



ProcessCore – Aplicación Web para la Planificación de Procesos Productivos

Memoria de Proyecto Final de Grado

Grado Multimedia

Ingeniería Web

<http://www.processcore.com>

Autor: David Nicanor Barraca Iborb

Consultor: Ignasi Lorente Puchades

Profesor: Carlos Casado Martinez

16 de Junio de 2015

Créditos/Copyright

Esta obra está sujeta a la licencia *Creative Commons Reconocimiento – NoComercial – SinObraDerivada* (CC BY-NC-ND)¹: No se permite un uso comercial de la obra original ni la generación de obras derivadas.



Figura 1: CC-BY-NC-SA

Usted es libre de:

Compartir: copiar y redistribuir el material en cualquier medio o formato

El licenciadore no puede revocar estas libertades mientras cumpla con los términos de la licencia.

Bajo las condiciones siguientes:



Reconocimiento (Attribution): Debe reconocer adecuadamente la autoría, proporcionar un enlace a la licencia e indicar si se han realizado cambios. Puede hacerlo de cualquier manera razonable, pero no de una manera que sugiera que tiene el apoyo del licenciadore o lo recibe por el uso que hace.



No Comercial (Non commercial): No puede utilizar el material para una finalidad comercial.



Sin obras derivadas (No Derivate Works): Si remezcla, transforma o crea a partir del material, no puede difundir el material modificado.

No hay restricciones adicionales: No puede aplicar términos legales o medidas tecnológicas que legalmente restrinjan realizar aquello que la licencia permite.

Avisos:

No tiene que cumplir con la licencia para aquellos elementos del material en el dominio público o cuando su utilización esté permitida por la aplicación de una excepción o un límite.

No se dan garantías. La licencia puede no ofrecer todos los permisos necesarios para la utilización prevista. Por ejemplo, otros derechos como los de publicidad, privacidad, o los derechos morales pueden limitar el uso del material.

¹ Texto completo de la licencia: <http://creativecommons.org/licenses/by-nc-nd/3.0/es/>

Dedicatoria

En primer lugar me gustaría dar las gracias a mi familia. A mi padre Nicanor, a mi madre Cristina, a mi hermano Roberto, y a mi novia Mamen, que me han apoyado en todo momento durante esta etapa. Así como por el esfuerzo que han realizado para poder formarme como persona y poder llegar a este punto de mi carrera, y finalmente concluir mis estudios universitarios. Todos ellos han sufrido conmigo cada uno de mis mejores y peores momentos. Espero poder corresponderos en un futuro de la misma manera.

También agradecer a todos los tutores, consultores y profesores de cada una de las asignaturas cursadas durante el grado que me han permitido un constante aprendizaje, sin los cuales no hubiese podido concluir este proyecto.

Agradecer a todos mis amigos, primos, tíos y demás familia también su cariño mostrado y que han hecho esta etapa de mi vida más agradable.

Sin todos ellos, esto nunca habría sido posible.

Muchísimas gracias a todos.

Abstract

Hoy en día multitud de empresas están obligadas a tener un control exhaustivo y eficaz en los procesos de planificación, y muchas de las soluciones que existen o no se adaptan a sus necesidades o una aplicación creada a su medida no es asumible económicamente. Por este motivo, este trabajo final de grado consiste en la creación de una aplicación *web* llamada ProcessCore que permita el control de los diferentes procesos de planificación, y en parte, adaptable a las necesidades del cliente.

Debido a que la magnitud del proyecto ProcessCore es bastante grande (gestión de pedidos, facturación, inventarios, almacenes, expediciones, confección de hojas de trabajo (*kanbans*), control de estado y tiempo de cada proceso mediante la inserción a través de lectores de códigos de barras, costes de producción, y multitud de acciones derivadas de los datos obtenidos), en este proyecto se va a presentar la primera fase que hace referencia a los inventarios y almacenes. Si el avance es favorable se intentará implantar la segunda fase, que hace referencia al control de pedidos y procesos productivos.

Palabras clave: Memoria, Trabajo de Fin de Grado, Ingeniería Web, Multimedia, Planificación, Proceso Productivo, Inventarios, Almacenes, Productividad, ProccesCore, Aplicación web

Notaciones y Convenciones

Título 1: Arial 24 Negrita

Título 2: Arial 18 Negrita

Título 3: Arial 12 Negrita

Texto en general: Arial 10 – Interlineado 1,5

Expresiones en otro idioma: Cursiva

Citas y notas a pie de página: Arial 8 – Interlineado sencillo

Código fuente: Courier New 8

Índice

1. Introducción/Prefacio	9
2. Descripción/Definición/Hipótesis	10
3. Objetivos	11
3.1 Principales	11
3.2 Secundarios	11
4. Marco teórico/Escenario	11
5. Contenidos	13
6. Metodología	19
7. Arquitectura de la aplicación/sistema/servicio	19
8. Plataforma de desarrollo	22
9. Planificación	23
10. Proceso de trabajo/desarrollo	24
11. APIs utilizadas	25
12. Diagramas UML	27
13. Prototipos	44
13.1 Lo-Fi	44
14. Perfiles de usuario	56
15. Usabilidad/UX	56
16. Seguridad	59
17. Tests	60
18. Versiones de la aplicación/servicio	61
19. Requisitos de instalación/implantación/uso	62
20. Instrucciones de instalación/implantación	64
21. Instrucciones de uso	65
22. Proyección a futuro	73
23. Conclusión/-es	74
Anexo 1. Entregables del proyecto	75
Anexo 2. Código fuente (extractos)	76
Anexo 3. Capturas de pantalla	90
Anexo 4. Libro de estilo	95
Anexo 5. Bibliografía	96

Figuras y tablas

Lista de imágenes, tablas, gráficos, diagramas, etc., numeradas, con títulos y las páginas en las que aparecen.

Índice de figuras

Figura 1: CC-BY-NC-SA	2
Figura 2: Arquitectura en tres capas	19
Figura 3: Diagrama de Gantt.....	¡Error! Marcador no definido.
Figura 4: FPDF Library.....	24
Figura 5: PHPMailer.....	24
Figura 6: Barcode Keyboard	25
Figura 7: Casos de uso referentes al sistema	26
Figura 8: Casos de uso generales	28
Figura 9: Casos de uso referentes a movimientos teóricos	34
Figura 10: Casos de uso referente a movimientos físicos.....	39
Figura 11: Prototipo Lo-Fi Login.....	44
Figura 12: Prototipo Lo-Fi Menú principal	44
Figura 13: Prototipo Lo-Fi Menú Gestión de usuarios.....	45
Figura 14: Prototipo Lo-Fi Nuevo usuario	45
Figura 15: Prototipo Lo-Fi Gestión alertas	46
Figura 16: Prototipo Lo-Fi Editar alerta	46
Figura 17: Prototipo Lo-Fi Gestión de productos.....	47
Figura 18: Prototipo Lo-Fi Nuevo producto	47
Figura 19: Prototipo Lo-Fi Gestión categorías	48
Figura 20: Prototipo Lo-Fi Inventario teórico	48
Figura 21: Prototipo Lo-Fi Inventario físico	49
Figura 22: Prototipo Lo-Fi Gestión almacenes.....	49
Figura 23: Prototipo Lo-Fi Nuevo almacén.....	50
Figura 24: Prototipo Lo-Fi Gestión estanterías	50
Figura 25: Prototipo Lo-Fi Movimiento teórico	51
Figura 26: Prototipo Lo-Fi Nuevo movimiento teórico	51
Figura 27: Prototipo Lo-Fi Movimiento físico - ENTRADA.....	52
Figura 28: Prototipo Lo-Fi Movimiento físico - SALIDA.....	52
Figura 29: Prototipo Lo-Fi Login (Smartphone).....	53
Figura 30: Prototipo Lo-Fi Menú principal (Smartphone)	53
Figura 31: Prototipo Lo-Fi Movimiento físico - ENTRADA (Smartphone).....	54
Figura 32: Prototipo Lo-Fi Movimiento físico - SALIDA (Smartphone).....	54
Figura 33: Prototipo Lo-Fi Inventario físico (Smartphone).....	55
Figura 34: Menú administrador	59
Figura 35: Menú gestor	59
Figura 36: Menú operario.....	59
Figura 37: Expresiones regulares	59
Figura 38: Email alerta fuera de stock.....	69

Figura 39: Email alerta punto de pedido	69
Figura 40: Código Login (controlador).....	76
Figura 41: Código Login (vista)	77
Figura 42: Código Conexión (modelo)	77
Figura 43: Código Página principal (controlador).....	78
Figura 44: Código Página principal (vista) 1ª parte	78
Figura 45: Código Página principal (vista) 2ª parte	79
Figura 46: Código Página principal (vista) 3ª parte	80
Figura 47: Código Producto (modelo) 1ª parte	81
Figura 48: Código Producto (modelo) 2ª parte	82
Figura 49: Código Producto (modelo) 3ª parte	83
Figura 50: Código Inventario teórico (controlador)	83
Figura 51: Código Inventario teórico (vista)	84
Figura 52: Código Imprimir etiquetas de palet (controlador) 1ª parte.....	85
Figura 53: Código Imprimir etiquetas de palet (controlador) 2ª parte.....	86
Figura 54: Pantalla menú principal.....	88
Figura 55: Pantalla nuevo producto	88
Figura 56: Pantalla inventario teórico.....	89
Figura 57: Pantalla movimiento teórico	89
Figura 58: Pantalla inventario físico	90
Figura 59: Pantalla gestión de estanterías.....	90
Figura 60: Etiquetas de palet	91
Figura 61: Etiquetas de estantería	91
Figura 62: Pantalla Smartphone menú principal	92
Figura 63: Pantalla Smartphone inventario físico.....	92
Figura 64: Pantalla Smartphone movimiento entrada	92
Figura 65: Pantalla Smartphone movimiento salida.....	92
Figura 66: Logo sobre fondo blanco.....	93
Figura 67: Logo sobre fondo oscuro	93
Figura 68: Iconos web.....	93

Índice de tablas

Tabla 1: Lenguajes capa de arquitectura	20
Tabla 2: Lenguajes cliente, servidor y bbdd.....	20
Tabla 3: Diagrama de clases UML.....	41
Tabla 4: Diagrama base de datos	42

1. Introducción

Durante estos últimos cuatro años la Universtat Oberta de Catalunya me ha brindado la posibilidad de realizar mis estudios y a su vez continuar con mi relación laboral. Estoy en la fase final del Grado Multimedia, en el cual he optado por la mención en ingeniería *web* que me ha dotado de competencias en el desarrollo de aplicaciones siguiendo las reglas de la ingeniería del *software*.

En mi actual entorno laboral disponemos de un *software* personalizado para la gestión íntegra de los procesos de producción de diferentes productos. Este programa aporta multitud de tareas: gestión de pedidos, gestión de clientes, gestión de proveedores, control de inventarios, movimientos de almacenes, confección de hojas de trabajo (*kanban*), control de tiempos en procesos de producción, expediciones, control de costes, facturación...

Por diferentes motivos nuestro proveedor del *software* mencionado nos ha comunicado que va a dejar de prestar servicio con el curso del tiempo, y por lo tanto, no realizará más actualizaciones ni corregirá errores que se puedan ocasionar. En el mercado actual no hay *softwares* que se adapten a nuestras necesidades, y los *softwares* hechos a medida a parte de tener unos costes muy elevados, estamos expuestos al mismo problema tratado.

Debido a esto, he propuesto realizar un *software* con tecnología web (PHP, MYSQL, HATML5, CSS3, Javascript...) en un servidor local como Wampserver, que ofrezca las mismas y nuevas herramientas de una manera más eficiente y actualizadas. Se construirá basándose en las reglas de la ingeniería del *software* y de la programación orientada a objetos. También se buscará la mejor manera para implementar una buena accesibilidad y usabilidad que ofrezca una buena experiencia de usuario, carente en muchos de los *softwares* que existen hoy en día en este campo.

El *software*, en un primer momento, se denominará ProcessCore y estará dirigido a pequeñas y medianas empresas. Por un lado, se intentará construir de tal forma que sea adaptable para diferentes clientes, es decir, personalizable. Por otro lado, a través de diferentes módulos, el usuario podrá activar o desactivar diferentes opciones a la hora de disponer de unas u otras herramientas.

Finalmente, como el proyecto ProcessCore es bastante amplio, en este Trabajo de Final de Grado se implementará la primera fase que corresponde a la gestión de inventarios y almacenes. Si la situación a lo largo del proyecto es favorable se intentará implantar la segunda fase, que hace referencia a la gestión de clientes, proveedores, albaranes, facturas y expediciones.

2. Descripción

Hoy en día la mayoría de empresas usan multitud de nuevas tecnologías con el fin de obtener, entre otros beneficios, unos mejores procesos productivos, aumento de la calidad de servicio o producto ofrecido, control de costes, etc. Por esto, las compañías demandan tecnologías de gestión y control adaptadas a sus necesidades, y que en muchas ocasiones requieren actualizaciones para su continuo crecimiento y adaptación a nuevas posibilidades.

La idea de negocio consiste en crear una aplicación *web* para la planificación de procesos productivos orientada a la pequeña y mediana empresa. Esta aplicación denominada ProcessCore tiene como finalidad la gestión y control de todo el proceso productivo: gestión de pedidos, facturación, inventarios, almacenes, expediciones, confección de hojas de trabajo (*kanbans*), control de estado y tiempo de cada proceso mediante la inserción a través de lectores de códigos de barras, costes de producción, y multitud de acciones derivadas de los datos obtenidos. En este trabajo de final de grado se va a presentar la primera fase que hace referencia a los inventarios y almacenes, y si la situación es favorable se implantará la segunda fase (control de pedidos y procesos productivos).

Por un lado, la aplicación dispondrá de lo que se denomina “Inventario Teórico”, que es donde se introducirán todos los movimientos de productos (entradas, salidas, recuentos de inventario...). De esta manera se podrá visualizar diferentes datos de cada producto: *stock* total, *stock* en curso, *stock* disponible, movimientos, etc. Para cada entrada se dispondrá la opción de imprimir una etiqueta que irá adherida a cada producto o conjunto de productos (por ejemplo un palet), y que servirá para introducirlo en los almacenes. De la misma manera se podrán crear categorías de productos.

Por otro lado, se encuentra lo que se denomina “Inventario Físico”, que estará formado por almacenes, los cuales contendrán estanterías y estas a su vez diferentes posiciones. Se podrán crear tantos almacenes, estanterías y posiciones necesarios. Cada posición tendrá un código de barras que hará referencia a su situación y por el cual se podrá introducir productos. Si un producto no está introducido en una posición, se marcará en el inventario como “Sin posicionar”. Es decir, desde este inventario se podrá controlar el stock físico o real.

El “Inventario Teórico” dispondrá de la función de envío de *email* cuando rebase el punto de pedido previamente definido para cada producto, así como cuando el *stock* disponible sea inferior a 0. De la misma manera, cuando exista un desajuste de un mismo producto entre el “Inventario Teórico” y el “Inventario Físico” que previamente se habrá definido en un porcentaje, también tendrá la función del aviso mencionado anteriormente.

Finalmente la aplicación *web* ProccesCore se construirá a partir de tecnología web (PHP, MYSQL, , HTAML5, CSS3, Javascript...) en un servidor local como Wampserver, basándose en las reglas de la ingeniería del software y de la programación orientada a objetos.

3. Objetivos

El objetivo principal de este trabajo de fin de grado es el desarrollo de la primera fase concerniente a la gestión de inventarios y almacenes de la aplicación ProcessCore, que tiene como finalidad completar la aplicación *web* de planificación de procesos productivos.

3.1 Principales

Objetivos clave del TFG:

- Crear la aplicación *web* “base” de ProcessCore.
- Implantar el control y gestión de usuarios.
- Implantar el control y gestión de productos.
- Implantar el control y gestión de almacenes y estanterías.
- Implantar inventarios y movimientos teóricos.
- Implantar inventarios y movimientos físicos.

3.2 Secundarios

Objetivos adicionales que enriquecen el TFG y que pueden sufrir variaciones.

- Realizar una aplicación *web* con diseño *responsive* para asegurar una buena visualización en dispositivos móviles para dar entrada y salida a productos en los almacenes.
- Permitir y gestionar el envío de notificaciones vía email sobre eventos (fuera de stock, punto de pedido...) a los usuarios a través de la aplicación.
- Implantar la impresión de etiquetas de estantería.
- Implantar la impresión de etiquetas de palet.

