saber qué productos han sido vendidos en esta factura. Es por eso que existe la tabla vinculada *FACTURAS\_CLI\_LIN* que hace referencia a cada línea de la factura y que contiene entre otros campos los siguientes:

- Id*: campo autonumérico, es la clave primaria de esta tabla.
- IdFactura*: clave primaria de la factura a la que pertenece.
- IdModelo*: clave primaria del artículo que se ha vendido.
- Cantidad, Precio, CodIVA*: datos referentes al importe de la línea

Esta estructuración no solamente es aplicable en los listados, si no que la información de los usuarios también es dividida en varias tablas. Se cuenta por ejemplo con una tabla de usuarios en la que se almacena la información más relevante, pero sus direcciones se guardan en otra tabla relacionada con esta para dar la posibilidad de almacenar varias direcciones por usuarios y además simplificar la tabla principal.

#### **4.2.2 mPDF, PHPSpreadsheet**

Para mostrar los informes pedidos se hace uso de dos librerías que atacan directamente a la base de datos y crean ficheros en el formato deseado para poder descargarlos.

La primera de ellas es mPDF, que permite crear ficheros PDFs. Para crearlos se vale de la programación orientada a objetos. El primer paso es llamar al constructor y hacer uso de los métodos de clase para establecer ciertos atributos como márgenes, formato, cabecera... Posteriormente su contenido se agrega mediante HTML. La forma más cómoda es con la creación de una tabla e ir insertando elementos en ella.

PHPSpreadsheet funciona de la misma manera. Es una librería que aporta sus propios objetos con sus métodos. En este caso no se hace uso del HTML, si no que se especifica el contenido por columnas, filas o celdas mediante el método específico. Permitiendo hacer las mismas modificaciones (colores, tipos de borde...) que permitiría una aplicación nativa.

#### **4.2.3 Swiftmailer**

En el módulo de la tienda pública cada vez que se da de alta un usuario nuevo se requiere un correo electrónico. Para confirmar la validez de ese correo y para tener la posibilidad de contactar con ese usuario en un futuro, necesitamos de una herramienta que nos permita realizar envíos automáticos de emails. La librería swiftmailer cumple esta función y su funcionamiento es muy parecido a las dos librerías explicadas anteriormente. A través de objetos PHP propios de esta biblioteca, se crean y envían correos. La única diferencia es que en este caso hace falta disponer de un servidor de correo en donde está alojada la aplicación. Sin embargo, esto no ocasiona un gran inconveniente puesto que actualmente la mayoría, si no todos los servidores, te permiten instalar un servidor de correo de manera fácil. Al igual que mPDF es posible dar estilos con etiquetas HTML a los correos enviados.





# 5. Resultados

## 5.1 Tienda pública

### 5.1.1 Proceso de compra

Uno de los objetivos marcados para el módulo de la tienda es conseguir que un potencial cliente realice una compra en los mínimos clicks posibles. El inconveniente de este requisito es que también es deseable que solo usuarios registrados puedan realizar compras.

Para que un usuario anónimo conserve los productos seleccionados se le crea una sesión de manera transparente, así cuando completa el registro, los productos del carro se le guardan también.

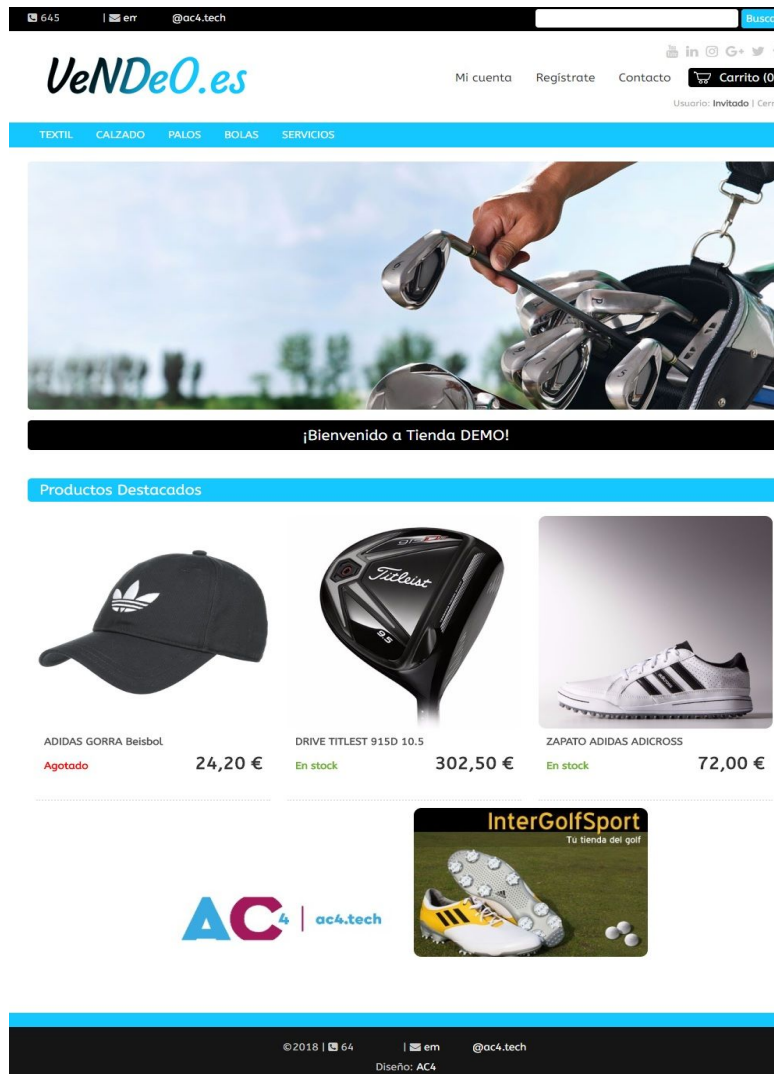


Imagen 8. Página principal de la tienda

A continuación se muestra una serie de imágenes con el proceso de compra para un usuario anónimo. Se ha conseguido que en el peor de los escenarios el proceso de compra se realice en solo siete transiciones. La primera pantalla con la que se encuentra el cliente es la pantalla inicial de la tienda, imagen 8. En ella se aprecia el buscador, los enlaces a las redes sociales de la tienda, la sesión de usuario, los artículos destacados y la publicidad. Todo ello personalizable desde el panel de gestión. La imagen grande que se ve es el slider, un complemento que va mostrando diferentes imágenes a lo largo del tiempo, también personalizable desde el panel de gestión. También se aprecia si hay stock disponible de los productos e incluso el mensaje personalizado de la tienda.

Una vez seleccionado el producto deseado, se abre su ficha. En ella aparecen los detalles guardados en la herramienta de gestión. Para los atributos definidos se muestra un selector que te permite elegir el valor deseado de ese atributo. Al cambiar el valor se actualiza el precio y el stock si fuera necesario. También desde la ficha se puede acceder al árbol de las familias y subfamilias. Cuando se pulsa sobre incluir al carrito se actualiza la vista de la página sin que esta sea refrescada, gracias a las llamadas AJAX.

The screenshot shows the product page for a Titleist 915D 10.5 driver on the VeNDeO.es website. The page layout includes a top navigation bar with contact information (645, en, @ac4.tech) and a search bar. The main header features the VeNDeO.es logo, social media icons, and links for 'Mi cuenta', 'Regístrate', 'Contacto', and 'Carrito (0)'. A secondary navigation bar lists categories: TEXTIL, CALZADO, PALOS, BOLAS, and SERVICIOS. The product title 'DRIVE TITLEST 915D 10.5' is displayed in a dark bar. Below the title, a breadcrumb trail reads 'INICIO | DRIVER | TITLEIST | DRIVE TITLEST 915D 10.5'. The product image is a large, detailed view of the driver head, with a smaller thumbnail below it. To the right of the image, the 'FICHA TÉCNICA' section shows a dropdown menu for 'PALO DIESTRO/ZURDO' set to 'ZURDO'. Below this, sections for 'DESCRIPCIÓN' and 'PRECIO' are visible. The price is listed as 250,00€, and the quantity is set to 1. A blue button labeled 'Incluir al Carrito' is positioned below the price and quantity. The footer contains copyright information (© 2018 | 645), contact details (en, @ac4.tech), and the designer's name (Diseño: AC4).

Imagen 9. Ficha del artículo

Una vez el cliente tiene todos los productos deseados, solo ha de pulsar sobre el botón de carrito para acceder a un resumen de su compra. Esto se puede ver en la imagen 10. Desde aquí se pueden eliminar artículos o modificar el número de ellos deseados. También se muestra el importe total. Para proceder al pago se pulsa sobre “Tramitar pedido”, en este caso como es una sesión anónima, el sistema lo detecta y mueve al cliente a la página de registro, sin que este pierda los artículos guardados en el carro. Si el cliente ya se encuentra registrado, este paso se omite y se le desplaza directamente al último paso, imagen 13, para finalizar su compra y proceder al pago.