4. Marco teórico/Escenario

Aproximadamente hará unos seis años, la empresa en la que actualmente estoy contratado, firmó un acuerdo con personal externo para el desarrollo de un software propio, es decir, hecho a medida para las necesidades de su producción. Ahora cumplirán tres meses que el proveedor comunicara a la empresa que abandonaba el desarrollo y corrección de posibles errores en el plazo de un año.

Actualmente para la compañía es indispensable disponer de un *software* o aplicación que satisfaga todas las necesidades en la gestión de la producción, y de esta manera aumentar su productividad en un mercado tan competitivo. Hoy en día es imprescindible disponer de una minuciosa y buena gestión en cada proceso productivo.

Debido a esto, primeramente se opta por buscar una empresa externa que siga con el mantenimiento y desarrollo del programa. Pero la mayoría de ellas no están dispuestas a continuar un proyecto tan grande, ya que pueden tener bastantes problemas al no conocer exactamente su desarrollo, además de un lenguaje de programación diferente al que ofertan. Solamente ofrecen la implantación de un nuevo *software* creado por ellos, lo que conlleva una inversión muy grande y estar expuestos al problema que se ha generado desde un principio.

Por otro lado, se estudia el mercado de empresas que ofrecen *softwares* dedicados a procesos productivos generalistas, donde el mantenimiento y corrección de errores están asegurados. Pero el gran problema que se detecta es la falta de muchas funcionalidades que no pueden ser adaptadas a las necesidades de la compañía.

Por todo esto, la elección de la temática de este trabajo de fin de grado está orientada al inicio del desarrollo de una nueva aplicación que se gestionará desde la propia empresa. En un principio, y lo que es concerniente a este TFG es la implantación de la primera fase, que corresponde a la gestión de inventario y almacenes. Además, entre otras funcionalidades como la gestión de alertas en stocks o desajustes en inventarios, también se implantará la gestión de usuarios, productos y categorías.

5. Contenidos

La aplicación web está formada por diferentes módulos y se presentarán unos u otros dependiendo del usuario que acceda al sistema (administrador, gestor u operario). La aplicación se realizará con diseño *responsive*, así de esta manera alcanzar una correcta visualización en distintos dispositivos (desde ordenadores de escritorio a *tablets* y móviles).

Los administradores podrán visualizar todos los módulos de la aplicación *web*, mientras que los gestores y operarios accederán solamente a módulos específicos de sus actividades. A continuación se detallan las páginas, así como los módulos a los que los usuarios pueden acceder.

Página Login

La página principal será accesible a todos los usuarios y mostrará los siguientes campos a introducir:

- Email
- Contraseña

En el fondo de la página aparecerá el logo e imágenes asociadas al programa.

Página Principal

Una vez el usuario ha accedido al sistema se visualizará la página principal, donde aparecerá un menú principal adaptado a cada tipo de usuario, es decir, según seas administrador, gestor u operario se cargarán unos u otros accesos. De la misma manera estará disponible para todos los usuarios la posibilidad del *logout*.

La cabecera permanecerá invariable en todas las secciones, así dispondrá a su izquierda el logo y el nombre del usuario que ha entrado al sistema acompañado del *logout*. En la parte central se alojará el título de la aplicación web, y finalmente a la derecha se insertará un reloj que indicará la hora, año mes y día.

Al igual que la cabecera, el pie se mostrará invariable y alojará diferentes opciones: enviar una consulta, consultar la ayuda y visualizar versiones. Del mismo modo, desde este espacio se tendrá acceso a las diferentes redes sociales en las que se encuentra activa la aplicación, así como la comprobación del cumplimiento de los estándares *web* W3C.

El cuerpo constará de dos secciones, el contenido principal donde se llevarán a cabo las principales actividades de la aplicación, y un contenido secundario donde se incluirán diferentes accesos directos e información relevante para el usuario².

Módulo “Gestión de Usuarios”

Acceso: Administrador.

Esta opción permitirá al administrador crear, editar y eliminar usuarios, así como de dotarles de mayores o menores accesos según el tipo de usuario elegido. La página mostrará la opción de dar de alta un usuario, editar un usuario y borrar un usuario.

Para la opción dar de alta un usuario mostrará un formulario a rellenar donde se introducirán todos los datos exigidos y que se detallan a continuación:

- Nombre
- Apellido
- Tipo de usuario (administrador, gestor u operario)
- Contraseña
- Email
- Teléfono

Para las opciones de editar o borrar un usuario, aparecerá el listado de usuarios y el icono que permitirá editar o borrar según la opción que se elija.

Módulo “Gestión de Alertas”

Acceso: Administrador.

Las alertas disponibles son:

- Fuera de stock. Cuando la cantidad de producto en el inventario teórico sea 0.
- Punto de pedido. Cuando la cantidad de punto de pedido establecida cuando se crea un producto es inferior a ésta.

Esta opción permitirá al administrador editar las alertas pero de una manera controlada, es decir, solamente podrá añadir o eliminar las cuentas de correo que recibirán las notificaciones.

² Consultar: Anexo 3. Capturas de pantalla. [Figura 54 Pantalla menú Principal](#).

Módulo “Gestión de Categorías”

Acceso: Administrador y gestor.

Esta opción permitirá al administrador y gestor: crear, editar y eliminar categorías. Aparecerá su acceso dentro del módulo “Gestión de productos”.

La opción crear categoría mostrará un formulario donde se dará nombre a la nueva categoría:

- Nombre

Mientras, las opciones de editar y eliminar categorías se mostrarán junta al listado de todas ellas, así una vez seleccionada la acción se llevará a cabo.

Módulo “Gestión de Productos”

Acceso: Administrador y gestor.

Esta opción permitirá al administrador y gestor: crear, editar y eliminar productos. De la misma manera se visualizará el listado de productos, donde se podrán filtrar por categorías, así como realizar la búsqueda por su nombre o código.

La opción crear productos permitirá dar de alta nuevos productos mediante la cumplimentación del formulario con los siguientes campos:

- Seleccionar categoría (listado de categorías existentes)
- Código
- Descripción
- Punto de pedido

Las opciones de editar y eliminar productos se mostrarán junto al listado de todos ellos, así una vez seleccionado permitirá realizar la acción que se quiera llevar a cabo³.

³ Consultar: Anexo 3. Capturas de pantalla. [Figura 55 Pantalla nuevo producto.](#)

Módulo “Movimientos Teóricos”

Acceso: Administrador y gestor.

Esta opción permitirá realizar movimientos teóricos (salidas y entradas) de todos los productos.

Por un lado, se mostrará el listado de movimientos teóricos realizados junto con las opciones de editar o eliminar. Por otro lado, estará la opción de crear un nuevo movimiento teórico al cual se le podrán añadir diferentes líneas de movimiento.

En primer lugar se añadirán tantas líneas como sean necesarias, una por producto, donde a través de una ventana emergente se cumplimentarán los siguientes campos:

- Producto
- Tipo de movimiento (entrada o salida)
- Cantidad

En segundo lugar, una vez añadidas todas las líneas necesarias se cumplimentarán los campos concernientes al movimiento:

- Fecha
- Referencia
- Observaciones

Finalmente, para cada línea de movimiento también tendrá las opciones de editar, eliminar e imprimir etiquetas de palet para cada producto, las cuales serán de utilidad para el control de inventario físico⁴.

Módulo “Imprimir Etiquetas Palet”

Acceso: Administrador y gestor.

Esta opción, disponible en cada línea de movimiento teórico, permitirá imprimir para cada entrada de producto las etiquetas para cada palet. Una vez seleccionado mostrará la descripción del producto, su código, cantidad total de entrada y se solicitará la cantidad por palet.

De esta manera se generará un pdf⁵ con las diferentes etiquetas para añadir a cada palet en las que se mostrará los siguientes datos:

- Descripción del producto

⁴ Consultar: Anexo 3. Capturas de pantalla. [Figura 57 Pantalla Movimiento Teórico](#).

⁵ Consultar: Anexo 3. Capturas de pantalla. [Figura 60 Pantalla Etiquetas de palet](#).

- Código
- Cantidad del palet
- Nombre del palet
- Código de barras (nombre del palet)

Módulo “Inventario Teórico”

Acceso: Administrador y gestor.

Esta opción permitirá visualizar principalmente el *stock* total, el *stock* disponible y el *stock* en curso de cada producto, así como la búsqueda de productos por código o descripción, o filtrar por categorías. Finalmente se añadirá el campo de stock físico para permitir la comparación y posibles desajustes entre inventarios de una manera más sencilla⁶.

Módulo “Inventario Físico”

Acceso: Administrador y gestor.

Esta opción permitirá visualizar el *stock* de los productos en el inventario físico. Por un lado se mostrará la posición de cada palet en el almacén agrupados por producto, así como la suma total. Por otro lado, si el palet no está asignado a una posición aparecerá con el texto “Sin posicionar”. Finalmente se podrá realizar una búsqueda por código o descripción de producto y por nombre de palet⁷.

Módulo “Gestión de almacenes”

Acceso: Administrador.

Esta página permitirá crear nuevos almacenes, estanterías y posiciones. Para crear los almacenes se solicitarán los siguientes datos:

- Nombre

⁶ Consultar: Anexo 3. Capturas de pantalla. [Figura 56 Pantalla Inventario teórico.](#)

⁷ Consultar: Anexo 3. Capturas de pantalla. [Figura 58 Pantalla Inventario físico.](#)

Para crear las estanterías y posiciones se solicitarán los siguientes datos:

- Almacén (seleccionado del listado)
- Descripción
- Código
- Número de filas
- Número de columnas

Finalmente se tendrá la opción de imprimir las etiquetas de cada posición seleccionando la opción de imprimir posiciones de cada estantería. De la misma manera, también estará disponible para cada almacén y estantería la opción de editar o eliminar⁸.

Módulo “Movimientos Físicos”

Acceso: Administrador y operario.

Esta opción permitirá realizar movimientos físicos (salidas y entradas) de cantidad de producto existente en cada palet.

Por un lado para las entradas⁹ se solicitarán los siguientes datos:

- Palet
- Posición

Por otro lado para las salidas¹⁰ se solicitarán los siguientes datos:

- Palet
- Cantidad

⁸ Consultar: Anexo 3. Capturas de pantalla. [Figura 59 Pantalla Gestión de estanterías.](#)

⁹ Consultar: Anexo 3. Capturas de pantalla. [Figura 64 Pantalla Movimiento físico de entrada.](#)

¹⁰ Consultar: Anexo 3. Capturas de pantalla. [Figura 65 Pantalla Movimiento físico de salida.](#)

6. Metodología

Actualmente la empresa en la que trabajo dispone de un software de gestión basado en Visual Fox Pro que permite realizar diferentes funciones en los procesos productivos. Sin embargo, como se ha comentado anteriormente, el proveedor deja de ofrecer sus servicios y no se dispondrá de nuevas mejoras ni corrección de errores. Por esto, se ha tomado la decisión de realizar el software desde cero, y así tener un control y conocimiento total de su desarrollo.

A partir de este software y de las opiniones de los usuarios actuales se recogerán la mayor parte de las ideas y funciones que se van a aplicar en el nuevo sistema. Además se realizará una investigación de los programas similares existentes hoy en día, así como una búsqueda de información de dicha temática. Una vez seleccionadas las funciones a desarrollar se estudiarán los requisitos necesarios para desarrollar la aplicación web, como son: el servidor local, los lenguajes de programación, maquetado, diseño y bases de datos, así como APIs o librerías que ayuden a su desarrollo.

De esta manera, una vez realizado este análisis, selección e interpretación de los datos, se empezará a trabajar con las tecnologías seleccionadas. Para cada nuevo avance se volverá a realizar un análisis de los resultados y sus aplicaciones.

Finalmente se realizarán los test de funcionamiento de todos los procesos ejecutados.

7. Arquitectura de la aplicación

La aplicación web se basa en el modelo de arquitectura en tres capas, y de esta manera separar los servicios de gestión de datos, las funcionalidades del dominio y la interfaz de usuario. Así, cada componente del sistema forma parte de una de las siguientes capas:

- 1) Capa de presentación
- 2) Capa del dominio
- 3) Capa de gestión de datos

Cada capa se comunica con la capa inmediatamente superior y con la capa inmediatamente inferior, tal y como muestra la figura.

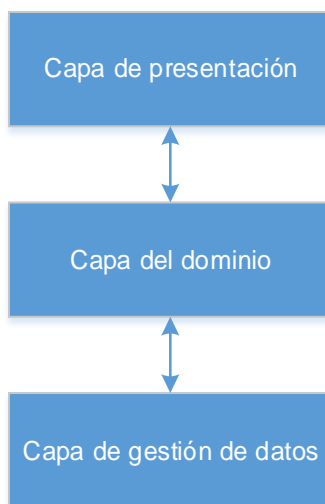


Figura 2: Arquitectura en tres capas

La capa de presentación, también llamada interfaz gráfica, gestionará la comunicación con el usuario, es decir, tendrá la responsabilidad de capturar las peticiones del usuario y mostrarle la información correspondiente. Los componentes de esta capa se comunicarán con la capa del dominio.

La capa de dominio, también llamada capa de negocio, implementará las clases de objetos que permiten representar el estado del dominio. Los componentes de esta capa se comunicarán con la capa de gestión de datos y con la capa presentación.

La capa de gestión de datos, también llamada capa de persistencia, tendrá la responsabilidad de mantener los datos del sistema (base de datos), es decir, debe crear, modificar o eliminar datos a petición de la capa de dominio.

Así pues, se muestra en la tabla 1 los lenguajes que se utilizará para cada capa de la arquitectura:

Lenguajes de marcado y programación						
Capas		HTML5	CSS3	Javascript	PHP	MySQL
	Capa de presentación	x	x	x	x	
	Capa de dominio				x	
	Capa de gestión de datos				x	x

Tabla 1: Lenguajes capa de arquitectura

Finalmente, en la tabla 2 se muestran los lenguajes utilizados en el lado del cliente, servidor y base de datos:

Lenguajes de marcado y programación					
	HTML5	CSS3	Javascript	PHP	MySQL
Cliente	x	x	x		
Servidor				x	
Base de datos					x

Tabla 2: Lenguajes cliente, servidor y bbdd

8. Plataforma de desarrollo

Para la construcción de la aplicación web se han utilizado diversos recursos tecnológicos: equipos informáticos, lenguajes de programación, programas de maquetado, servidor local, base de datos, así como diversas herramientas específicas, los cuales se presentan a continuación:

Lenguajes de marcado y programación:

- HTML 5
- CSS 3
- Javascript
- PHP
- MySQL

Programas:

- Microsoft Office: Word, MS Project y Visio
- Adobe CS6: Photoshop e Illustrator
- Notepad++
- Magic Draw 17.0.5 (Diagramas UML)
- Workbench GPL 4.2.47 (Diagramas BBDD)

Herramientas en línea:

- Diagramas online Gantter: www.gantter.com

Otras herramientas:

- Servidor wampserver
- Ordenador de sobremesa y portátil.

9. Planificación

La planificación de las actividades y tareas a realizar en este trabajo de final de grado son el proceso que se deriva de la definición de los hitos del proyecto. Los hitos son situaciones, productos, estadios del trabajo. Las actividades son acciones necesarias para alcanzar los hitos. Las tareas son acciones dentro de cada actividad.

Para representar este plan de actividades se ha seleccionado el diagrama de Gantt, el cual muestra el tiempo en el eje de las abscisas, mientras que en el eje de ordenadas se disponen todas las actividades que forman el proyecto. De esta manera, se ha utilizado para realizar esta representación de forma gráfica de la aplicación web, la herramienta de diagramas online Gantter <http://www.gantter.com> y MS Project.