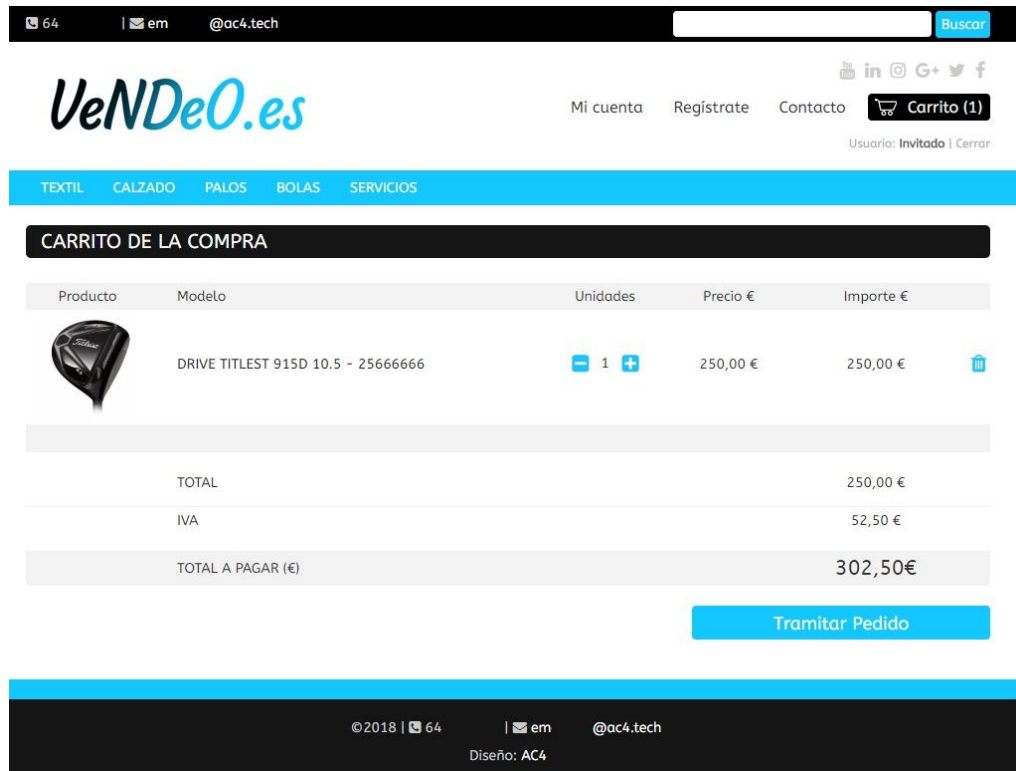


Imagen 10. Carrito de la sesión del invitado

Siguiendo con la traza de una sesión anónima la siguiente transición corresponde al registro del usuario.

Para finalizar con los datos del cliente se le pide que agregue una dirección. En esta transición se puede ver como el nombre de la sesión ya se ha cambiado. Una vez guardada la nueva dirección, se muestra la pantalla final.

The image shows the registration form on the VeNDeO.es website. The header includes the phone number 64, email em, and social media handle @ac4.tech. The logo VeNDeO.es is prominently displayed. Navigation links for 'Mi cuenta', 'Regístrate', and 'Contacto' are present, along with a shopping cart icon showing 'Carrito (1)'. The user is identified as 'Invitado'. A blue navigation bar contains categories: TEXTIL, CALZADO, PALOS, BOLAS, and SERVICIOS. The form title is 'FORMULARIO DE REGISTRO'. It contains the following fields: C.I.F. / N.I.F., Nombre, Teléfono, Contraseña, Tipo Cliente (dropdown menu with 'Particular' selected), Apellidos, Email, and Repita contraseña. A blue 'Registrarse' button is at the bottom. The footer shows copyright ©2018, phone 64, email em, social media @ac4.tech, and 'Diseño: AC4'.

Imagen 11. Formulario de registro

The image shows the address form on the VeNDeO.es website. The header is identical to the registration page. The user is now identified as 'vendeo@email.com'. The form title is 'Nueva Dirección'. It contains the following fields: Dirección (text input with 'Carrer del Molino, 2'), País (dropdown menu with 'ESPAÑA' selected), Población (dropdown menu with 'ABENGIBRE' selected), Teléfono (text input with '666666666'), Provincia (dropdown menu with 'ALBACETE' selected), Código Postal (text input with '55555'), and Email (text input with 'vendeo@email.com'). Two blue buttons, 'Volver' and 'Guardar', are at the bottom. The footer is identical to the registration page.

Imagen 12. Formulario de direcciones

El proceso de compra finaliza con la imagen 13. Esta pantalla muestra un resumen con los datos personales, la dirección y los artículos a adquirir. Pulsar sobre el botón comprar nos redirige a la pasarela de pago del banco, elimina los artículos del carrito y regresa al área personal del cliente, donde se ven los pedidos realizados.

The screenshot shows the checkout page of VeNDeO.es. At the top, there is a navigation bar with the logo, social media icons, and a shopping cart icon labeled 'Carrito (1)'. Below this is a blue navigation bar with categories: TEXTIL, CALZADO, PALOS, BOLAS, and SERVICIOS. The main content is divided into three sections: 'Datos Personales', 'Dirección', and 'Cesta'.


**Datos Personales**

|                 |  |        |   |           |                                      |
|-----------------|--|--------|---|-----------|--------------------------------------|
| C.I.F. / N.I.F. | <input type="text" value="1234567-x"/> | Nombre | <input type="text" value="Hugo"/>             | Apellidos | <input type="text" value="Montoto"/> |
| Teléfono        | <input type="text" value="666666666"/> | Email  | <input type="text" value="vendeo@email.com"/> |           |                                      |

**Dirección**

|           |   |               |                                    |
|-----------|---|---------------|------------------------------------|
| Dirección | <input type="text" value="Carrer del Molino, 2"/> |               |                                    |
| Población | <input type="text" value="ALBACETE"/>             | Código Postal | <input type="text" value="55555"/> |

**Cesta**

| Producto   | Modelo                             | Unidades | Precio € | Importe € |
|--|------------------------------------|----------|----------|-----------|
|  | DRIVE TITLEST 915D 10.5 - 25666666 | 1        | 250,00 € | 250,00 €  |
| TOTAL  |                                    |          |          | 250,00 €  |
| IVA  |                                    |          |          | 52,50 €   |
| TOTAL A PAGAR (€)  |                                    |          |          | 302,50€   |

[Comprar ahora](#)

Imagen 13. Finalización de compra

### 5.1.2 Exploración de artículos

Un aspecto clave en cualquier tienda es la búsqueda de archivos. En este proyecto se especificó que esta búsqueda se pudiera hacer de forma directa o a través del árbol de familias. Ambos métodos están disponibles desde cualquier página de la tienda, ya que están incluidas en la cabecera.

En la imagen 14 se puede ver el campo de búsqueda en la parte superior y también el árbol desplegado de familias y subfamilias correspondientes a a la sección de textil.

Junto a estos dos métodos de exploración se ha añadido un sistema de filtrado, que permite especificar más la búsqueda. Este sistema de filtros se basa en los atributos asociados a los artículos, y se puede ver en la imagen 15.