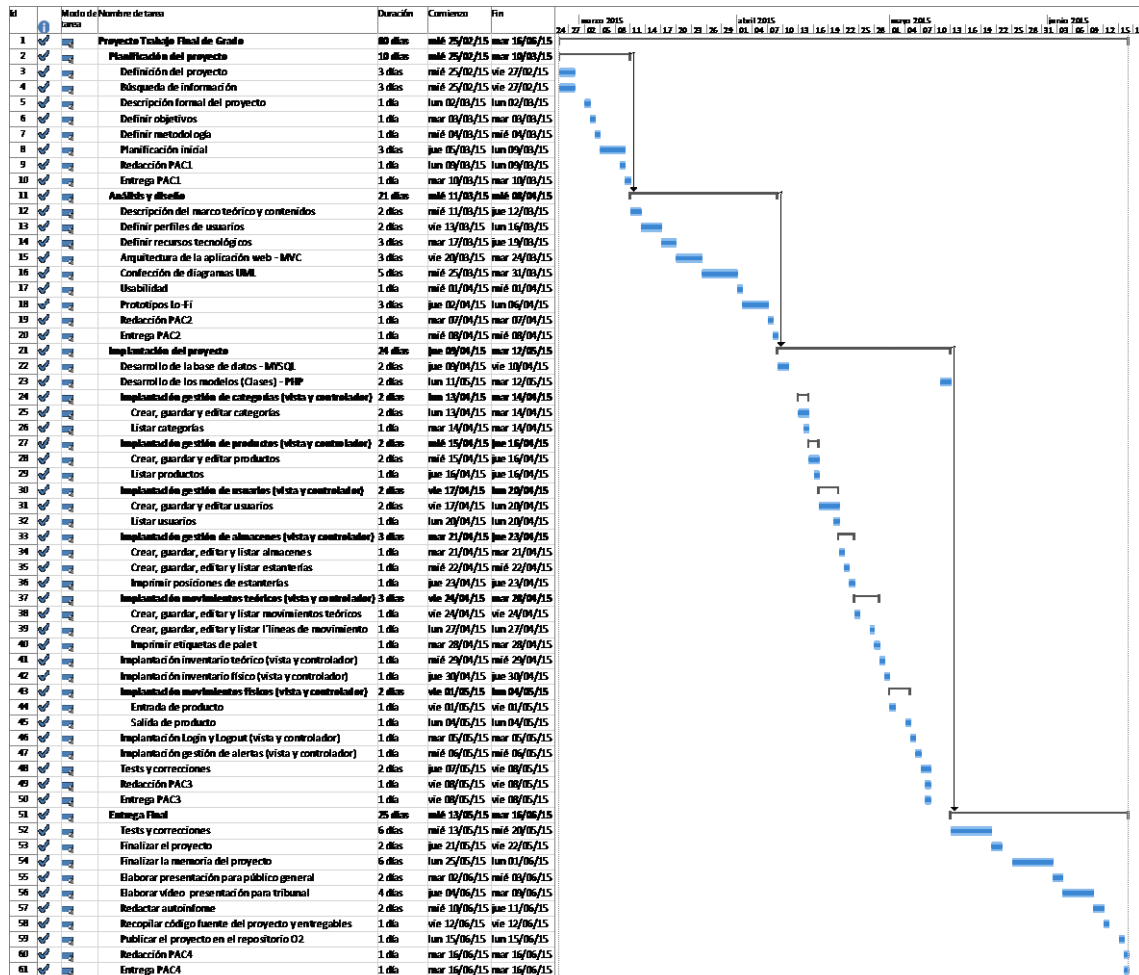


Figura 3: Diagrama de Gantt

10. Proceso de trabajo/desarrollo

El proceso del proyecto de trabajo de final de grado está estructurado en diferentes fases, las cuales se han definido en el diagrama de Gantt. De esta manera se inicia una propuesta de proyecto hasta su total puesta en marcha. A continuación se exponen los principales hitos del proyecto:

- Planificación del proyecto: definición del proyecto, búsqueda de información, descripción formal del proyecto, definición de objetivos y la planificación inicial.
- Análisis y diseño: comprende la descripción del marco teórico y contenidos, definición de perfiles de usuarios y recursos tecnológicos, definición de la arquitectura de la aplicación, diagramas UML y usabilidad.
- Implantación del proyecto: se ha definido en diferentes hitos por su mayor complejidad:
 - o Desarrollo de la base de datos
 - o Desarrollo de las clases (modelos)
 - o Implantación “gestión de categorías” (vista y controlador)
 - o Implantación “gestión de productos” (vista y controlador)
 - o Implantación “gestión de usuarios” (vista y controlador)
 - o Implantación “gestión de almacenes” (vista y controlador)
 - o Implantación “movimientos teóricos” (vista y controlador)
 - o Implantación “movimientos físicos” (vista y controlador)
 - o Implantación “inventario teórico” (vista y controlador)
 - o Implantación “inventario físico” (vista y controlador)
 - o Implantación “login & logout” (vista y controlador)
 - o Implantación “gestión de alertas” (vista y controlador)
- Entregables:
 - o Memoria del proyecto: se realiza según avanza el proyecto.
 - o Autoinforme: se confecciona al final del proyecto.
 - o Video presentación: se confecciona al final del proyecto.
 - o Presentación en Prezi: se confecciona al final del proyecto.
 - o Proyecto: se entrega al final con todos los archivos generados.

11. APIs utilizadas

Para llevar a cabo el proyecto se han utilizado las siguientes APIs y librerías:

FPDF Library



Figura 4: FPDF Library

FPDF es una clase escrita en PHP que permite generar documentos PDF directamente desde PHP, es decir, sin usar la biblioteca PDFlib.

Se ha utilizado para crear las etiquetas concernientes a la identificación de palets y de posiciones de estanterías. Se incluye también la clase “code128” que permite la conversión a códigos de barras, y de esta manera poder ser leídos por lectores de códigos de barra y terminales con aplicaciones destinadas a este uso.

PHPMailer



Figura 5: PHPMailer

PHPMailer es una potente y sencilla clase para envío de email desde PHP. Su uso en el proyecto se ha presentado a la hora la notificación de alertas mediante el envío de estas vía email. También se utiliza para el formulario de contacto.

Barcode Keyboard



Figura 6: Barcode Keyboard

Barcode Keyboard es una aplicación para sistemas Android que proporciona un teclado con escáner de código de barras integrado. De esta manera en nuestra aplicación permitirá escanear códigos de barras de palets y posiciones, principalmente para la entrada y salida de productos del inventario físico.

12. Diagramas UML

12.1 Casos de uso

A continuación se expone los principales casos de uso:

Este subsistema representa todos los casos de uso referentes a las tareas asociadas al **Sistema**:

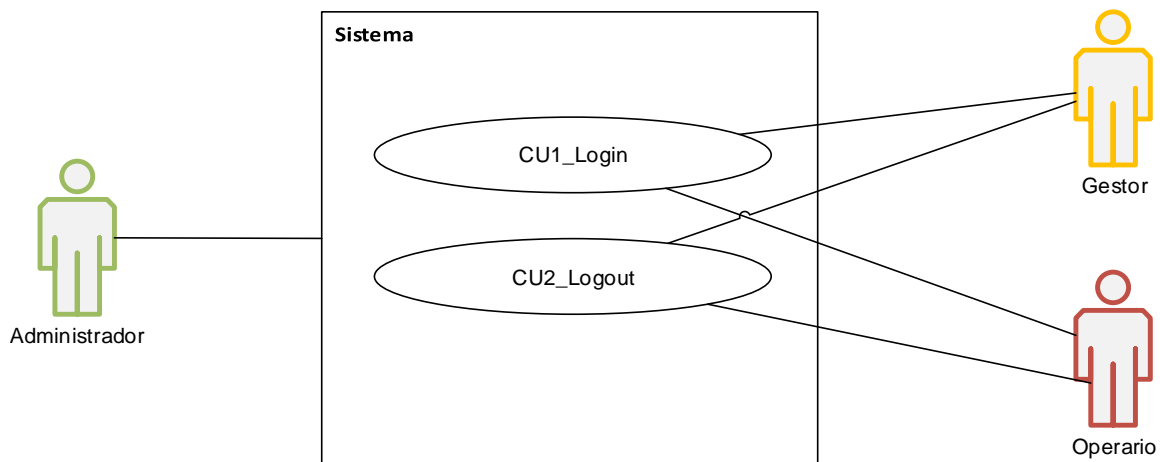


Figura 7: Casos de uso referentes al sistema

CU1_Login

Nombre: Login

Actores: Administrador, gestor y operario.

Resumen de la funcionalidad: permite al usuario identificarse ante el sistema

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla "Login".
- 2) El usuario introduce los datos de identificación.
- 3) El sistema comprueba que los datos de identificación son correctos y registra la identificación del usuario en su sesión.

Flujo de acontecimientos alternativos:

- 2ª) El usuario cierra la ventana.
- 2ª.1) El caso de uso finaliza.
- 3ª) Los datos proporcionados por el usuario no son correctos.
- 3ª.1) El sistema muestra la pantalla “Login” con un mensaje de error.
- 3ª.2) Ir al punto 2.

Pantallas:

- Login
 - Datos solicitados:
 - Email
 - Contraseña

CU2_Logout

Nombre: Logout

Actores: Administrador, gestor y operario.

Resumen de la funcionalidad: permite al usuario salir del sistema

Flujo de acontecimientos principal:

- 1) El sistema muestra la opción “Logout”.
- 2) El usuario selecciona la opción “Logout”.
- 3) El usuario sale del sistema y muestra la pantalla “Login”.

Pantallas:

- Login
 - Datos solicitados:
 - Email
 - Contraseña

Este subsistema representa todos los casos de uso referentes a las tareas **Generales**:

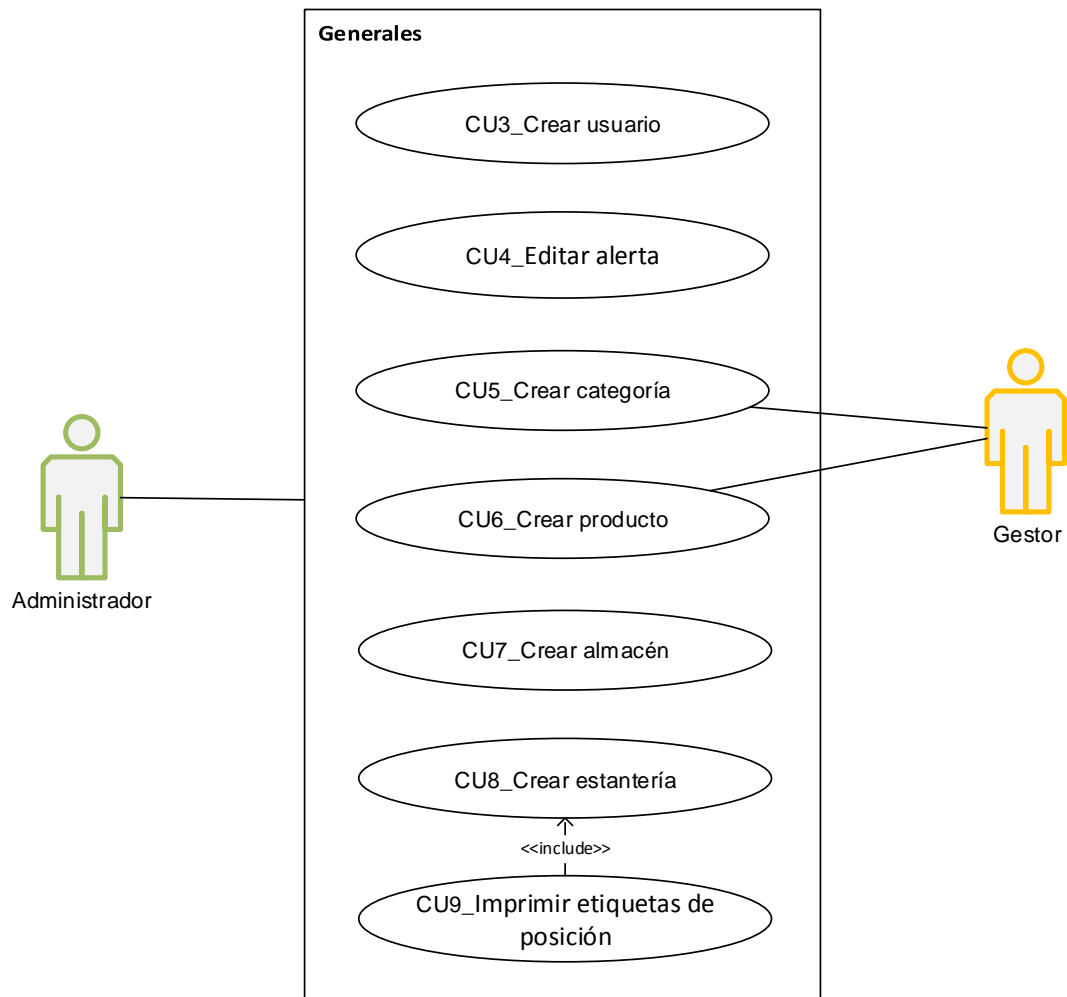


Figura 8: Casos de uso generales

CU3_Crear usuario

Nombre: Crear nuevo usuario

Actores: Administrador.

Resumen de la funcionalidad: permite al administrador crear un usuario

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Nuevo usuario”.
- 2) El usuario introduce los datos de los campos obligatorios.
- 3) El usuario selecciona “Crear”.
- 4) El sistema comprueba los datos y crea el nuevo usuario.

Flujo de acontecimientos alternativos:

- 2ª) El usuario introduce “Cancelar”.
- 2ª.1) El caso de uso finaliza.
- 3ª) Los datos proporcionados por el usuario no son correctos.
- 3ª.1) El sistema muestra la pantalla “Alta nuevo usuario” con un mensaje de error.
- 3ª.2) Ir al punto 2.

Pantallas:

- Crear usuario
Datos solicitados:
 - Nombre
 - Apellidos
 - Tipo de usuario (administrador, gestor u operario)
 - Pass
 - Email
 - Teléfono

CU4_Editar alerta

Nombre: Editar alerta

Actores: Administrador.

Resumen de la funcionalidad: permite al usuario editar una alerta.

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Editar alerta”.
- 2) El usuario introduce los datos requeridos.
- 3) El sistema comprueba que los datos son correctos y edita la alerta.

Flujo de acontecimientos alternativos:

2ª) El usuario introduce “Cancelar”.

2ª.1) El caso de uso finaliza.

3ª) Los datos proporcionados por el usuario no son correctos.

3ª.1) El sistema muestra la pantalla “Crear alerta” con un mensaje de error.

3ª.2) Ir al punto 2.

Pantallas:

- Editar alerta

Datos solicitados:

- Emails

CU5_Crear categoría

Nombre: Crear categoría

Actores: Administrador y gestor.

Resumen de la funcionalidad: permite al usuario crear una categoría de productos.

Flujo de acontecimientos principal:

1) El sistema muestra la pantalla “Crear nueva categoría”.

2) El usuario introduce los datos requeridos.

3) El sistema comprueba que los datos son correctos y crea una nueva categoría.

Flujo de acontecimientos alternativos:

2ª) El usuario introduce “Cancelar”.

2ª.1) El caso de uso finaliza.

3ª) Los datos proporcionados por el usuario no son correctos.

3ª.1) El sistema muestra la pantalla “Crear nueva categoría” con un mensaje de error.

3ª.2) Ir al punto 2.

Pantallas:

- Crear categoría

Datos solicitados:

- Nombre

CU6_Crear producto

Nombre: Crear producto

Actores: Administrador y gestor.

Resumen de la funcionalidad: permite al usuario crear un producto.

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Nuevo producto”.
- 2) El usuario introduce los datos requeridos.
- 3) El sistema comprueba que los datos son correctos y crea un nuevo producto.

Flujo de acontecimientos alternativos:

- 2^a) El usuario introduce “Cancelar”.
- 2^a.1) El caso de uso finaliza.
- 3^a) Los datos proporcionados por el usuario no son correctos.
- 3^a.1) El sistema muestra la pantalla “Alta nuevo producto” con un mensaje de error.
- 3^a.2) Ir al punto 2.

Pantallas:

- Crear producto
 - Datos solicitados:
 - Seleccionar categoría
 - Código
 - Descripción
 - Punto de pedido

CU7_Crear almacén

Nombre: Crear almacén.

Actores: Administrador.

Resumen de la funcionalidad: permite al usuario crear un almacén.

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Nuevo almacén”.
- 2) El usuario introduce los datos requeridos.
- 3) El sistema comprueba que los datos son correctos y crea un nuevo almacén.

Flujo de acontecimientos alternativos:

- 2ª) El usuario introduce “Cancelar”.
- 2ª.1) El caso de uso finaliza.
- 3ª) Los datos proporcionados por el usuario no son correctos.
- 3ª.1) El sistema muestra la pantalla “Crear almacén” con un mensaje de error.
- 3ª.2) Ir al punto 2.

Pantallas:

- Crear almacén
Datos solicitados:
 - Nombre

CU8_Crear estantería

Nombre: Crear estantería.

Actores: Administrador.

Resumen de la funcionalidad: permite al usuario crear una estantería.

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Nueva estantería”.
- 2) El usuario introduce los datos requeridos.
- 3) El sistema comprueba que los datos son correctos y crea una nueva estantería.

Flujo de acontecimientos alternativos:

- 2ª) El usuario introduce “Cancelar”.
- 2ª.1) El caso de uso finaliza.
- 3ª) Los datos proporcionados por el usuario no son correctos.
- 3ª.1) El sistema muestra la pantalla “Crear estantería” con un mensaje de error.
- 3ª.2) Ir al punto 2.

Pantallas:

- Crear estantería
 Datos solicitados:
 - Seleccionar almacén
 - Descripción
 - Código
 - Número de filas
 - Número de columnas

CU9_Imprimir etiquetas de posición

Nombre: Imprimir etiquetas de posición

Actores: Administrador.

Caso de uso relacionados: Crear estantería.

Resumen de la funcionalidad: permite al usuario imprimir etiquetas para las posiciones de cada estantería.

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Imprimir posiciones”.
- 2) El usuario selecciona “Imprimir posiciones” de la estantería deseada.
- 3) El sistema comprueba los datos introducidos.
- 4) El sistema muestra la pantalla “Imprimir posición”.
- 5) El usuario selecciona imprimir.

Flujo de acontecimientos alternativos:

- 2ª) El usuario introduce “Volver”.
- 2ª.1) Ir al punto 1.

Pantallas:

- Imprimir posición
- Gestión de estanterías

Este subsistema representa todos los casos de uso referentes a las tareas asociadas al **Inventario Teórico y Movimientos Teóricos**:

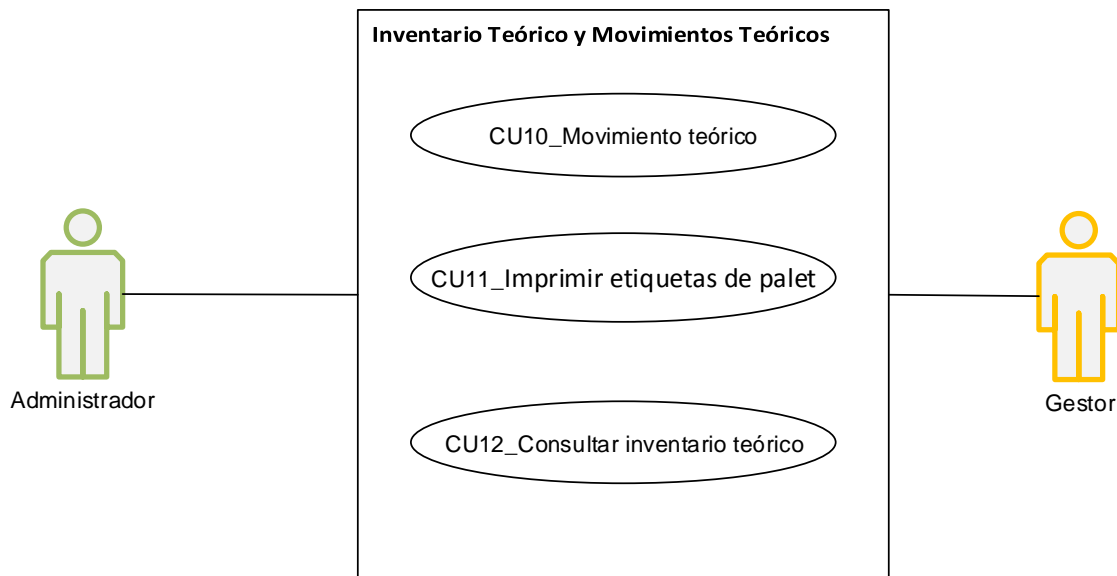


Figura 9: Casos de uso referentes a movimientos teóricos

CU10_Movimiento teórico

Nombre: Movimiento teórico.

Actores: Administrador y gestor.

Resumen de la funcionalidad: permite al usuario introducir una cantidad de producto en el inventario teórico.

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla "Movimientos teóricos".
- 2) El usuario selecciona "Nuevo movimiento teórico".
- 3) El sistema muestra la pantalla "Movimiento teórico".
- 4) El usuario selecciona "Nueva línea".
- 5) El sistema muestra la pantalla "Guardar línea movimiento teórico".
- 6) El usuario introduce los datos requeridos.
- 7) El sistema comprueba los datos y añade la nueva línea.

- 8) El sistema muestra la pantalla “Movimiento teórico”.
- 9) El sistema introduce los datos requeridos.
- 10) El sistema comprueba los datos requeridos.
- 11) El sistema muestra la pantalla “Movimientos teóricos”.

Flujo de acontecimientos alternativos:

- 3ª) El usuario introduce “Cancelar”.
- 3ª.1) El caso de uso finaliza.
- 7ª) Los datos proporcionados por el usuario no son correctos.
- 7ª.1) El sistema muestra la pantalla “Guardar línea movimiento teórico”.
- 7ª.2) Ir al punto 5.

Pantallas:

- Movimiento teórico
Datos solicitados:
 - Fecha
 - Referencia
 - Observaciones
- Guardar línea movimiento teórico
Datos solicitados:
 - Seleccionar producto
 - Tipo de movimiento (entrada o salida)
 - Cantidad

CU11_Imprimir etiquetas de palet

Nombre: Imprimir etiquetas de palet

Actores: Administrador y gestor.

Caso de uso relacionados: Movimiento teórico.

Resumen de la funcionalidad: permite al usuario imprimir etiquetas para los palets y su posterior almacenamiento.

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Movimiento teórico”.
- 2) El usuario selecciona “Imprimir etiquetas de palet” de la línea de movimiento correspondiente.
- 3) El sistema comprueba los datos introducidos.
- 4) El sistema muestra la pantalla “Imprimir etiquetas de palet”.
- 5) El usuario introduce los datos requeridos.
- 6) El usuario selecciona imprimir.

Flujo de acontecimientos alternativos:

- 2ª) El usuario introduce “Cancelar”.
- 2ª.1) El caso de uso finaliza.
- 3ª) Los datos proporcionados por el usuario no son correctos.
- 5ª.1) El sistema muestra la pantalla “Imprimir etiquetas de palet” con un mensaje de error.
- 5ª.2) Ir al punto 4.

Pantallas:

- Movimiento teórico
 - Imprimir etiquetas de palet
- Datos solicitados:
- Cantidad por palet

CU12_Consultar inventario teórico

Nombre: Consultar inventario teórico.

Actores: Administrador y gestor.

Resumen de la funcionalidad: permite al usuario consultar el listado de productos con sus correspondientes stocks (stock total, stock disponible, stock en curso y stock físico)

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Inventario teórico”.
- 2) El usuario visualiza los stocks

Flujo de acontecimientos alternativos:

- 2ª) El usuario introduce "Cancelar".
- 2ª.1) El caso de uso finaliza.
- 2b) El usuario selecciona ver el resultado por tipo de categoría.
- 2b.1) El sistema muestra la pantalla con los resultados.
- 2b.2) Ir al punto 2.
- 2c) El usuario introduce el producto a buscar.
- 2c.1) El sistema muestra la pantalla con los resultados.
- 2c.2) Ir al punto 2.

Pantallas:

- Inventario teórico
 - Datos solicitados:
 - Búsqueda por código o producto
 - Seleccionar categoría

Este subsistema representa todos los casos de uso referentes a las tareas asociadas al **Inventario Físico y Movimientos Físicos**:

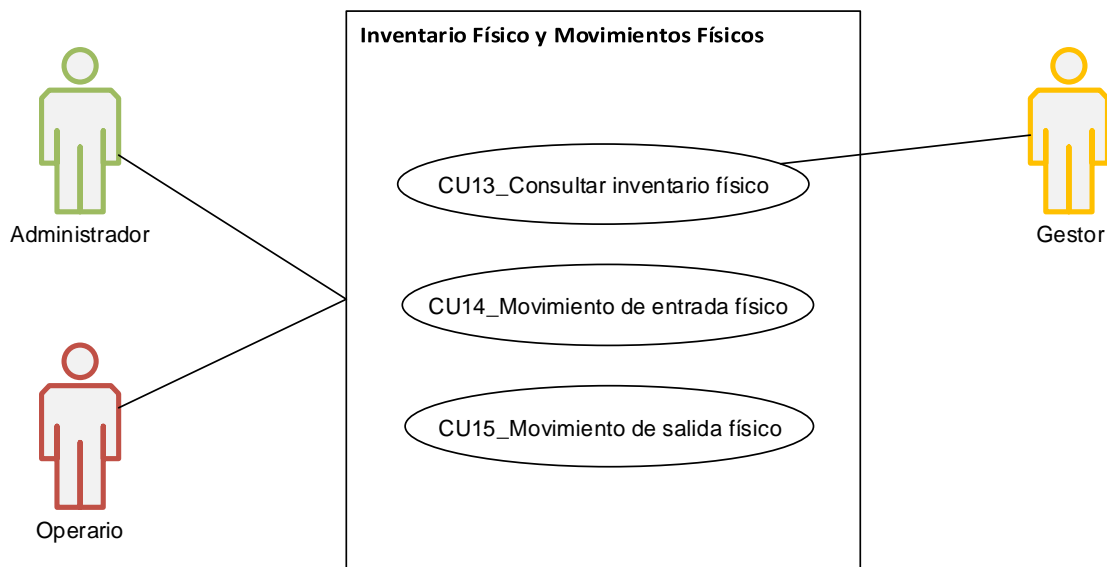


Figura 10: Casos de uso referente a movimientos físicos

CU13_Consultar inventario físico

Nombre: Consultar inventario físico.

Actores: Administrador, gestor y operario.

Resumen de la funcionalidad: permite al usuario consultar el listado de palets con su correspondiente nombre y datos del producto continente (nombre, código y cantidad), así como su posición en el almacén.

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Inventario físico”.
- 2) El usuario visualiza los datos.

Flujo de acontecimientos alternativos:

- 2^a) El usuario introduce “Cancelar”.
- 2^a.1) El caso de uso finaliza.
- 2b) El usuario introduce el producto a buscar.
- 2b.1) El sistema muestra la pantalla con los resultados.
- 2b.2) Ir al punto 2.

Pantallas:

- Inventario teórico
- Datos solicitados:
- Búsqueda por código, producto o palet.

CU14_Movimiento físico de entrada

Nombre: Movimiento físico de entrada.

Actores: Administrador y operario.

Resumen de la funcionalidad: permite al usuario posicionar en los almacenes una unidad de producto.

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Movimiento físico – ENTRADA”.
- 2) El usuario introduce el palet.
- 3) El usuario introduce la posición.
- 4) El sistema comprueba los datos y actualiza el inventario físico.
- 5) El sistema muestra la pantalla “Movimiento físico – ENTRADA” con la última entrada.

Flujo de acontecimientos alternativos:

- 2ª) El usuario introduce “Cancelar”.
- 2ª.1) El caso de uso finaliza.
- 4ª) Los datos proporcionados por el usuario no son correctos.
- 4ª.1) El sistema muestra la pantalla “Movimiento físico – ENTRADA” con un mensaje de error.
- 4ª.2) Ir al punto 1.

Pantallas:

- Movimiento físico – ENTRADA
Datos solicitados:
 - Palet
 - Posición

CU15_Movimiento físico de salida

Nombre: Movimiento físico de salida.

Actores: Administrador y operario.

Resumen de la funcionalidad: permite al usuario extraer una cantidad de producto de cada palet.

Flujo de acontecimientos principal:

- 1) El sistema muestra la pantalla “Movimiento físico – SALIDA”.
- 2) El usuario introduce el palet.
- 3) El usuario introduce la cantidad.

- 4) El sistema comprueba los datos y actualiza el inventario físico.
- 5) El sistema muestra la pantalla “Movimiento físico – SALIDA” con la última salida.

Flujo de acontecimientos alternativos:

- 2ª) El usuario introduce “Cancelar”.
- 2ª.1) El caso de uso finaliza.
- 4ª) Los datos proporcionados por el usuario no son correctos.
- 4ª.1) El sistema muestra la pantalla “Introducir cantidad producto físico” con un mensaje de error.
- 4ª.2) Ir al punto 1.

Pantallas:

- Movimiento físico – SALIDA
 Datos solicitados:
 - Palet
 - Cantidad

12.2 Diagrama de clases UML

A continuación se expone el diagrama de clases de la aplicación web donde se puede observar todos los atributos y las principales funciones de cada una de éstas. Por cuestiones de legibilidad se han omitido los *getters* y los *setters*. Dichas clases se ubican en la carpeta modelos del proyecto.

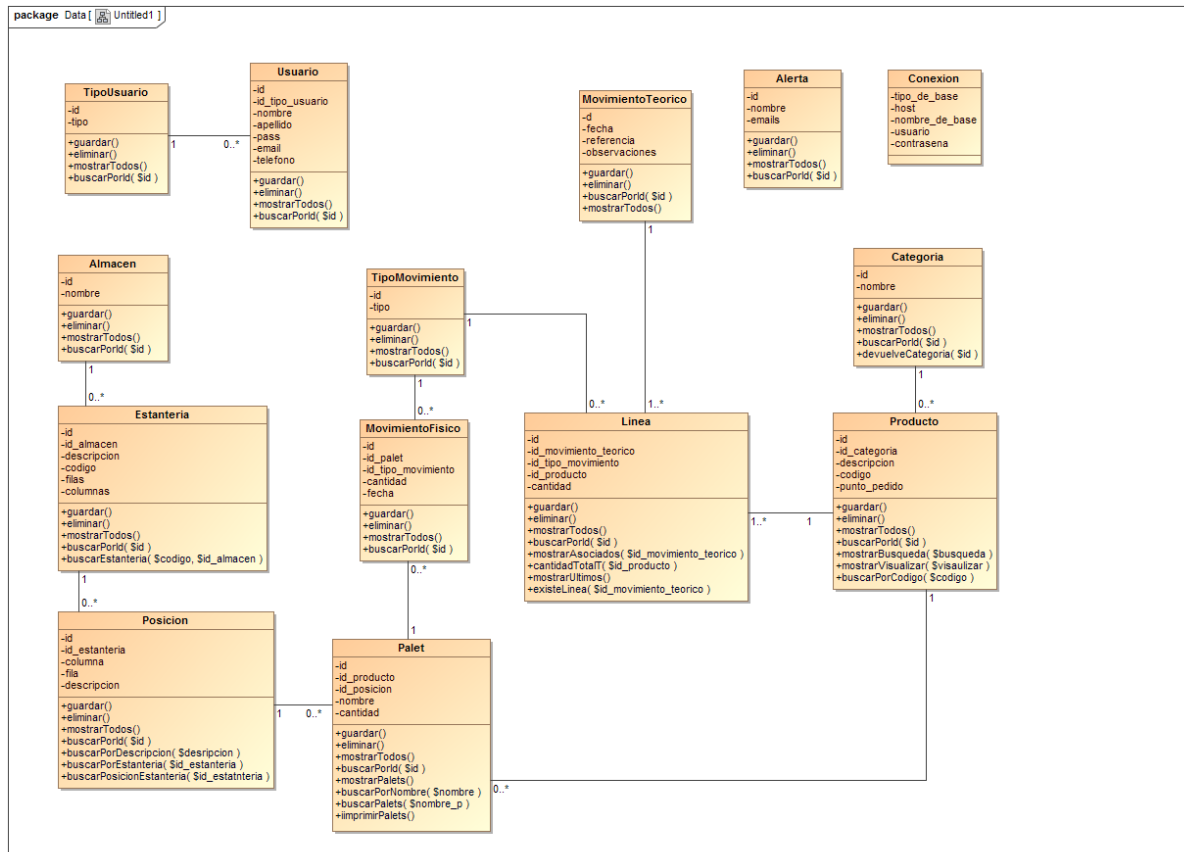


Tabla 3: Diagrama de clases UML

12.3 Diagrama base de datos

A continuación se expone el diagrama entidad relación de la base de datos Mysql 5.5.24 con motor de almacenamiento InnoDB para todas sus tablas y cotejamiento utf8_spanish_ci. Para cada tabla se muestra sus atributos, la clave primaria, las claves foráneas, el tipo de atributo y su longitud. La base de datos se ubica en la carpeta bbdd del proyecto.