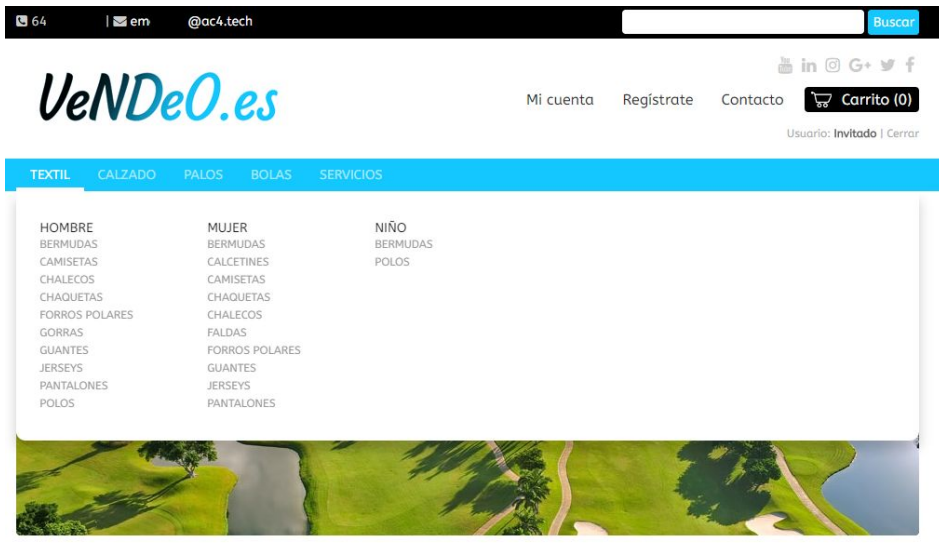


Imagen 14. Árbol de la sección textil

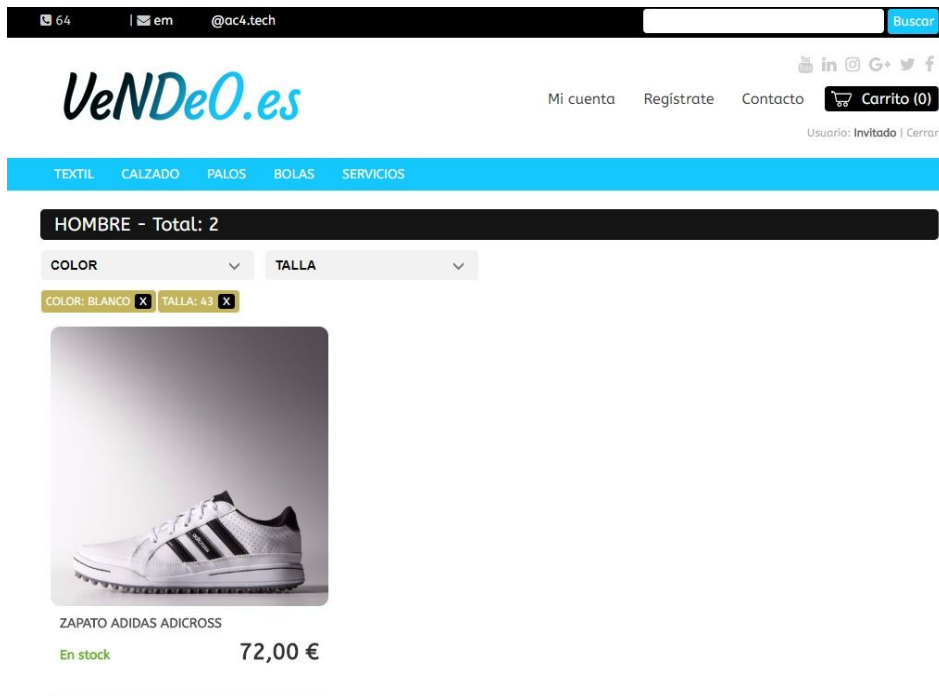


Imagen 15. Resultado de la aplicación de los filtros.



## 5.2 Herramienta de gestión

### 5.2.1 Interfaz general y listados

En esta sección se mostrará brevemente los resultados obtenidos en el módulo de gestión. Se puede obtener una visión más detallada de la interfaz de usuario en el anexo 1 que corresponde con el manual de la aplicación. Se ha optado por una interfaz sencilla y fácilmente modulable para permitir distintas vistas dependiendo del rol del usuario. En la imagen 16 se muestra la pantalla inicial, desde un punto de vista del administrador, ya que si fuera un usuario sin privilegios, algunas secciones como usuarios o tienda web no estarían visibles.

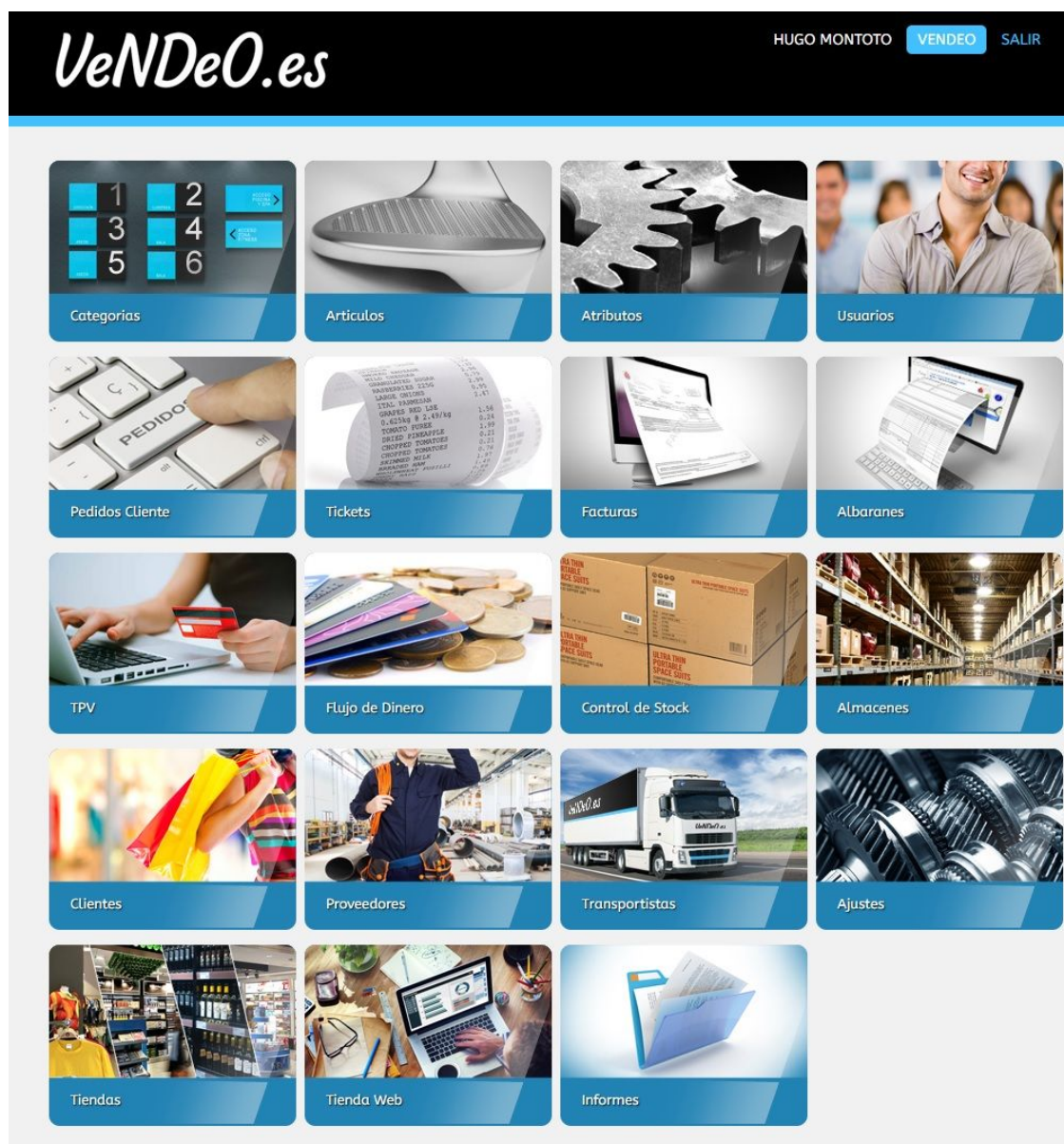


Imagen 16. Menú principal de la herramienta de gestión

Como se ha mencionado a lo largo del proyecto, los listados son una parte muy importante de la aplicación. Concentrar mucha información en una sola pantalla es complicado, sin embargo, el uso de los grids ha facilitado este objetivo. En la imagen 17 se puede apreciar un resumen de los tickets expedidos a los clientes. Además del filtrado por columnas, se ofrece al usuario poder mostrar solo los tickets expedidos en un rango de fechas con el uso del filtro que aparece en la parte superior.