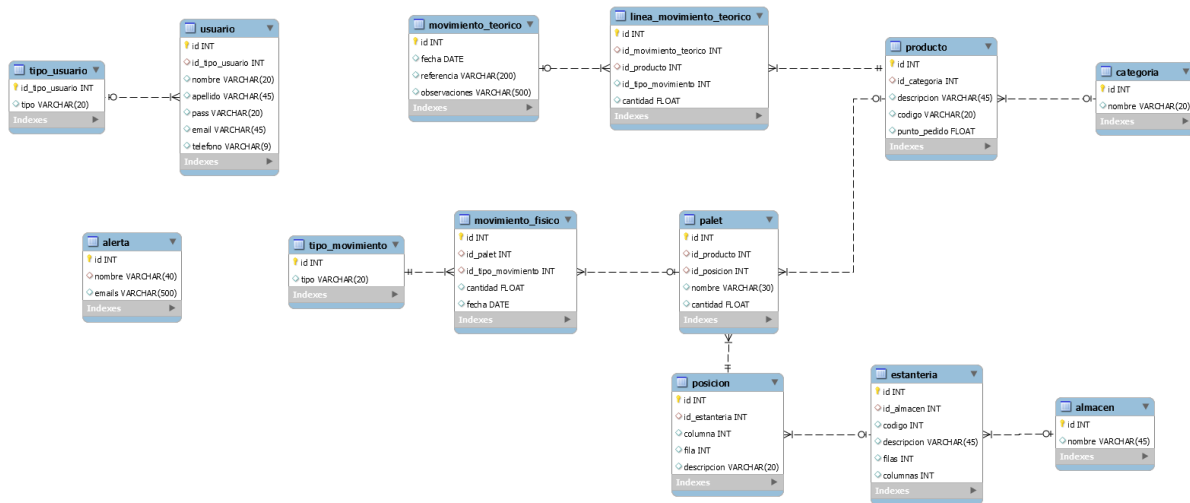


Tabla 4: Diagrama base de datos

13. Prototipos

13.1 Lo-Fi

Pantalla (escritorio-tablet) Login

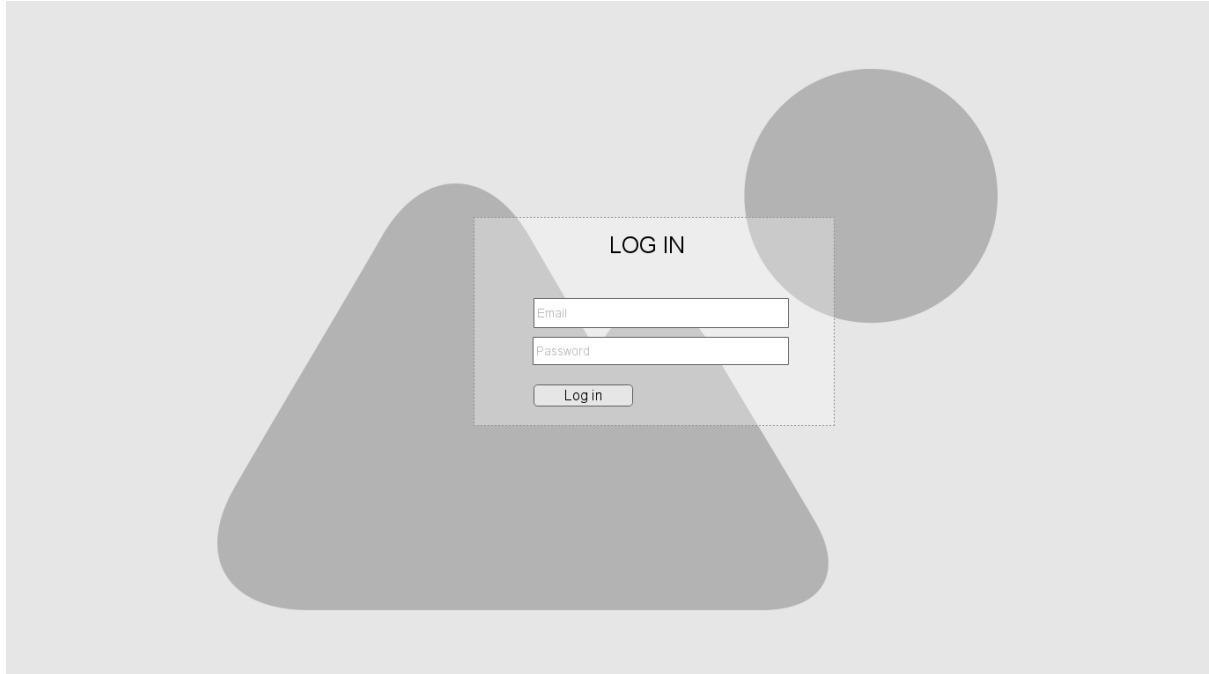


Figura 11: Prototipo Lo-Fi Login

Pantalla (escritorio-tablet) Menú principal administrador

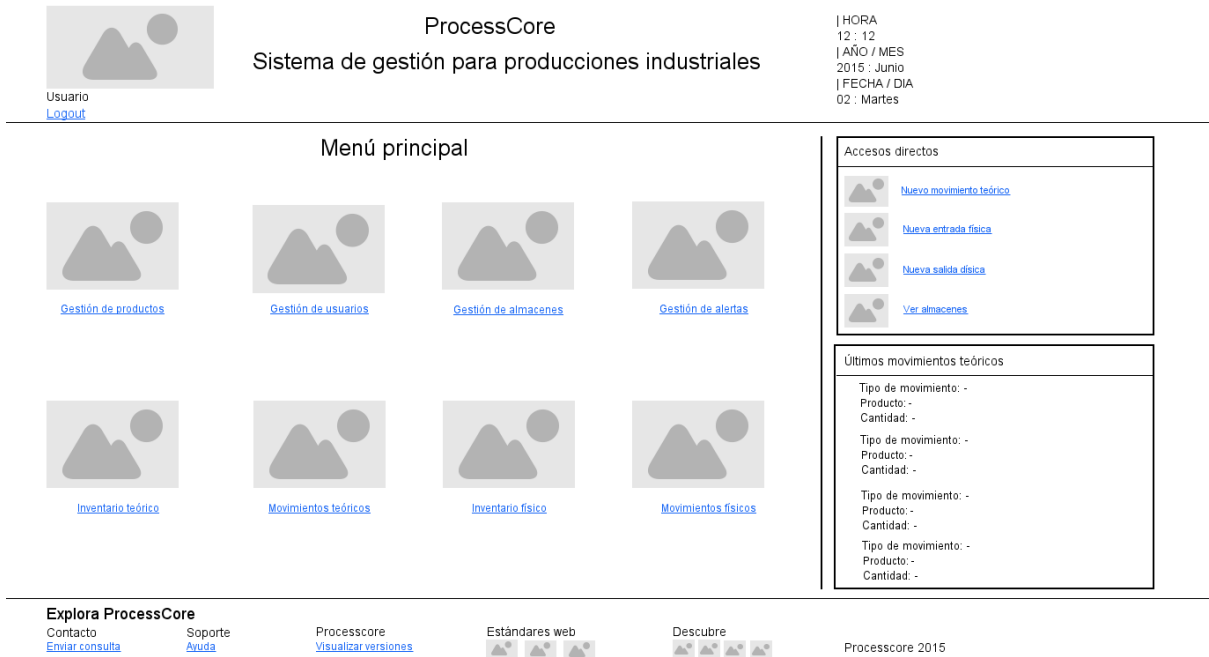


Figura 12: Prototipo Lo-Fi Menú principal

Pantalla (escritorio-tablet) Gestión de usuarios

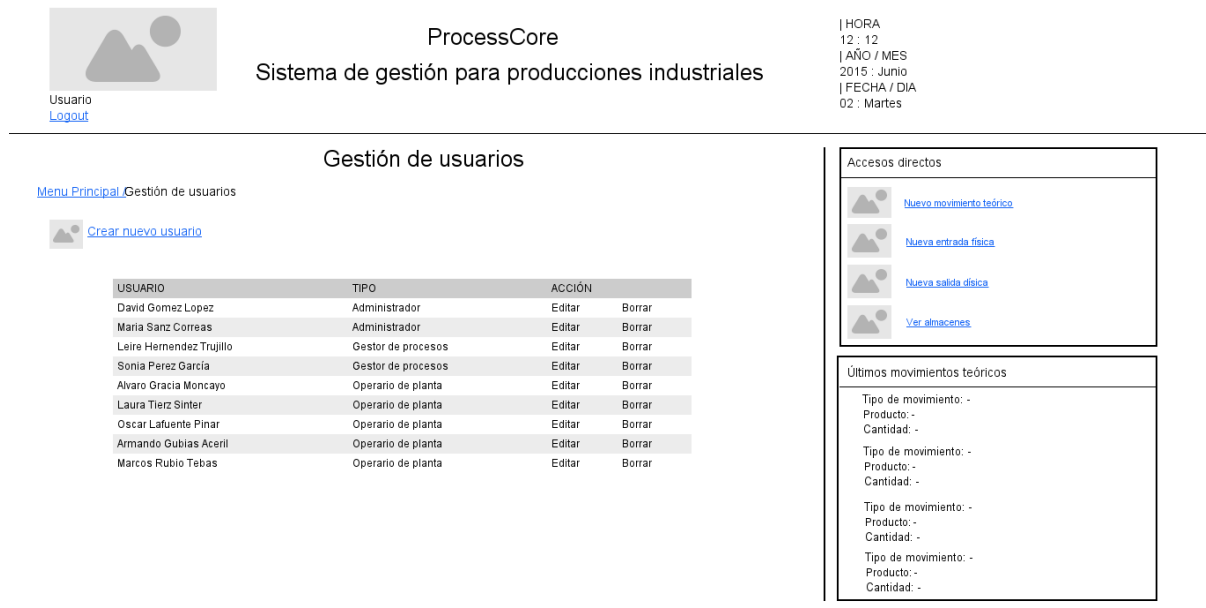


Figura 13: Prototipo Lo-Fi Menú Gestión de usuarios

Pantalla (escritorio-tablet) Alta nuevo usuario

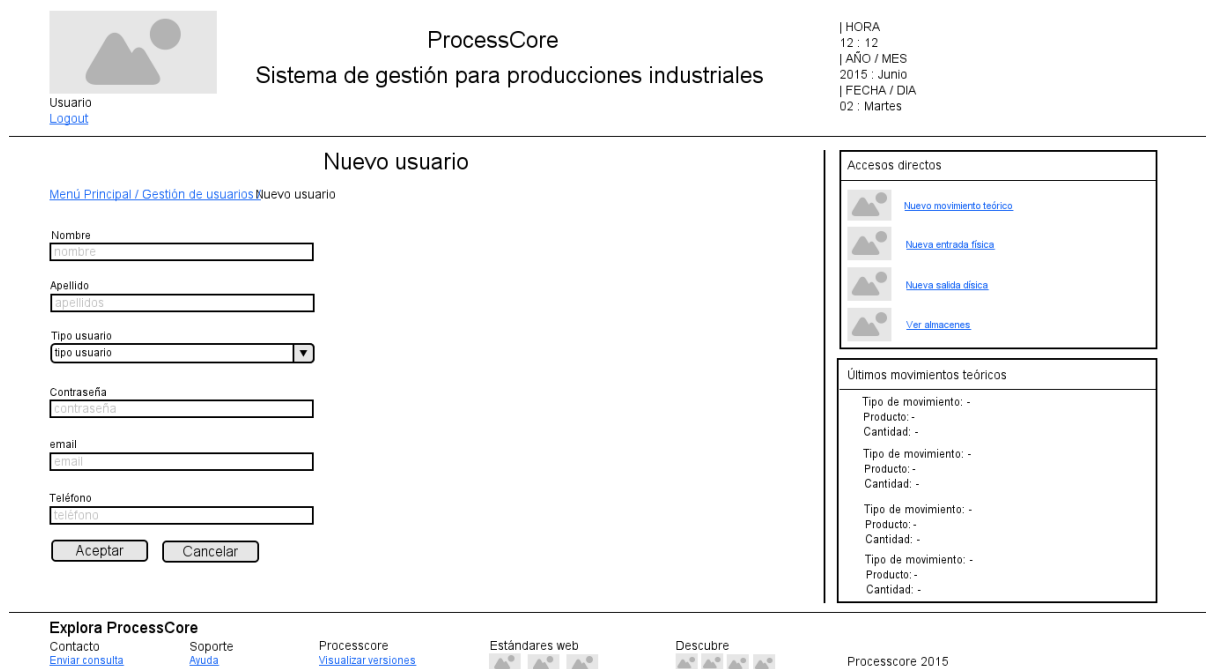


Figura 14: Prototipo Lo-Fi Nuevo usuario

Pantalla (escritorio-tablet) Gestión de alertas



Figura 15: Prototipo Lo-Fi Gestión alertas

Pantalla (escritorio-tablet) Editar alerta

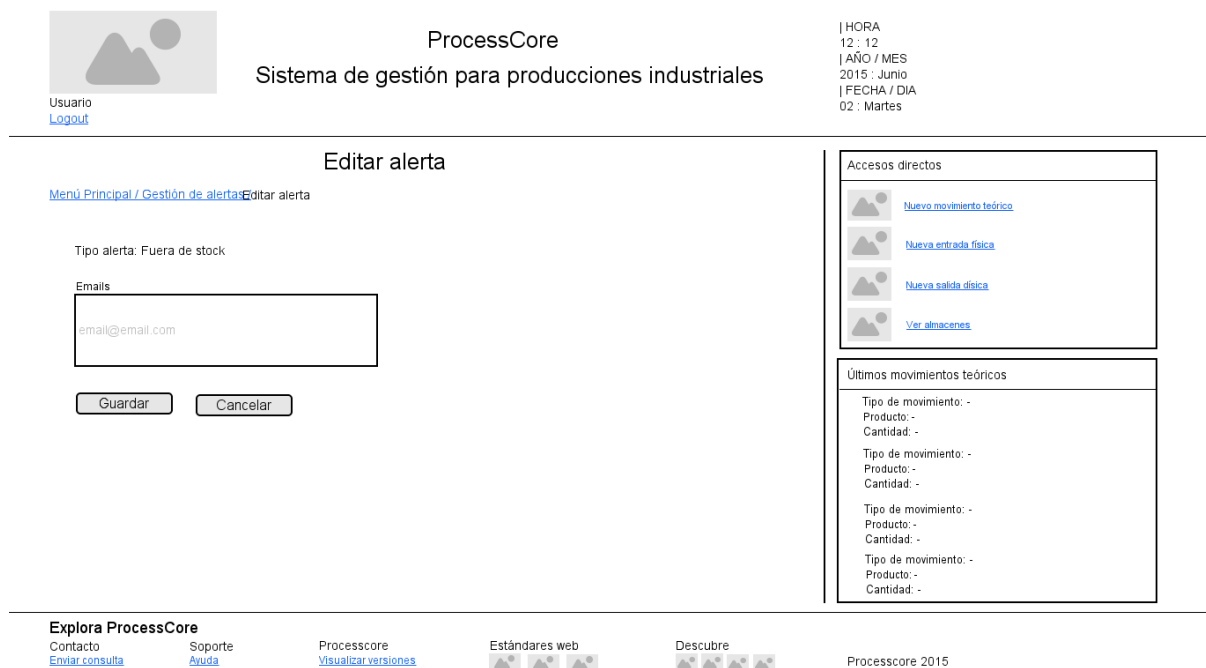


Figura 16: Prototipo Lo-Fi Editar alerta

Pantalla (escritorio-tablet) Gestión de productos



Figura 17: Prototipo Lo-Fi Gestión de productos

Pantalla (escritorio-tablet) Alta nuevo producto

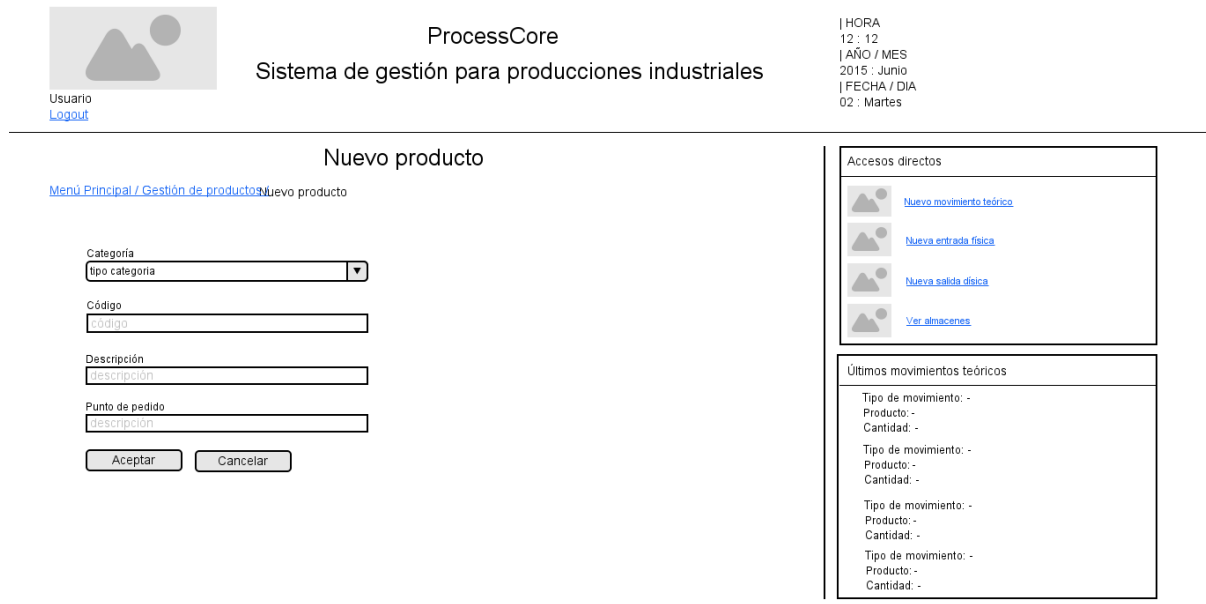


Figura 18: Prototipo Lo-Fi Nuevo producto

Pantalla (escritorio-tablet) Gestión Categorías

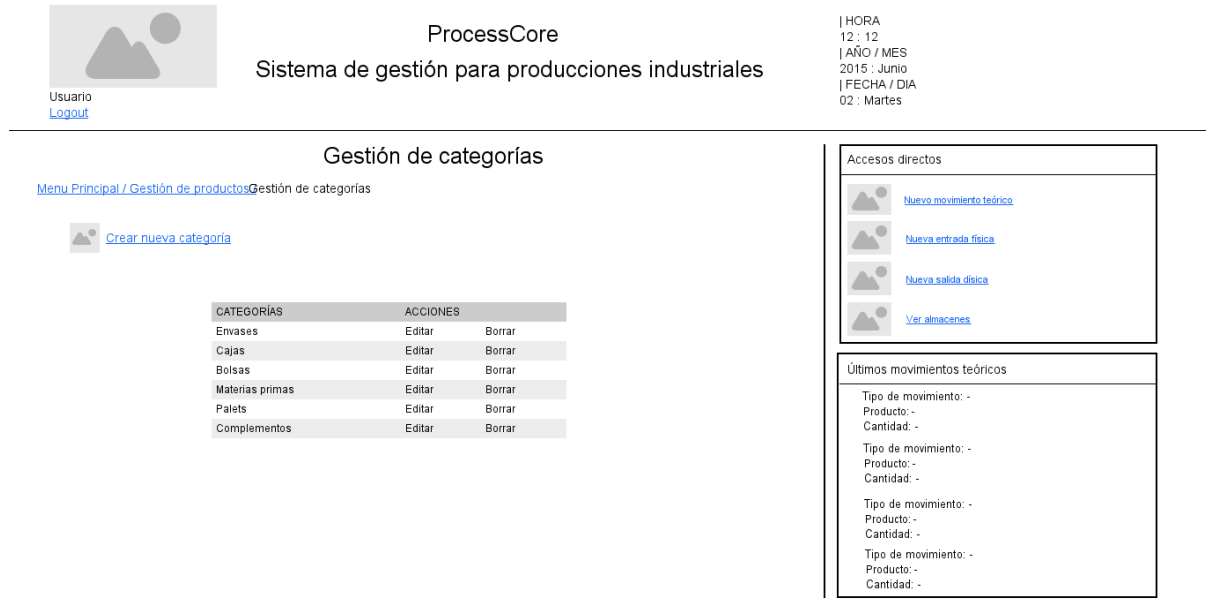


Figura 19: Prototipo Lo-Fi Gestión categorías

Pantalla (escritorio-tablet) Inventario teórico

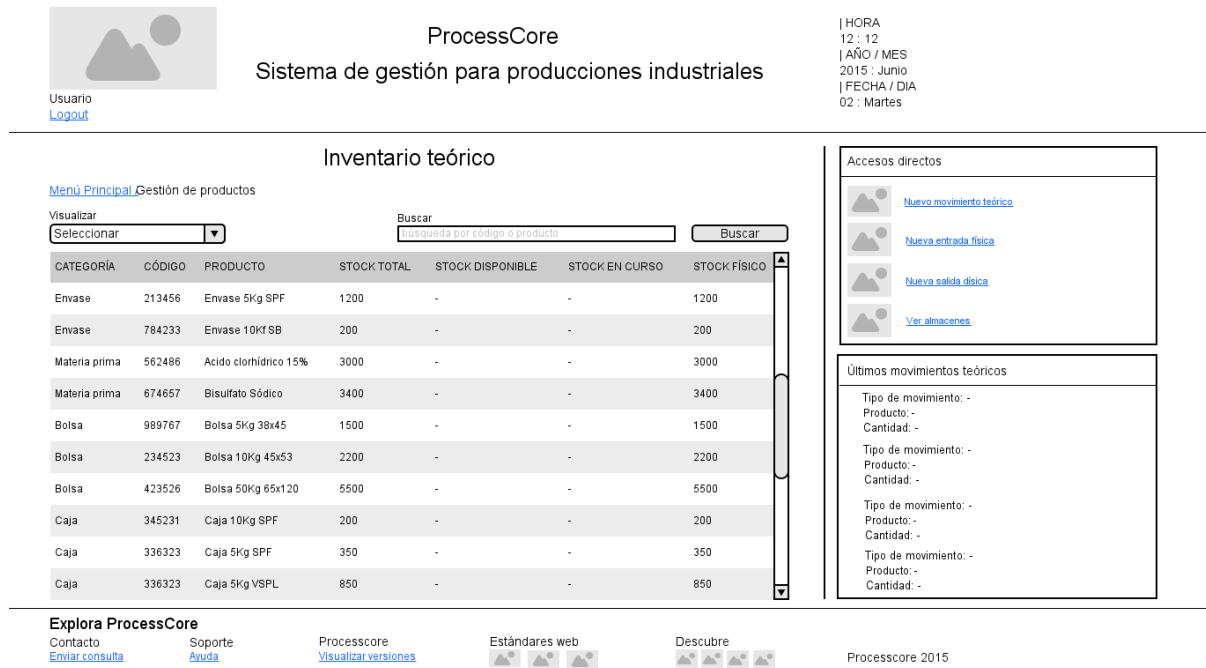



Figura 20: Prototipo Lo-Fi Inventario teórico

Pantalla (escritorio-tablet) Inventario físico



ProcessCore
Sistema de gestión para producciones industriales

| HORA
12 : 12
| AÑO / MES
2015 : Junio
| FECHA / DIA
02 : Martes

Inventario físico

[Menú Principal](#) / [Inventario físico](#)

Buscar

CÓDIGO	PRODUCTO	POSICIÓN ALMACÉN	PALET	CANTIDAD
213456	Envase 5Kg SPF	AL1E1C1F1	P115	1000
213456	Envase 5Kg SPF	AL1E1C1F2	P116	1000
213456	Envase 5Kg SPF	AL1E1C1F2	P117	650