| TICKETS A CLIENTES |            |       |             |           |            |             |                                     |              |        |
|--------------------|------------|-------|-------------|-----------|------------|-------------|-------------------------------------|--------------|--------|
|                    |            | Desde | 01-01-2018  | hasta     | 03-07-2018 | Tienda DEMO |                                     |              |        |
| Nº Ticket          | Fecha      | Año   | Tienda      | Base      | IVA        | Total       | Pagado                              | Forma Pago   | Nombre |
| x                  | x          | x     | x           | x         | x          | x           | x                                   | x            | x      |
| CEN-FS201800053    | 28-03-2018 | 2018  | Tienda DEMO | 6,00€     | 1,26€      | 7,26€       | <input checked="" type="checkbox"/> | VARIAS       | Caja   |
| CEN-FS201800052    | 28-03-2018 | 2018  | Tienda DEMO | 250,00€   | 52,50€     | 302,50€     | <input checked="" type="checkbox"/> | TARJETA VISA | Caja   |
| CEN-FS201800051    | 28-03-2018 | 2018  | Tienda DEMO | 225,00€   | 47,25€     | 272,25€     | <input checked="" type="checkbox"/> | CONTADO      | Caja   |
| CEN-FS201800050    | 26-03-2018 | 2018  | Tienda DEMO | 1.400,00€ | 0,00€      | 1.400,00€   | <input checked="" type="checkbox"/> | CONTADO      | Caja   |
| CEN-FS201800049    | 26-03-2018 | 2018  | Tienda DEMO | 1.000,00€ | 210,00€    | 1.210,00€   | <input checked="" type="checkbox"/> | CONTADO      | Caja   |
| CEN-FS201800048    | 23-03-2018 | 2018  | Tienda DEMO | 250,00€   | 52,50€     | 302,50€     | <input checked="" type="checkbox"/> | TARJETA VISA | Caja   |
| CEN-FS201800047    | 23-03-2018 | 2018  | Tienda DEMO | 250,00€   | 52,50€     | 302,50€     | <input checked="" type="checkbox"/> | CONTADO      | Caja   |
| CEN-FS201800046    | 23-03-2018 | 2018  | Tienda DEMO | 250,00€   | 52,50€     | 302,50€     | <input checked="" type="checkbox"/> | TARJETA VISA | Caja   |
| CEN-FS201800045    | 23-03-2018 | 2018  | Tienda DEMO | 250,00€   | 52,50€     | 302,50€     | <input checked="" type="checkbox"/> | CONTADO      | Caja   |
| CEN-FS201800044    | 23-03-2018 | 2018  | Tienda DEMO | 250,00€   | 52,50€     | 302,50€     | <input checked="" type="checkbox"/> | CONTADO      | Caja   |
| CEN-FS201800043    | 23-03-2018 | 2018  | Tienda DEMO | 250,00€   | 52,50€     | 302,50€     | <input checked="" type="checkbox"/> | CONTADO      | Caja   |
| CEN-FS201800042    | 22-03-2018 | 2018  | Tienda DEMO | 6,00€     | 1,26€      | 7,26€       | <input checked="" type="checkbox"/> | CONTADO      | Caja   |
| CEN-FS201800041    | 20-03-2018 | 2018  | Tienda DEMO | 250,00€   | 52,50€     | 302,50€     | <input checked="" type="checkbox"/> | CONTADO      | Caja   |
| CEN-FS201800040    | 20-03-2018 | 2018  | Tienda DEMO | 250,00€   | 52,50€     | 302,50€     | <input checked="" type="checkbox"/> | CONTADO      | Caja   |

Imagen 17. Listado de tickets a cliente

### 5.1.2 Informes

Uno de los objetivos de la aplicación es la generación de informes. A partir del apartado informes que aparece en la imagen 16 se puede acceder a un menú en el que elegir qué informe se quiere. Algunos de estos vuelven a tener otro submenú para elegir una serie de filtros como el periodo del que se quiere obtener información o al cliente del que queremos obtener los datos. En la imagen 18 se ve un ejemplo de informe general.

### 5.1.2 Terminal Punto de Venta

Otro de los puntos esenciales en la aplicación era la inclusión de una TPV. Finalmente se ha conseguido integrar una TPV muy completa. Dispone de acceso a todos los artículos, realiza descuentos en función del cliente, funciona con lectores de códigos de barras y genera tickets de forma automática entre muchas de sus funciones. En la imagen 19 se aprecia su estado visual y también se pueden ver algunas de sus características.



| Rango de fechas:                  |                                   | Histórico                    |                       |                               |
|-----------------------------------|-----------------------------------|------------------------------|-----------------------|-------------------------------|
| <b>VENDEDOR</b>                   |                                   | TODOS                        |                       |                               |
| <b>SECCIÓN</b>                    |                                   | TODAS                        |                       |                               |
| <b>FAMILIA</b>                    |                                   | TODAS                        |                       |                               |
| <b>SUBFAMILIAS</b>                |                                   | TODAS                        |                       |                               |
| <b>MODELOS</b>                    |                                   | TODOS                        |                       |                               |
| <b>Nº de ventas:</b>              |                                   | 98                           |                       |                               |
| <b>Nº factura:</b> CEN-N201800012 | <b>Fecha:</b> 2018-07-02 00:00:00 | <b>Cliente:</b> Hugo Montoto | <b>Vendedor:</b> WEB  |                               |
| #ID                               | Nombre                            | Cantidad                     | Descuento             | Precio Venta                  |
| 25666666                          | DRIVE TITLEST 915D 10.5           | 1,00                         | 0,00%                 | 302,50€                       |
| <b>Factura Base</b>               |                                   | <b>250,00 €</b>              | <b>Factura IVA</b>    | <b>52,50 € Total 302,50 €</b> |
| <b>Nº factura:</b> CEN-N201800011 | <b>Fecha:</b> 2018-03-08 00:00:00 | <b>Cliente:</b>              | <b>Vendedor:</b> WEB  |                               |
| #ID                               | Nombre                            | Cantidad                     | Descuento             | Precio Venta                  |
| 35655R                            | ADIDAS GORRA Beisbol              | 1,00                         | 0,00%                 | 24,20€                        |
| <b>Factura Base</b>               |                                   | <b>20,00 €</b>               | <b>Factura IVA</b>    | <b>4,20 € Total 24,20 €</b>   |
| <b>Nº factura:</b> CEN-N201800008 | <b>Fecha:</b> 2018-01-16 11:22:28 | <b>Cliente:</b> Caja CENTRAL | <b>Vendedor:</b> hugo |                               |
| #ID                               | Nombre                            | Cantidad                     | Descuento             | Precio Venta                  |
| 35655A                            | ADIDAS GORRA Beisbol              | 1,00                         | 0,00%                 | 24,20€                        |
| <b>Factura Base</b>               |                                   | <b>20,00 €</b>               | <b>Factura IVA</b>    | <b>4,20 € Total 24,20 €</b>   |
| <b>Nº factura:</b> CEN-N201800007 | <b>Fecha:</b> 2018-01-16 09:57:04 | <b>Cliente:</b> Caja CENTRAL | <b>Vendedor:</b> hugo |                               |
| #ID                               | Nombre                            | Cantidad                     | Descuento             | Precio Venta                  |
| 35655A                            | ADIDAS GORRA Beisbol              | 1,00                         | 0,00%                 | 24,20€                        |
| <b>Factura Base</b>               |                                   | <b>20,00 €</b>               | <b>Factura IVA</b>    | <b>4,20 € Total 24,20 €</b>   |
| <b>Nº factura:</b> CEN-N201800006 | <b>Fecha:</b> 2018-01-15 12:59:01 | <b>Cliente:</b> Caja CENTRAL | <b>Vendedor:</b> hugo |                               |

Imagen 18. Informe general de ventas

**TPV**
HUGO MONTOTO | VENDEO

CLIENTE 
TIENDA

LECTURA DEL CÓDIGO DE BARRAS

BOLAS CALLAWAY HEX  
6 €/U. IVA N.I.  
NÚMERO DE SERIE  
123456

- 1 +

7.26

Dto. 0 %

---

DRIVE TITLEST 915D 10.5  
250 €/U. IVA N.I.  
NÚMERO DE SERIE  
25666666

- 1 +

302.5

Dto. 0 %

309.76 €

789

456

123

0,

TOTAL ARTICULOS : 2

Imagen 19. Interfaz TPV



# 6. Conclusiones

---

Este proyecto ha sido el primer proyecto en desarrollo web que he realizado. Nunca antes había utilizado la mayoría de los lenguajes usados, sin embargo, disponer de las bases de programación de otros lenguajes me ha ayudado mucho a reducir la curva de aprendizaje.

No obstante durante la realización del proyecto ha habido ocasiones en las que avanzar era muy complicado. Al no tener un conocimiento amplio del lenguaje la realización de algunas funcionalidades se hacían mediante aproximaciones que complicaba mucho el código cuando, en realidad, el propio lenguaje disponía de funciones propias para realizar esta funcionalidad.

Creo que la aplicación ha alcanzado todos los objetivos de los que se partía, salvo quizás mantener la simplicidad. Durante la realización del proyecto han ido apareciendo problemas que complicaron la programación y sobre todo el diseño. Es esto lo que realmente me ha servido para ver que una fase de análisis ha de ser mucho más exhaustiva de lo que normalmente son. Dedicar más tiempo en esta fase te puede ayudar a paliar futuros inconvenientes y evitar pérdidas de tiempo asociadas a estos.

En definitiva, este proyecto me ha ayudado mucho a comprender cómo funciona el desarrollo web y además me ha aportado conocimientos fuera del ámbito de la programación, dándome experiencia en el mundo empresarial y en todas las fases por las que pasa un proyecto como la captación de clientes, búsqueda de financiación o exposición del producto.



# 7. Bibliografía y documentación consultada

---

## 7.1 Referencias

---

[1] Alexandre Fonseca. (2014). Fundamentos del E-Commerce: Tu guía de comercio electrónico y negocios online. España: Bubok.

[2] Becker, A. (2008). Electronic Commerce: Concepts, Methodologies, Tools, and Applications (4 Volumes) (pp. 1-2522). Hershey, PA: IGI Global.

[3] W3Techs - World Wide Web Technology Surveys. Usage of content management systems for websites. [https://w3techs.com/technologies/overview/content\\_management/all](https://w3techs.com/technologies/overview/content_management/all)

[4] Internet Retailer Online Marketplaces Database. (2017) .  
<https://www.digitalcommerce360.com/2018/01/11/infographic-largest-online-marketplaces-world/>

[5] Cuponation. (2017). Ranking e-commerce 2017  
<https://www.cuponation.es/estudios/ranking-ecommerce>

[6] Greg Bensinger. (27 Junio , 2012). Competing With Amazon on Amazon. The Wall Street Journal.

[7] Shelley Powers (2016) Learning Node, 2nd Edition. Sebastopol, CA: O'REILLY

[8] W3Techs - World Wide Web Technology Surveys. Usage statistics and market share of jQuery for websites. <https://w3techs.com/technologies/details/js-jquery/all/all>

[9] Ionic Usage Statistics <https://trends.builtwith.com/framework/Ionic>

## 7.2 Software empleado

---

·Editor de texto Sublimetext. <https://www.sublimetext.com/>

- Base de datos MySQL. <https://www.mysql.com/>
- Servidor Apache. <https://www.apachefriends.org/es/index.html>
- Servidor FTP FileZilla. <https://filezilla-project.org/>

## 7.3 Documentación

---

Cody Lindley. (2009) jQuery Cookbook. Sebastopol, CA: O'REILLY

Robin Nixon. (2014) Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML. Sebastopol, CA: O'REILLY

<http://php.net>

<https://www.w3schools.com>

<https://desarrolloweb.com/php/>



ANEXO A

## Manual de usuario

---



# Vendeo: Manual de usuario

---



# CONTENIDOS

---

## CONTENIDOS

### ARTÍCULOS

DAR DE ALTA UN ARTÍCULO

### ALBARANES

DAR DE ALTA UN ALBARÁN

FACTURAR ALBARÁN

### TPV

USO DE LA TPV

FORMAS DE PAGO EN LA TPV

### CONTROL DE STOCK

### CLIENTES

DAR DE ALTA UN CLIENTE

### TIENDAS

PASOS PARA DAR DE ALTA UNA TIENDA NUEVA

CONFIGURAR NUEVA TIENDA

CONFIGURAR TIENDA WEB

# ARTÍCULOS

## DAR DE ALTA UN ARTÍCULO

Antes de dar de alta un artículo es necesario comprender que significa esto en Vendeo.

Un artículo es un item que posee diversos atributos como un color o talla. Un ejemplo de artículo sería: ZAPATO ADIDAS ADICROSS. De los artículos obtenemos los modelos. Un modelo es un artículo con sus atributos definidos. Un ejemplo de modelo sería: ZAPATOS ADIDAS ADICROSS{ TALLA:40, COLOR:ROJO}.

Para dar de alta un artículo:

MENÚ PRINCIPAL->ARTÍCULOS->Nuevo Artículo (icono + en esquina superior derecha)

| Id    | Sección | Familia  | SubFamilia | Nombre               | Stock | Activo                              |
|-------|---------|----------|------------|----------------------|-------|-------------------------------------|
| 35036 | TEXTIL  | GORRAS   | UNISEX     | ADIDAS GORRA Beisbol | 1     | <input checked="" type="checkbox"/> |
| 35365 | TEXTIL  | BERMUDAS | CABALLERO  | ADIPURE Bermuda      | 0     | <input checked="" type="checkbox"/> |
| 35366 | TEXTIL  | BERMUDAS | CABALLERO  | ADIPURE Bermuda      | 0     | <input checked="" type="checkbox"/> |

- 1) Seleccionamos la familia a la que pertenece y aceptamos.
- 2) Buscamos el artículo en el grid

| Id    | Sección | Familia | SubFamilia | Nombre                    | Stock | Activo                              |
|-------|---------|---------|------------|---------------------------|-------|-------------------------------------|
| 35203 | CALZADO | ADIDAS  | CABALLERO  | ZAPATO ADIDAS ADICROSS    | 38    | <input checked="" type="checkbox"/> |
| 35205 | CALZADO | ADIDAS  | CABALLERO  | ZAPATO ADIDAS ADICROSS IV | 0     | <input checked="" type="checkbox"/> |
| 35207 | CALZADO | ADIDAS  | CABALLERO  | ZAPATO ADIDAS ADICROSS V  | 0     | <input checked="" type="checkbox"/> |

- 3) Editamos haciendo doble click.

La ficha del artículo cuenta con varias pestañas.

- **General:** información general del artículo.

**ZAPATO ADIDAS ADICROSS**

General Fotos Ficheros Reglas de Precios Stock Pedidos a Proveedores Clientes Atributos Modelos

Nombre: ZAPATO ADIDAS ADICROSS

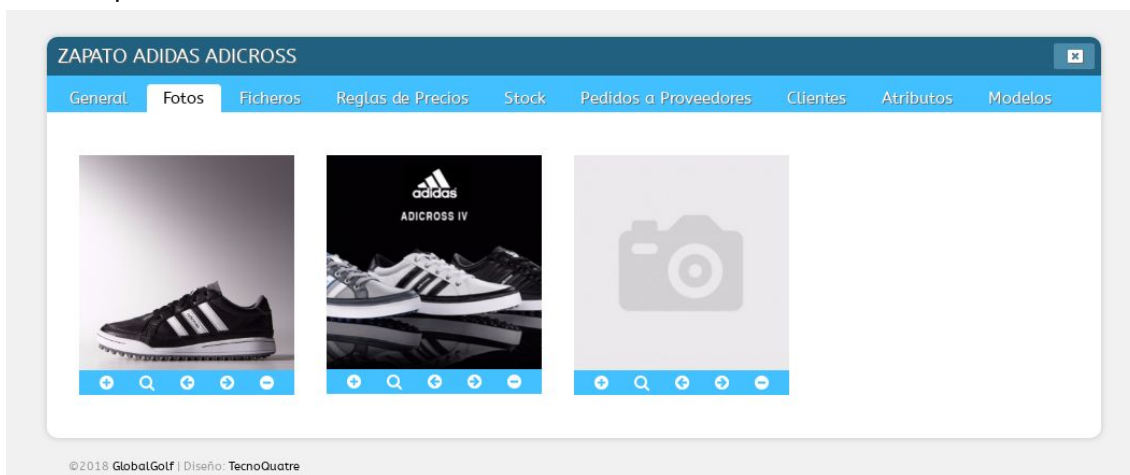
Nombre EN: Nombre en inglés

Sección: CALZADO Familia: ADIDAS SubFamilia: CABALLERO

Sección EN: SHOES Familia EN: SubFamilia EN:

Proveedor: Seleccione proveedor Activo: SI Destacado: Si

- **Fotos:** permite subir, eliminar, reordenar fotos.



- **Ficheros:** permite subir, eliminar ficheros, como pudieran ser libros de instrucciones o folletos del artículo.

**ZAPATO ADIDAS ADICROSS**

General Fotos Ficheros Reglas de Precios Stock Pedidos a Proveedores Clientes Atributos Modelos

| Nombre   | Ficheros             | Tipo            | Orden | Fecha               | Visible                             |
|----------|----------------------|-----------------|-------|---------------------|-------------------------------------|
| ADICROSS | 171213_131446_22.pdf | application/pdf | 1     | 13-12-2017 13:14:46 | <input checked="" type="checkbox"/> |

- **Reglas de precios:** permite crear reglas de precios para clientes o grupos de clientes específicos. Las reglas de precio se aplican a un modelo concreto o a todos

los modelos de un mismo artículo en función de nuestras necesidades.

| Beneficiario | Descuento | Modelo   |
|--------------|-----------|----------|
| Provatinas   | 10.00%    | TODOS    |
| R Mora       | 50.50%    | TODOS    |
| SOCIOS GOLD  | 3.00%     | TODOS    |
| Hugo Montoto | 3.00%     | 36121-43 |

· **Stock:** muestra el stock disponible de todos los modelos en todos los almacenes. También se da la posibilidad de crear/eliminar stock.

| Cod. Barras | Cantidad | Almacén          | Tienda   | Referencia |
|-------------|----------|------------------|----------|------------|
| 36121-44    | 10       | ALMACÉN LOCAL    | CENTRAL  |            |
| 36121-43    | 3        | ALMACÉN VALENCIA | VALENCIA | 36121-43   |

· **Pedidos a proveedor:** muestra información de los pedidos a proveedor hechos que contienen algún modelo de este artículo.

| Nº Pedido       | Estado      | Cod. Barras | Cantidad | Almacén       | Fecha      | Usuario |
|-----------------|-------------|-------------|----------|---------------|------------|---------|
| CEN-PP201800001 | EN CREACION | 36121-43    | 1        | ALMACÉN LOCAL | 2018-01-04 | hugo    |
| CEN-PP201700003 | RECIBIDO    | 36121-43    | 5        | ALMACÉN LOCAL | 2017-12-19 | hugo    |