562486	Acido clorhídrico 15%	AL1E1C2F3	P215	1250
562486	Acido clorhídrico 15%	AL1E1C23	P216	1250
562486	Acido clorhídrico 15%	AL1E1C2F4	P217	1250
674657	Bisulfato Sódico	AL1E1C3F1	P665	1000
674657	Bisulfato Sódico	AL1E1C3F1	P666	1000
989767	Bolsa 5Kg 38x45	AL1E2C1F1	P365	25000
234523	Bolsa 10Kg 45x53	AL1E2C1F1	P322	15000
423526	Bolsa 50Kg 65x120	AL1E2C1F2	P289	1000

Accesos directos

- [Nuevo movimiento teórico](#)
- [Nueva entrada física](#)
- [Nueva salida física](#)
- [Ver almacenes](#)

Últimos movimientos teóricos

Tipo de movimiento: -
Producto: -
Cantidad: -

Tipo de movimiento: -
Producto: -
Cantidad: -

Tipo de movimiento: -
Producto: -
Cantidad: -

Tipo de movimiento: -
Producto: -
Cantidad: -

Explora ProcessCore

Contacto [Enviar consulta](#) | Soporte [Ayuda](#) | Processcore [Visualizar versiones](#) | Estándares web | Descubre

Processcore 2015

Figura 21: Prototipo Lo-Fi Inventario físico

Pantalla (escritorio-tablet) Gestión de almacenes



ProcessCore
Sistema de gestión para producciones industriales

| HORA
12 : 12
| AÑO / MES
2015 : Junio
| FECHA / DIA
02 : Martes

Gestión de almacenes

[Menu Principal](#) / [Gestión de almacenes](#)

[Crear nuevo almacén](#) [Gestión estanterías](#)

ALMACENES	ACCIONES	
Almacén1	Editar	Borrar
Almacén2	Editar	Borrar
Almacén3	Editar	Borrar

Accesos directos

- [Nuevo movimiento teórico](#)
- [Nueva entrada física](#)
- [Nueva salida física](#)
- [Ver almacenes](#)

Últimos movimientos teóricos

Tipo de movimiento: -
Producto: -
Cantidad: -

Tipo de movimiento: -
Producto: -
Cantidad: -

Tipo de movimiento: -
Producto: -
Cantidad: -

Tipo de movimiento: -
Producto: -
Cantidad: -

Explora ProcessCore

Contacto [Enviar consulta](#) | Soporte [Ayuda](#) | Processcore [Visualizar versiones](#) | Estándares web | Descubre