· **Cientes:** muestra información de los clientes que han comprado este artículo.

| NombreCom    | CIF       | Tipo           | Nº de Documento | Cantidad | Pedo (€) | Fecha      |
|--------------|-----------|----------------|-----------------|----------|----------|------------|
| R Mora       | 14794652X | Factura Normal | CEN-N201700019  | 1,00     | 35,34€   | 22-03-2018 |
| Provatinas   | 19836888W | Factura Normal | CEN-N201700017  | 1,00     | 64,25€   | 22-05-2017 |
| R Mora       | 14794652X | Factura Normal | CEN-N201700009  | 1,00     | 72,00€   | 20-12-2016 |
| Hugo Montoto | 71641537  | Factura Normal | TQ-N201700002   | 1,00     | 72,00€   | 19-12-2016 |
| Provatinas   | 19836888W | Factura Normal | TQ-N201700001   | 1,00     | 64,80€   | 19-12-2016 |
| CajaTienda   | CAJA      | Ticket         | CEN-FS201700005 | 1,00     | 72,00€   | 19-03-2021 |
| Hugo Montoto | 71641537  | Ticket         | CEN-FS201700002 | 1,00     | 72,00€   | 19-12-2016 |

· **Atributos:** permite agregar/eliminar atributos al artículo. Para asignar un atributo a un artículo ese atributo tiene que estar definido. Los atributos se definen en MENÚ PRINCIPAL -> ATRIBUTOS.

| Atributo | Descripción | Tipo   | Valores  | Mínimo | Máximo |
|----------|-------------|--------|--|--------|--------|
| TALLA    | TALLA       | NUMERO |  | 16     | 50     |
| COLOR    | COLOR       | LISTA  | ROJO;NARANJA;AMARILLO;VERDE;CIAN;AZUL;VIOLETA;BLANCO |        | 0      |

· **Modelos:** permite dar de alta a nuevos modelos, indicando el precio de venta y dando valores a los atributos.

| Id   | Cod. Barras | Precio Venta | Precio Coste | Atributo 1 | Valor | Atributo 2 | Valor | Atributo 3 | Valor |
|------|-------------|--------------|--------------|------------|-------|------------|-------|------------|-------|
| 1245 | 36121-43    | 59,50€       | 10,00€       | TALLA      | 43    |            |       |            |       |
| 1247 | 36121-44    | 59,00€       | 10,00€       | TALLA      | 44    |            |       |            |       |

## ALBARANES

### DAR DE ALTA UN ALBARÁN

Los albaranes en Vendeo funcionan como un justificante de entrega de artículos. Esto quiere decir que se ve influenciado por el stock disponible. No se puede crear un albarán si no existe stock. Por otra parte, los albaranes no proporcionan ingresos. Para que el pago del albarán se tramite hay que facturarlos.

Para **dar de alta** un albarán:

MENÚ PRINCIPAL->ALBARANES->Nuevo Albarán (icono + en esquina superior derecha).

Después de dar de alta el albarán podemos añadir artículos desde la pestaña Artículos.

Añadir Artículo
✕

|                 |                      |            |               |
|-----------------|----------------------|------------|---------------|
| Cod. Barras     | 35655R               |            |               |
| Nombre          | ADIDAS GORRA Beisbol | Referencia | 35655R        |
| Precio Unidad   | 20,00 €              | Descuento  | 0 %           |
| Precio Rebajado | 20,00 €              | IVA        | Normal (21) ▾ |
| Número de serie |                      | Cantidad   | 1             |
| Notas           |                      |            |               |

Guardar
Cancelar

*El descuento sólo se aplica al artículo de este albarán*

## FACTURAR ALBARÁN

Desde la pestaña General del albarán pulsar sobre FACTURAR. Facturar el albarán deshabilitará las modificaciones y creará una entrada en la sección Flujo de Dinero.

*No se puede facturar un albarán sin artículos*

*Un albarán solo puede generar una factura*

Tras pulsar el botón se nos muestra una ventana que nos muestra las opciones de pago. Este diálogo nos permite abonar la factura en un solo pago o dividirla en varios. Para realizar un pago único simplemente pulsamos sobre la modalidad de pago deseada. Si por el contrario se quiere dividir la factura pulsamos sobre Dividir. Introducimos la cantidad y hacemos click en la modalidad de pago deseada. Repetimos hasta que las fracciones sumen el importe total.

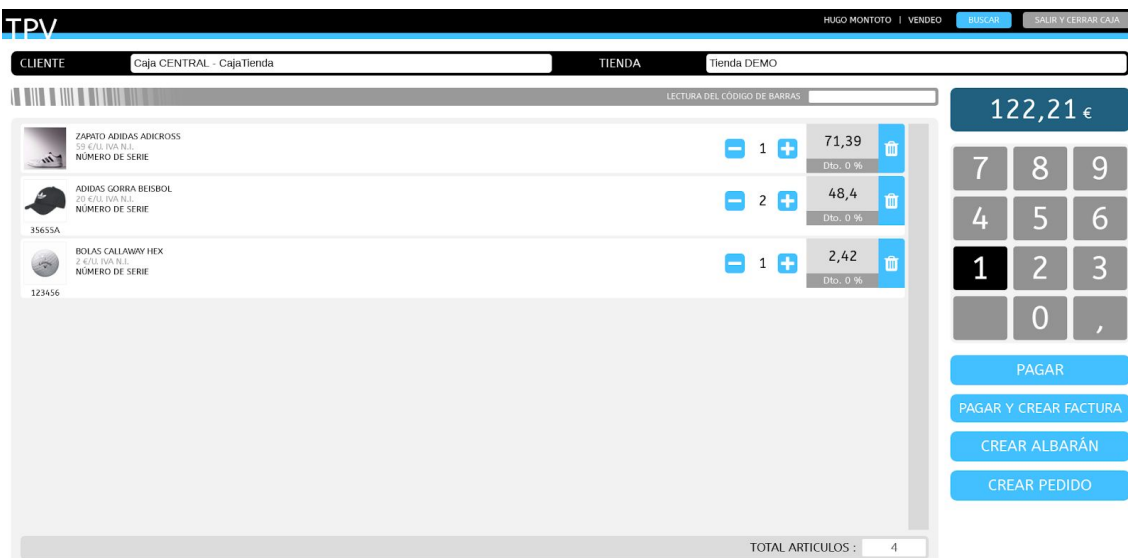


## TPV

### USO DE LA TPV

Para comenzar con la TPV: MENÚ PRINCIPAL -> TPV -> Iniciar sesión

Aparecerá una pantalla similar a la siguiente:



Se observan tres secciones, en orden desde la parte superior:

- Cabecera:** Muestra el usuario que está usando la TPV y la tienda a la que pertenece. Contiene un botón que nos permite hacer búsquedas de los artículos por su nombre y después agregarlos a la cesta.
- Campos Cliente y Tienda:** La selección de tienda influye en el stock disponible.
- Cuerpo de TPV:** Incluye un campo de texto donde introducir un código de barras para añadir productos a la cesta. Permite modificar las cantidades de los artículos, eliminarlos de la cesta y/o aplicar descuentos.

Hay cuatro operaciones principales que se pueden realizar en la TPV.

- Pagar:** la operación usual. Procesa los artículos y crea un ticket si existe el suficiente stock. Después de hacer click aparece un diálogo con formas de pago.

- 💡 *Esta operación modifica el stock*

- 💡 *Esta operación modifica el flujo de dinero*

- Pagar y crear factura:** el mismo caso anterior, pero en esta ocasión genera una factura con los datos del cliente indicado.

- 💡 *Esta operación modifica el stock*

- 💡 *Esta operación modifica el flujo de dinero*

- Crear albarán:** crea un albarán nuevo, se resta el stock correspondiente pero no se genera ningún pago.

- 💡 *Esta operación modifica el stock*

- 💡 *Esta operación NO modifica el flujo de dinero*

- 💡 *El albarán se factura desde la sección Albaranes*

- Crear pedido:** permite crear pedidos de artículos de los que no hay en stock. El pedido se podrá gestionar desde la sección Pedidos.

- 💡 *Esta operación NO modifica el stock*

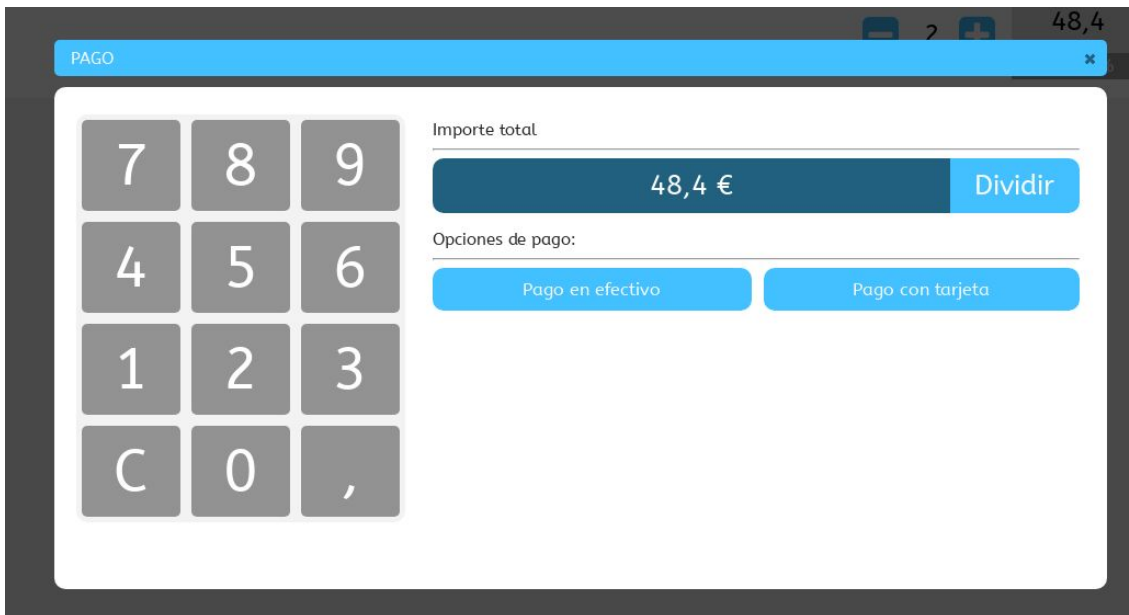
- 💡 *Esta operación NO modifica el flujo de dinero*

- 💡 *Se puede operar sobre el pedido en la sección Pedidos*

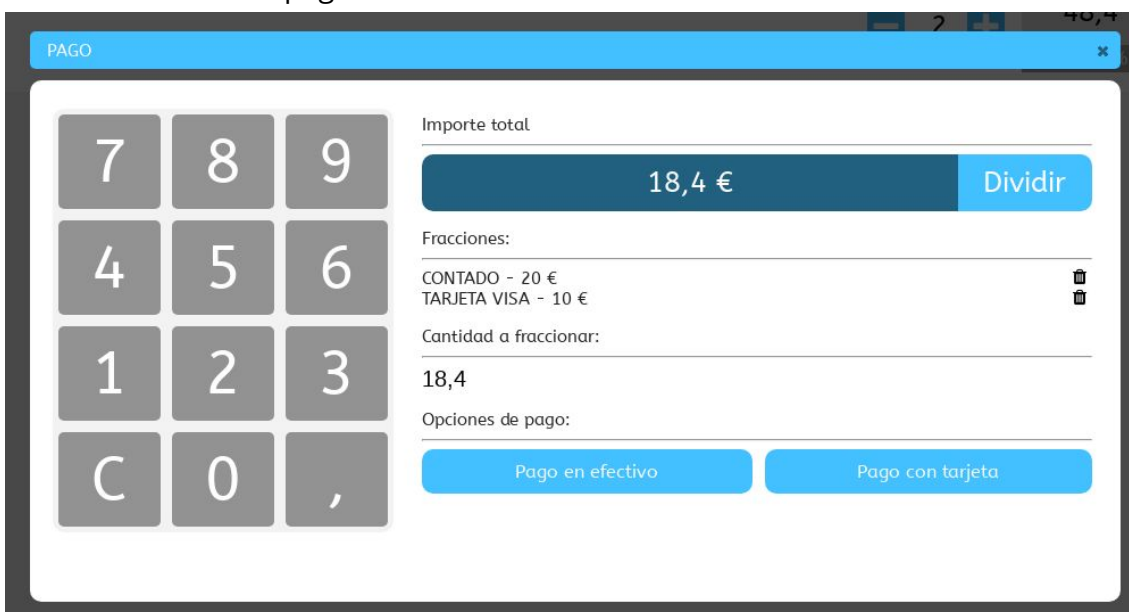
## FORMAS DE PAGO EN LA TPV

Una vez seleccionada la operación deseada, es posible que nos salga un diálogo de pago como el que sigue:





Para efectuar el cobro simplemente hacer click sobre el método de pago deseado.  
Si se desea dividir el pago hacer click sobre Dividir:



- 1) Introducir con el panel numérico la cantidad de la fracción
- 2) Pulsar sobre el método de pago deseado
- 3) Repetir tantas veces sea necesario
- 4) El pago se cierra cuando la suma de las fracciones equivale al importe total

## CONTROL DE STOCK

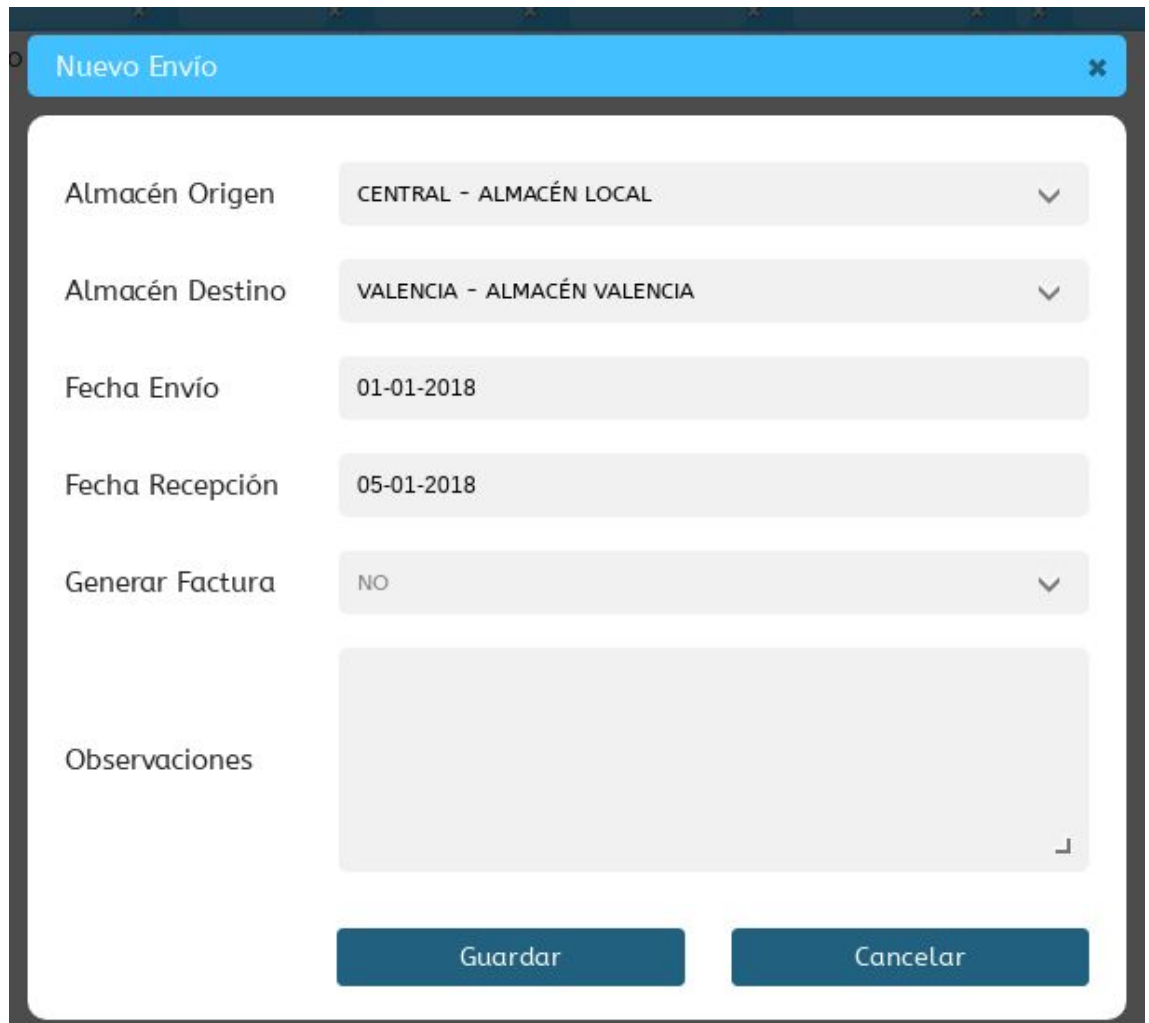
Esta sección está dividida en tres subsecciones:

- **Stock disponible:** muestra todo el stock disponible en todos los almacenes. Permite añadir/editar/eliminar entradas.
- **Entradas de material:** muestra los envíos de los proveedores y los envíos entre almacenes. Solo se puede agregar una entrada mediante el número de envío del proveedor o almacén.
- **Envíos de material:** muestra los envíos de material entre almacenes

## FUNCIONAMIENTO DE ENVÍOS Y ENTRADAS

EL funcionamiento de los **envíos** en Vendeo es sencillo:

- 1) Creamos el “paquete” con los datos necesarios( almacén destino,almacén origen, fecha envío...). El “paquete” queda marcado como “EN ORIGEN”



Nuevo Envío

Almacén Origen CENTRAL - ALMACÉN LOCAL

Almacén Destino VALENCIA - ALMACÉN VALENCIA

Fecha Envío 01-01-2018

Fecha Recepción 05-01-2018

Generar Factura NO

Observaciones

Guardar Cancelar

- 2) Agregamos los modelos deseados al paquete desde la pestaña artículos. El stock del almacén de origen se verá modificado
- 3) Una vez listo, desde la pestaña general marcamos el pedido como “ENVIADO”
- 4) El pedido mantiene el estado de “EN TRANSITO” hasta que se confirme su recepción desde la sección de Entradas de material
- 5) Una vez marcado como recibido el stock del almacén de destino se modifica

El funcionamiento de las **entradas** de material es similar a los envíos.