Processcore 2015

Figura 22: Prototipo Lo-Fi Gestión almacenes

Pantalla (escritorio-tablet) Alta nuevo almacén

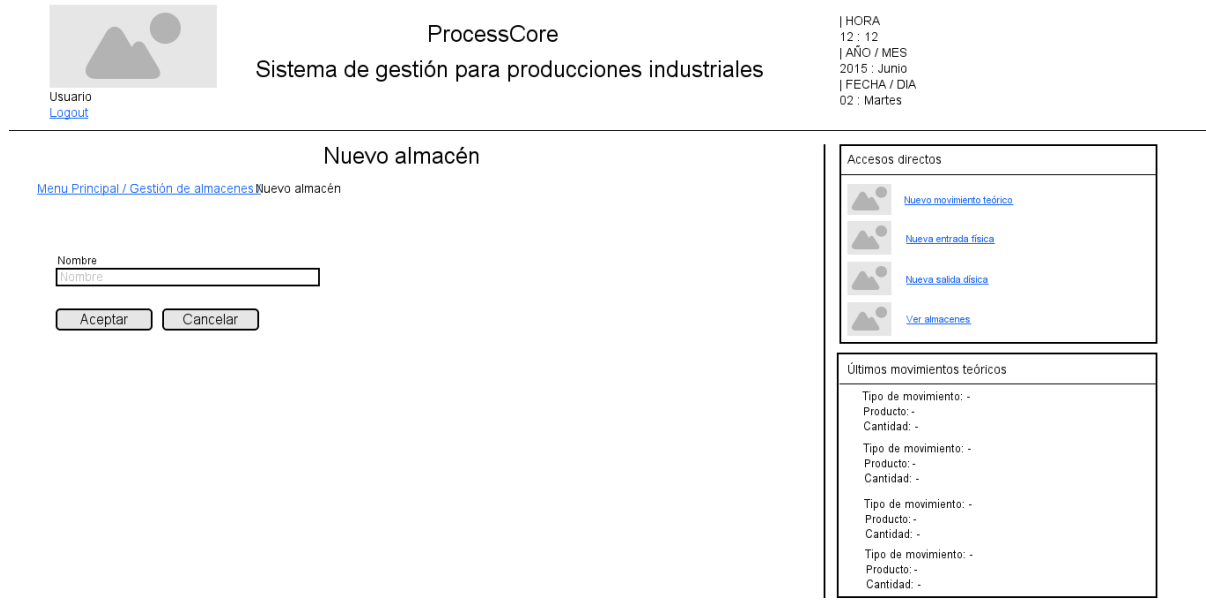


Figura 23: Prototipo Lo-Fi Nuevo almacén

Pantalla (escritorio-tablet) Gestión estanterías

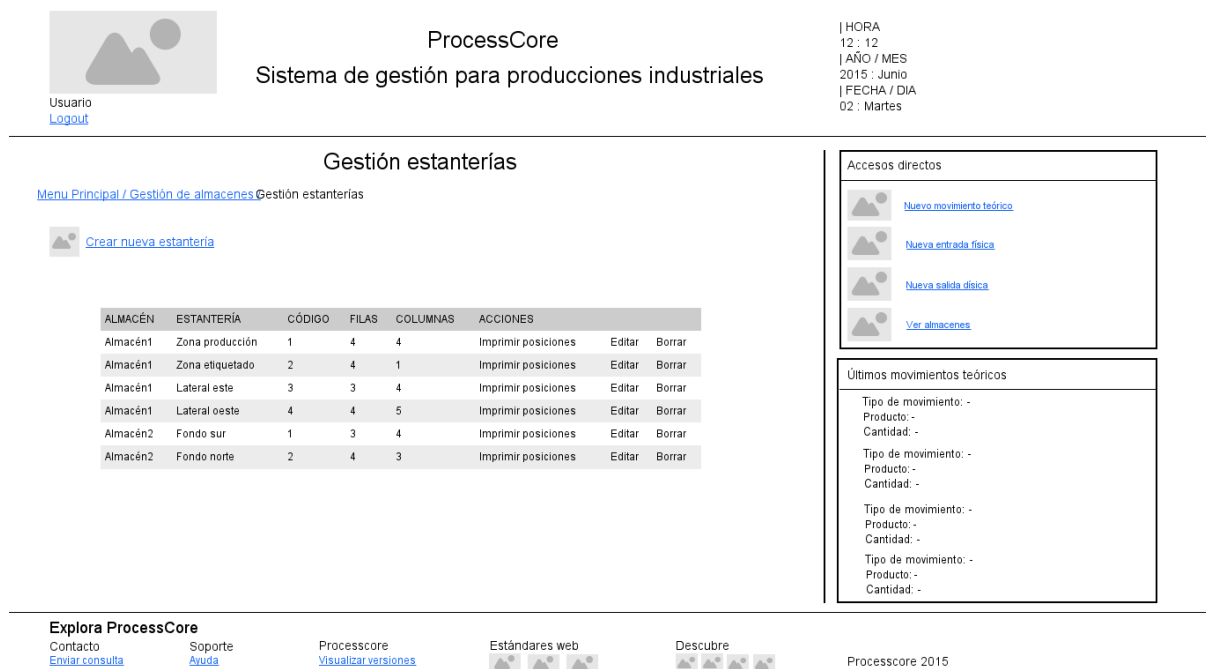


Figura 24: Prototipo Lo-Fi Gestión estanterías

Pantalla (escritorio-tablet) Movimientos teóricos

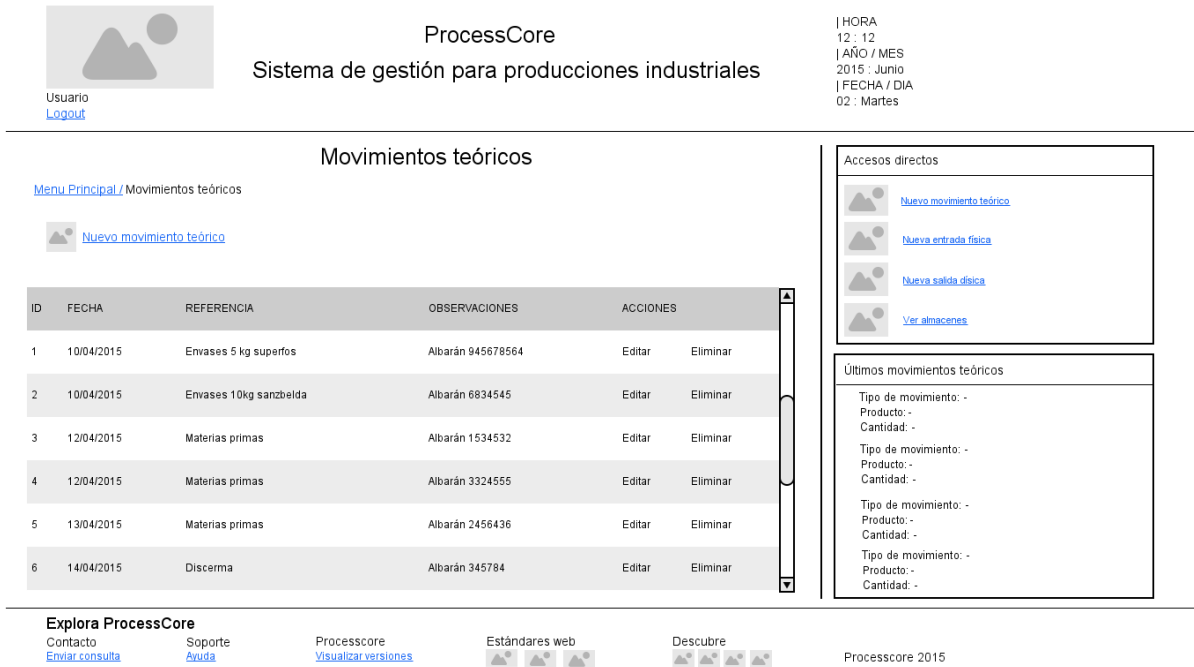


Figura 25: Prototipo Lo-Fi Movimiento teórico

Pantalla (escritorio-tablet) Nuevo movimiento teórico

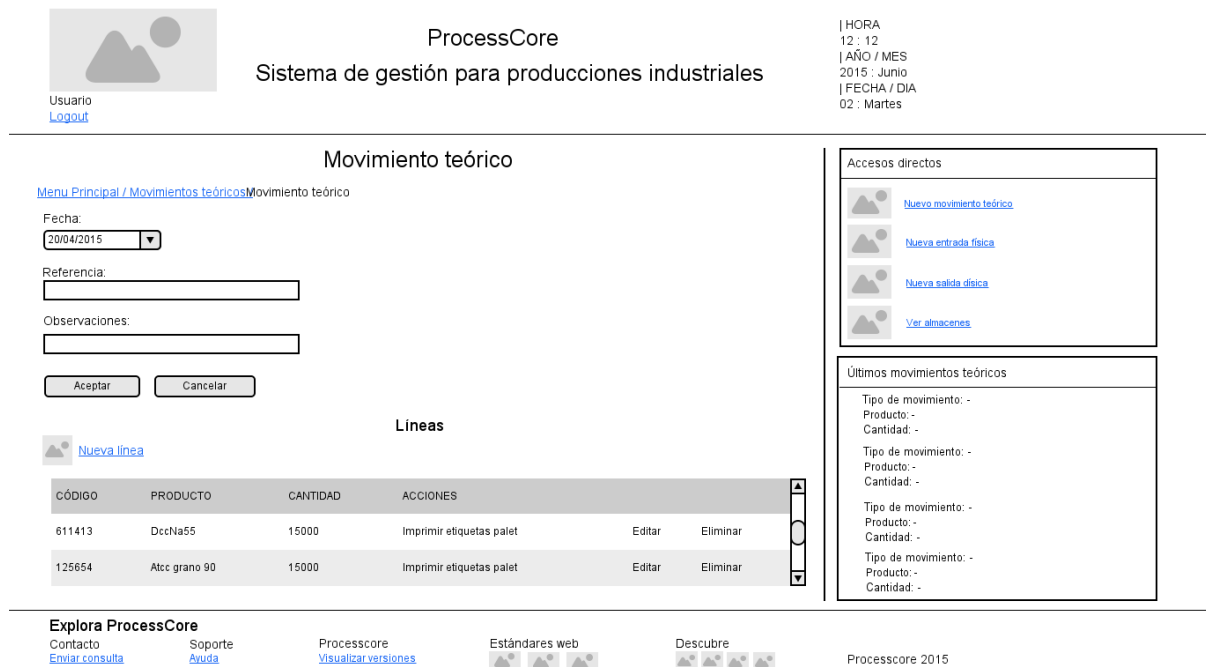


Figura 26: Prototipo Lo-Fi Nuevo movimiento teórico

Pantalla (escritorio-tablet) Movimiento físico (introducir)

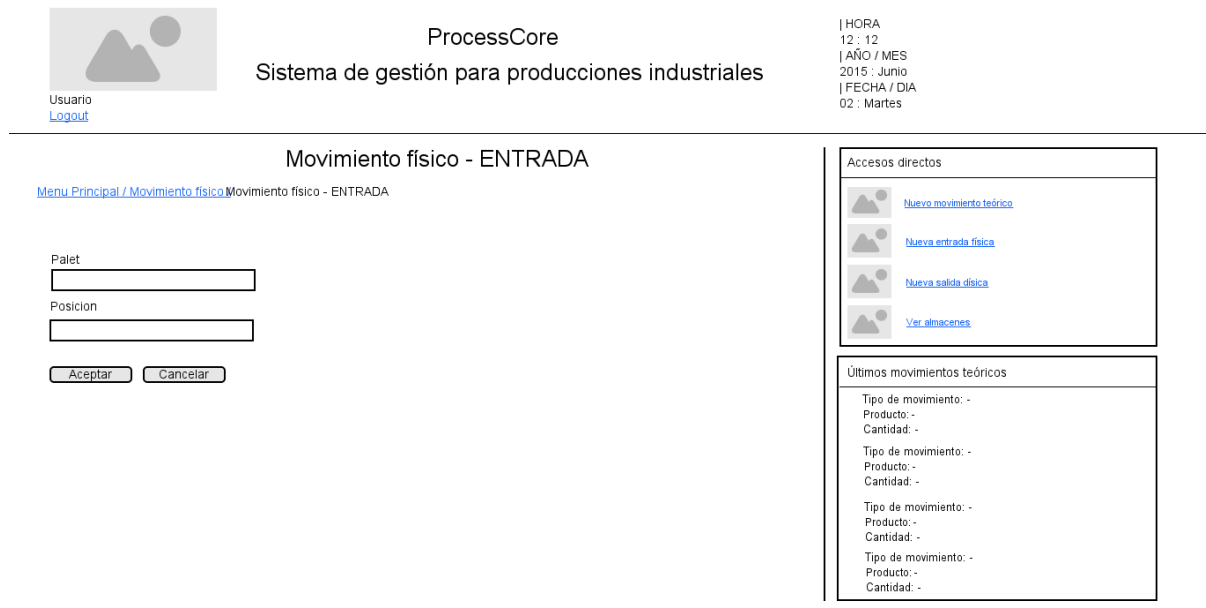


Figura 27: Prototipo Lo-Fi Movimiento físico - ENTRADA

Pantalla (escritorio-tablet) Movimiento físico (extraer)

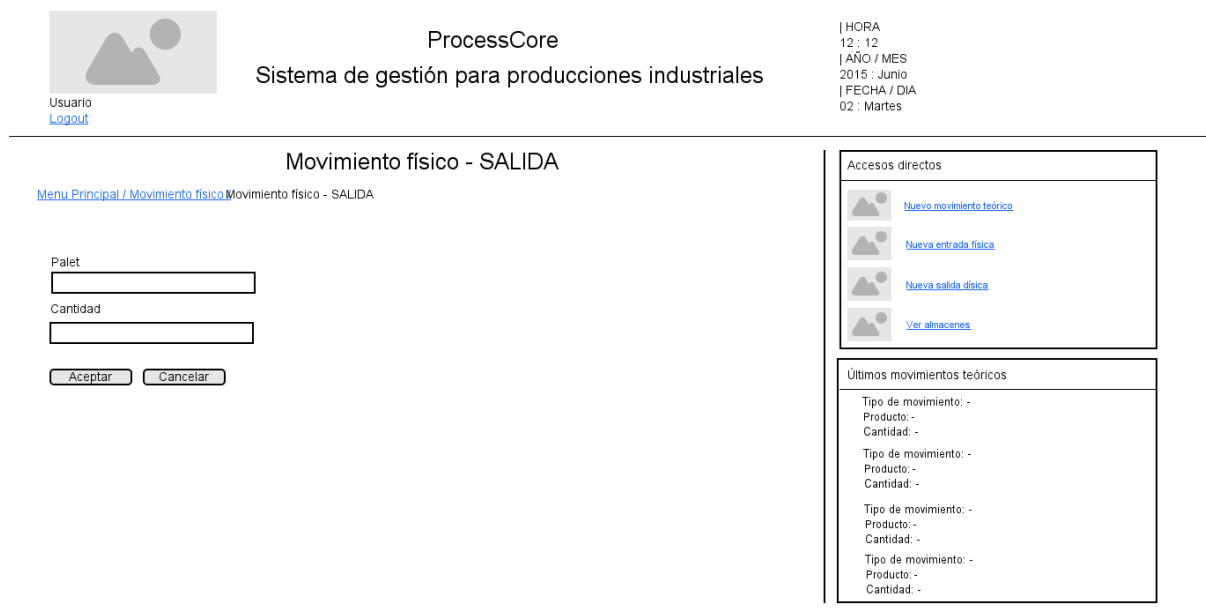


Figura 28: Prototipo Lo-Fi Movimiento físico - SALIDA

Pantalla (Smartphone-> usuario: operario) Login

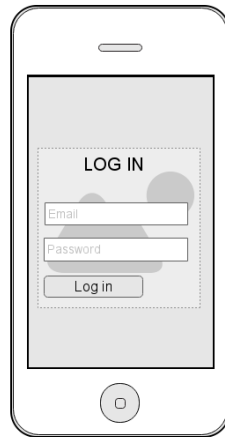


Figura 29: Prototipo Lo-Fi Login (Smartphone)

Pantalla (Smartphone-> usuario: operario) Menú principal



Figura 30: Prototipo Lo-Fi Menú principal (Smartphone)

Pantalla (Smartphone-> usuario: operario) Movimientos Físico – ENTRADA

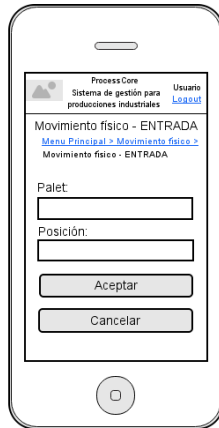


Figura 31: Prototipo Lo-Fi Movimiento físico - ENTRADA (Smartphone)

Pantalla (Smartphone-> usuario: operario) Movimientos Físico – SALIDA



Figura 32: Prototipo Lo-Fi Movimiento físico - SALIDA (Smartphone)

Pantalla (Smartphone-> usuario: operario) Inventario físico



Figura 33: Prototipo Lo-Fi Inventario físico (Smartphone)

14. Perfiles de usuario



Usuario 1 (administradores) – control total : crear usuarios, usuarios, crear productos y categorías de productos, crear almacenes, crear estanterías, crear posiciones para estanterías, introducir, extraer y redefinir cantidad de materiales del inventario teórico y físico, recepción de avisos vía email o sms, visualizar stocks, movimientos de un producto, imprimir etiquetas para el almacenamiento de productos.



Usuario 2 (gestores de procesos productivos) – crear productos y categorías de productos, introducir, extraer y redefinir cantidad de productos del inventario teórico, recepción de avisos vía email o sms, visualizar stocks, movimientos de un producto, imprimir etiquetas para almacenamiento de productos.



Usuario 3 (operarios de planta) – Consultar inventario físico, introducir y extraer materiales del inventario físico.

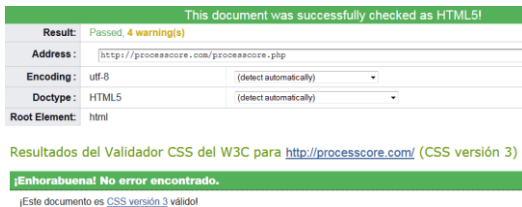
15. Usabilidad/UX

La aplicación se basa en los principios fundamentales para el diseño de interfaces de entorno gráfico, y de esta manera no obliga al usuario a aprender situaciones complejas, sino que debe poder navegar de una manera intuitiva, rápida y lo más efectiva posible. A continuación se exponen los diez principios generales de usabilidad en los que se rige:

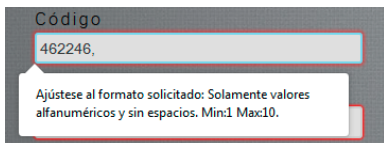
1. **Visibilidad del estado del sistema.** En todo momento el sistema informa al usuario de lo que está ocurriendo y dar respuesta en un tiempo razonable, por ejemplo cuando el login es incorrecto.



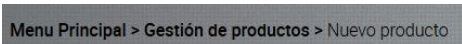
2. **Consistencia entre el sistema y el mundo real.** La aplicación utiliza el lenguaje del usuario, con expresiones conocidas, así como un orden lógico en la información propuesta.
3. **Control del usuario.** El usuario tendrá la capacidad de abandonar en cualquier momento una situación indeseada o accidental.
4. **Consistencia y estándares.** El lenguaje utilizado será coherente con la convenciones del sistema operativo y seguirá los estándares W3C.



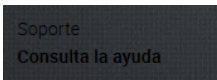
5. **Prevención de errores.** Evitar la existencia de errores. Si en algún caso se producen, se tendrá la opción de confirmación antes de ejecutar la corrección. Por ejemplo, en los formularios se indica los datos incorrectos.



6. **Es mejor reconocer que recordar.** Para no obligar al usuario a memorizar detalles de la navegación, los objetos, acciones y opciones estarán a la vista. La ayuda del sistema estará a la vista para un fácil acceso. Como ejemplo se muestra el *breadcrumb* que ayuda a una mejor experiencia de usuario.



7. **Flexibilidad y eficiencia de uso.** La aplicación está preparada para satisfacer tanto a usuarios novatos como experimentados.
8. **Diseño práctico y sencillo.** Las pantallas de la aplicación no contienen información innecesaria, y de esta manera no entorpece la navegación.
9. **Los mensajes de error** se presentan con un lenguaje sencillo, indicando el problema de manera precisa, y se sugiere las posibles soluciones.
10. **Ayuda y documentación.** Debido a que la aplicación pueda tener alguna complejidad, se dispone de documentación para su resolución. Esta ayuda será de fácil localización, exactamente en el pie de la aplicación.



11. **Uso de metáforas.** Se utilizan metáforas para una mejor comprensión.



12. **Diseño responsive.** La aplicación se adapta según la resolución de la pantalla. Para los dispositivos móviles como es el caso del operario se ocultan el contenedor secundario y el pie, además de adaptar correctamente los iconos y textos.

```
@media handheld, only screen and (max-width: 400px) {  
  div#contenedor_principal, div#contenedor_secundario {  
    background: rgba(255, 255, 255, 0.6);  
    display: inline-block;  
    min-height: 600px;  
    vertical-align: top;  
    width: 80%;  
  }  
  
  div#contenedor_secundario, footer {  
    display:none;  
  }  
  
  section#tiempo {  
    display:none;  
  }  
  
  h1 {  
    font-size: .7em;  
  }  
  section#login {  
    font-size: .8em;  
  }  
}
```

16. Seguridad

La aplicación web en principio se ha diseñado para usar en red local, es decir, menos expuesta a ataques y accesos maliciosos desde Internet. No obstante para una mayor seguridad para acceder a la aplicación se ha implantado un sistema de acceso por contraseña. En el futuro se intentará mejorar este método para una mayor seguridad, por ejemplo mediante el cifrado de la contraseña.

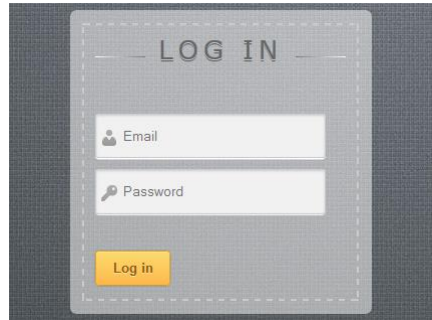


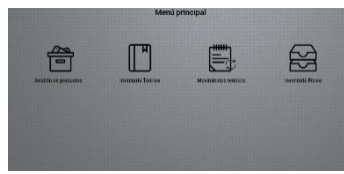
Figura 31: Prototipo Lo-Fi Movimiento físico - ENTRADA (Smartphone)

Por otro lado, según el tipo de usuario que accede al sistema se cargan unos u otros accesos. De esta manera, a parte de simplificar la aplicación para cada tipo de usuario también evita que puedan introducir o visualizar datos no concernientes a sus tareas.