- 1) Para dar de alta una nueva entrada es necesario un número de paquete, ya sea envíos entre almacenes o envíos de proveedores.

💡 Si no se sabe la numeración exacta, introducir el código de la tienda mostrará todas las entradas pendientes de esa tienda.

- 2) Aceptamos el diálogo. Nos crea una entrada con los datos del “paquete” y en el estado “PROCESANDO”
- 3) Si todos los datos son correctos podemos cerrar el entrada marcandola como “RECIBIDA”
- 4) Al marcarla se modifica el stock en el almacén de destino, se deshabilitan las modificaciones en la entrada y se marca como “EN DESTINO” al envío asociado.

## CLIENTES

### DAR DE ALTA UN CLIENTE

Para dar de alta un cliente:

MENÚ PRINCIPAL->CLIENTES>Nuevo Cliente (icono + en esquina superior derecha).

| NIF       | Nombre | Apellidos | NombreCom    | Teléfono  | Fax | Email                  | Forma Pago |
|-----------|--------|-----------|--------------|-----------|-----|------------------------|------------|
| CAJA      | Caja   | CENTRAL   | CajaTienda   | 963270288 |     | info@tecoquatre.es     | CONTADO    |
| 71641537X | Hugo   | Montoto   | Hugo Montoto | 668676818 |     | hmontoto@tecoquatre.es | CONTADO    |

- 1) Introducimos los datos y guardamos

💡 *El NIF ha de ser correcto para poder guardar el cliente*

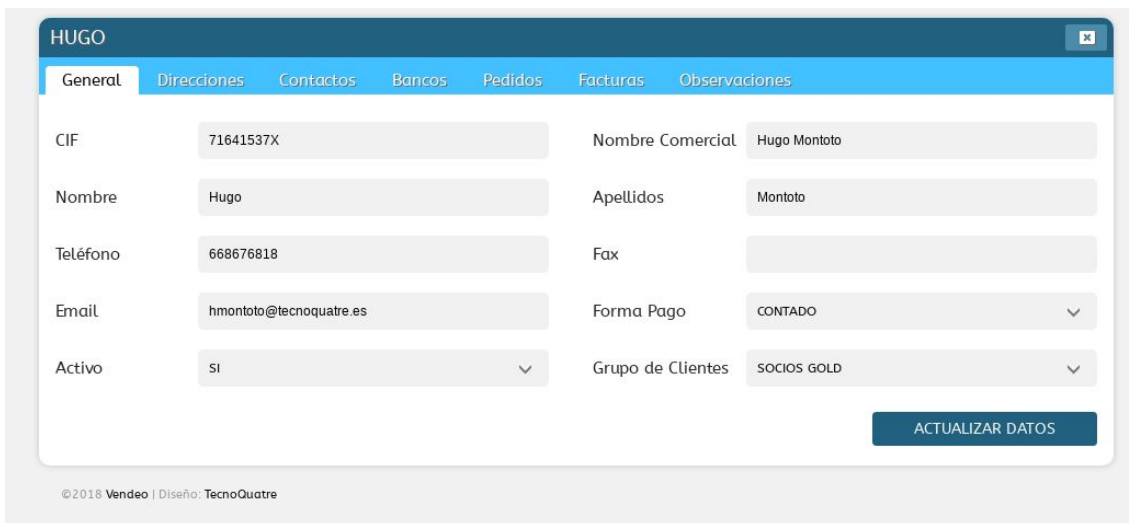
2) Buscamos al cliente en el grid



| NIF       | Nombre | Apellidos | NombreCom    | Teléfono  | Fax | Email                   | Forma Pago |
|-----------|--------|-----------|--------------|-----------|-----|-------------------------|------------|
| 71641537X | Hugo   | Montoto   | Hugo Montoto | 668676818 |     | hmontoto@tecnoquatre.es | CONTADO    |

3) Editamos haciendo doble click.

La ficha del cliente cuenta con varias pestañas.



HUGO

General | Direcciones | Contactos | Bancos | Pedidos | Facturas | Observaciones

CIF: 71641537X | Nombre Comercial: Hugo Montoto

Nombre: Hugo | Apellidos: Montoto

Teléfono: 668676818 | Fax:

Email: hmontoto@tecnoquatre.es | Forma Pago: CONTADO

Activo: SI | Grupo de Clientes: SOCIOS GOLD

ACTUALIZAR DATOS

© 2018 Vendeo | Diseño: TecnoQuatre

- General:** datos generales del cliente
- Direcciones:** cartera de direcciones del cliente
- Contactos:** cartera de contactos del cliente
- Bancos:** cuentas bancarias del cliente
- Pedidos:** listado de pedidos hechos por ese cliente
- Facturas:** listado de facturas a nombre de ese cliente
- Observaciones:** observaciones sobre ese cliente

## TIENDAS

### PASOS PARA DAR DE ALTA UNA TIENDA NUEVA

- 1) Acceder a [Vendeo](#) y entrar en la sección "Dar de alta nueva tienda".
- 2) Introducir:

- a) Id de tienda, no puede contener espacios, es el identificador de la tienda para los administradores dentro de la aplicación.
- b) Nombre de la tienda. Se mostrará en la tienda web.
- c) Usuario y contraseña. Sólo usuarios con permisos especiales pueden dar de alta nuevas tiendas.

Si todo es correcto ya se puede acceder a la nueva tienda desde la ventana anterior. La primera vez que se inicie sesión lo ha de hacer el usuario que ha creado la tienda.

## CONFIGURAR NUEVA TIENDA

- 1) El primer paso es dar de alta una nueva tienda de cliente y asociar un almacén. Tiendas-> Nueva Tienda.
- 2) ALMACENES -> Nuevo Almacén, elegir la tienda creada del desplegable.
- 3) Una vez hecho esto se ha de rellenar los siguientes campos para que la tienda física sea operativa.
  - a) Categorías
  - b) Artículos
  - c) Atributos
- 4) Es recomendable crear usuarios específicos para esta tienda.

## CONFIGURAR TIENDA WEB

Para configurar la tienda web es necesario haber hecho previamente los pasos referentes a la tienda física.

- 1) Acceder al submenú Tienda Web.
- 2) Es posible modificar el logo que se muestra en la web, las redes sociales, la publicidad y el slider mostrados en la página inicial de la tienda web.

TEXTIL   CALZADO   PALOS   BOLAS   SERVICIOS



¡Bienvenido a Vendeo!

Productos Destacados



ADIDAS GORRA Beisbol

Agotado

24,20 €



ZAPATO ADIDAS ADICROSS

En stock

72,00 €



- 1.Logo   2.Accesos a redes sociales.   3.Emplazamiento de la publicidad   4.Slider

Para que estos cambios sean visibles a la web es necesario realizar una serie de cambios en el servidor.

- 1) Copiar el directorio 'public\_html/tienda' a 'public\_html/'

- 2) Renombrar la nueva carpeta creada al nombre deseado, por ejemplo "InterGolf".
- 3) Acceder a 'public\_html/Intergolf/inc/identificadores.php' y modificar las líneas 6 y 7 donde están las variables que apuntan a la nueva tienda:
  - a) \$IdentificadorTienda = 'idTienda' -> proporcionado al crear la tienda
  - b) \$IdentificadorTiendaCliente = 'idTiendaCliente' -> tienda específica de la que se obtendrá los artículos disponibles
- 4) Ya se podría acceder a la nueva tienda desde: vendeo.es/Intergolf

Si se desea acceder desde el formato: Intergolf.vendeo.es es necesario detallarlo en el servidor. Concretamente en Dominios -> Subdominios si se usa cPanel.