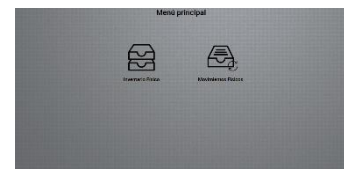
Menú administrador

Figura 34: Menú administrador



Menú gestor

Figura 35: Menú gestor



Menú operario

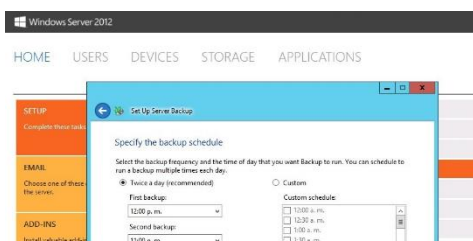
Figura 36: Menú operario

Para la introducción de datos a través de los formularios también se intenta controlar los datos para evitar posibles problemas. Con el atributo *pattern* incluido en HTML5 se delimitará mediante expresiones regulares el contenido insertado por el usuario en los inputs de los formularios. De esta manera se podrá validar diferentes valores como son *emails*, fechas, cantidades, códigos de productos...

```
pattern="[a-zA-Z0-9 ñáéíóúÁÉÍÓÚ]{1,10}$"
pattern="[a-zA-Z0-9 ñáéíóúÁÉÍÓÚ()]{3,40}$"
pattern="[0-9]{1,10}$"
```

Figura 37: Expresiones regulares

Finalmente, donde se aloja el servidor local se realiza una copia de seguridad completa diariamente, tanto de la aplicación como de la base de datos. Se realiza a través de las aplicaciones que proporciona el propio Windows server 2012 y se guarda en un disco duro dedicado a estas copias de seguridad.



17. Tests

Durante el desarrollo del proyecto se han realizado varias pruebas, principalmente tests funcionales, técnicos y de usabilidad. A continuación se presentan los diferentes test que se han ejecutado:

Tests técnicos:

- Se comprueba el correcto funcionamiento en los principales navegadores (Firefox, Internet Explorer, Safari, Google Chrome y Opera.)
- Se comprueba la correcta adaptación de la aplicación web para dispositivos con diferentes resoluciones de pantalla, intentado obtener un buen diseño *responsive*.
- Tests W3C para HTML5 y CSS3

Tests funcionales:

- Se prueba que todas las funciones realizan las tareas correctamente. Se lleva a cabo diferentes operaciones y comprobaciones para evitar al máximo cualquier tipo de error.

Tests de usabilidad:

- Se proponen diferentes tareas a los usuarios que logran concluir con diferentes resultados:
 - Crear nuevo producto
 - Crear nuevo usuario
 - Crear nuevo almacén con estanterías
 - Imprimir posiciones de una estantería
 - Editar alerta
 - Consultar el inventario teórico y físico
 - Entradas y salidas de movimientos teóricos
 - Imprimir etiquetas de palet ya impresas
 - Entradas y salidas de productos de almacén
- De las diferentes tareas anteriormente propuestas se concluyen con éxito en su mayoría, excepto en las siguientes:
 - Editar alerta -> Debido a las dudas propuestas por los usuarios, se opta por introducir en la página de alertas un texto que explica su función y como llevar a cabo la edición.
 - Imprimir etiquetas de palet ya impresas -> Algunos usuarios no son capaces de llevar a cabo esta acción. Queda pendiente para futuras mejoras.

Debido a estos errores, mientras se mejora la aplicación se ha optado por incluir un acceso de ayuda para orientar al usuario en las diferentes funcionalidades. También se crea un formulario de consulta para plantear cualquier duda a los técnicos. Estas opciones se ubican en el pie de la aplicación web.

18. Versiones de la aplicación/servicio

A continuación se muestran las versiones junto con las implantaciones que lleva cada una. En este proyecto se ha concluido la versión 1.0, en la que se realizarán mejoras y corrección de errores (versiones 1.1, 1.2 ...)

- Versión 1.0
 - Gestión de productos
 - Gestión de categorías
 - Gestión de almacenes
 - Gestión de alertas
 - Inventario teórico
 - Movimientos teóricos
 - Inventario físico
 - Movimientos físicos
- Versión 2.0
 - Gestión de clientes
 - Gestión de proveedores
 - Gestión de pedidos
 - Gestión de albaranes
 - Gestión de facturas
 - Gestión de expediciones
- Versión 3.0
 - Planificación de la producción
 - Gestión de líneas de producción
 - Generación hojas de trabajo (*kanbans*)
 - Control de producción
 - Gráficos de resultados

19. Requisitos

Software

La aplicación web se ha desarrollado con las últimas tecnologías web como son HTML5 y CSS3, así como el uso de diferentes lenguajes de programación (Javascript, PHP y Mysql). En principio su uso está destinado a nivel de red local, pero no se descarta su uso a través de Internet, con lo cual sería necesario disponer de acceso a cualquiera de las redes mencionadas. Añadir que para la función de lector de códigos de barras (uso como operario) en principio se utilizará una Demo que requiere disponer de un dispositivo con sistema Android. Finalmente recordar que los usuarios no deben tener bloqueado las funciones de Javascript en los navegadores.

Principales software operativos compatibles:

- Microsoft Windows Vista o versión posterior
- Mac OS X Leopard 10.5.8 o Mac OS X Snow Leopard 10.6.2 o versión posterior
- Ubuntu v10.10 o versión posterior
- Android KitKat o versión posterior
- iOS 6.0 o versión posterior

Principales navegadores compatibles:

- Internet Explorer 9.0 o posterior
- Google Chrome en cualquiera de sus versiones
- Safari 5 o posterior
- Firefox 5 o posterior
- Opera 10 o posterior

Para su instalación y uso será necesario instalar un servidor local, por ejemplo Wampserver que proporcionará los cuatro elementos necesarios para un servidor web: un Sistema Operativo (Windows), un manejador de base de datos (MySQL), un software para servidor web (Apache) y un software de programación script Web (PHP). Actualmente la versión utilizada es la 2.2 que incluye PHP 5.4.3, MySQL 5.5.24 y Apache 2.2.22.

Hardware

Servidor

- Equipo informático de escritorio o servidor con los sistemas operativos y navegadores compatibles.

Usuarios: Administrador y gestor

- Equipo informático de escritorio o Tablet con los sistemas operativos y navegadores compatibles.
- Impresora para las etiquetas de palet y posiciones de estantería.

Usuarios: Operario.

- Dispositivo móvil con sistema operativo, navegadores compatibles y además la opción de instalar una aplicación que permita la lectura de códigos de barras desde el teclado.

Formación/Conocimientos

Disponer conocimientos informáticos a nivel de usuario.

Lectura de las instrucciones de uso y requisitos aportados en la memoria.

20. Instrucciones de instalación/implantación

El usuario final (administrador, gestor y operario) no necesita realizar ninguna instalación, simplemente cumplir los requisitos mencionados anteriormente para poder acceder desde su dispositivo y a través de un navegador. Solamente el usuario operario tendrá que instalar además en su dispositivo móvil una aplicación similar a la que se ha mencionado anteriormente (Barcode Keyboard) para la lectura de los códigos de barras.

En el caso del equipo servidor es necesario instalar un servidor local, de esta manera se propone la siguiente forma de llevarse a cabo:

1. Descargar *Wampserver* desde la siguiente dirección: <http://www.wampserver.es/>
2. Instalar *Wampserver*.
3. Iniciar *Wampserver*.
4. Copiar la carpeta “processcore” y pegar en la siguiente dirección del servidor C:/wamp/www/.
5. Acceder a phpMyAdmin a través de la siguiente dirección <http://localhost/phpmyadmin/>
6. Crear una nueva base de datos llamada “pc_bbdd”.
7. Importar la base de datos (pc_bbdd.sql) que está ubicada en la carpeta processcore/bbdd.
8. Acceder a la aplicación a través de <http://localhost/processcore>

No obstante, actualmente no es necesario poner en marcha un servidor local para realizar tests y comprobar su funcionamiento, ya que está disponible en modo online en la siguiente dirección: <http://www.processcore.com>

21. Instrucciones de uso

Login



Es necesario disponer de un email y contraseña para acceder al sistema.

Menú principal

Por un lado se muestran los accesos concernientes a cada tipo de usuario en el contenedor principal. Por otro lado la cabecera, el pie y el contenedor secundario se mantienen para los usuarios administrador y gestor, sufriendo modificaciones estéticas según las resoluciones de los dispositivos, de manera que algunos de éstos pueden llegar a desaparecer por temas de legibilidad y espacio.

En la cabecera aparece el logo “**ProcessCore**” que permite siempre volver al “Menú Principal” y seguidamente bajo este la opción de “**Logout**” para que el usuario pueda salir del sistema. A la derecha se muestra un reloj que se actualiza cada minuto donde se puede ver la hora, año, mes y día actual.

El contenedor secundario está dividido en dos secciones, una sección superior que muestra los accesos directos y otra sección inferior para visualizar los últimos cinco movimientos teóricos.

El contenedor principal muestra la actividad principal de la aplicación, desde el menú principal hasta todas las funciones que se detallan en los siguientes puntos.

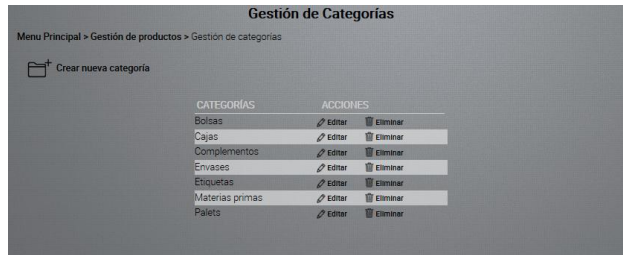
El pie muestra las siguientes opciones:

- Enviar consulta (para que los técnicos de la aplicación puedan resolver cualquier incidencia).
- Ayuda (manual de ayuda para el usuario).
- Visualizar versiones (muestra las versiones anteriores, actuales y futuras).
- W3C (comprobar que se cumplen los estándares web).
- Redes sociales (accesos a las redes donde se encuentra ProcessCore).



Gestión de categorías

Para visualizar las categorías es necesario acceder a través de “Gestión de productos”, y seleccionar “Gestión de categorías”. De esta manera se podrán visualizar las categorías, editarlas o eliminarlas. Finalmente para crear una nueva categoría se puede acceder desde la página “Gestión de categorías” y seleccionar “Crear nueva categoría”.



Gestión de productos

Para visualizar los productos es necesario acceder a través de “Gestión de productos”. De esta manera se podrán visualizar los productos existentes, editarlos y eliminarlos. Del mismo modo se puede visualizar filtrando los productos por categoría o realizando una búsqueda por producto o código.



Para crear un nuevo producto se puede acceder desde la página “Gestión de productos” y seleccionar “Crear nuevo producto”, donde será necesario cumplimentar el formulario. A tener en cuenta que **no pueden existir dos productos con el mismo código**.

Gestión de usuarios

Para visualizar los usuarios activos y el tipo de usuario correspondiente es necesario seleccionar “Gestión de usuarios” del “Menú principal”. Esto va a permitir de la misma manera editar y eliminar usuarios. Para crear un nuevo usuario en la página “Gestión de usuarios” seleccionar “Crear nuevo usuario”, donde será necesario cumplimentar los datos requeridos.

USUARIO	TIPO	ACCIONES
Administrador Principal	Administrador	Editar Eliminar
Gestor Oficina	Gestor	Editar Eliminar
Operario Planta	Operario	Editar Eliminar

Gestión de almacenes

Para crear almacenes es necesario acceder a través del icono “Gestión de almacenes”. En ésta página, a parte de visualizar, editar o eliminar almacenes, se encuentran las opciones de “Crear nuevo almacén” donde se cumplimentará un formulario para la creación de un nuevo almacén.

ALMACENES	ACCIONES
Almacén Principal	Editar Eliminar
Nave 1	Editar Eliminar
Nave 2	Editar Eliminar

Gestión de estanterías

En la página “Gestión de almacenes” se encuentra la opción “Gestión de estanterías” que permitirá crear nuevas estanterías y posiciones.

Por un lado, a la hora de crear una nueva estantería es importante tener en cuenta lo siguiente:

- **El código no puede ser igual para dos estanterías que están en un mismo almacén** (se recomienda crear códigos correlativos (1,2,3...)).
- **El número de filas está actualmente limitado a un máximo de 4**, ya que para los almacenes para los que está previsto esta es su máxima altura. En versiones posteriores se modificará para que el cliente pueda seleccionar la altura que desee.

Menu Principal > Gestión de almacenes > Gestión de estanterías > Nueva estantería

Almacén
Almacen Principal

Descripción

Código

Número de filas

Número de columnas

Aceptar Cancelar

Por otro lado, a la hora de editar una estantería, si se quiere modificar el número de filas o columnas, y previamente se han imprimido las posiciones, es necesario eliminar la estantería y crear una nueva.

Gestión de posiciones

Las posiciones se crean la primera vez que se imprimen. Después de esto se pueden imprimir tantas veces como se quiera. Se puede acceder desde la opción “Imprimir Posiciones” de la página “Gestión de estanterías”.

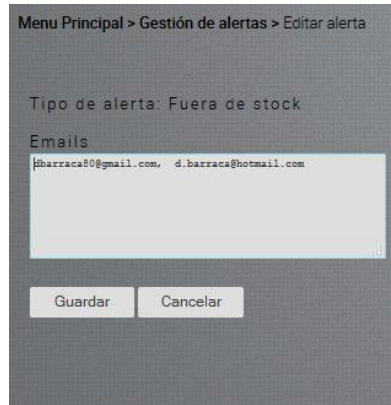
Menu Principal > Gestión de almacenes > Gestión de estanterías

Crear nueva estantería

ALMACÉN	ESTANTERÍA	CÓDIGO	FILAS	COLUMNAS	ACCIONES
1	Zona Produccion	1	4	5	Imprimir posiciones Editar Eliminar
1	Zona etiquetado	2	3	4	Imprimir posiciones Editar Eliminar

Gestión de alertas

Se accede a través de la opción “Gestión de alertas” del “Menú principal”. Permite editar las alertas actualmente activas: Fuera de stock y Punto de pedido. De esta manera, se puede añadir o eliminar cuentas de correo electrónico para que reciban las notificaciones correspondientes. Siempre que se añadan cuentas de correo se separarán mediante comas.



Fuera de stock: Enviará una notificación a las cuentas de correo introducidas cuando un producto tenga la cantidad inferior o igual a 0.

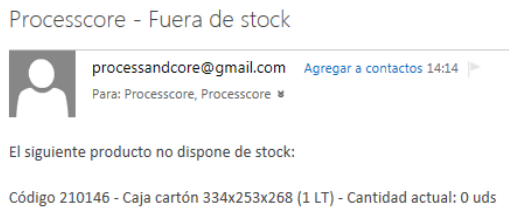


Figura 38: Email alerta fuera de stock

Punto de pedido. Enviará una notificación a las cuentas de correo introducidas cuando un producto tenga la cantidad igual o inferior al punto de pedido definido en el producto.

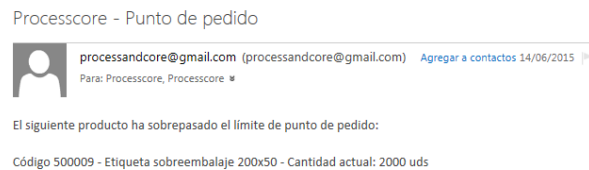


Figura 39: Email alerta punto de pedido

Inventario teórico

Se accede a través de la opción “Inventario Teórico” del “Menú principal”. Permite visualizar el inventario teórico de cada producto, donde se muestra los siguientes datos:

- Categoría
- Código
- Producto

- Stock total (stock teórico que se calcula por las entradas y salidas teóricas)
- Stock disponible (previsto para futuras versiones que registrará la cantidad de producto sin utilizar en producción).
- Stock en curso (previsto para futuras versiones que registrará la cantidad de producto empleado en producción).
- Stock físico (stock total físico que sirve para comparar los desajustes que se puedan ocasionar respecto al stock teórico).

Existe la opción de filtrar por categoría y de realizar la búsqueda por producto o código.

Inventario teórico

Menu Principal > Inventario teórico

Visualizar: Actualizar

Buscar: Aceptar

CATEGORÍA	CÓDIGO	PRODUCTO	STOCK TOTAL	STOCK DISPONIBLE	STOCK CURSO	STOCK FISICO
Materias primas	611413	DccNa 55	0	-	-	
Materias primas	611421	DccNa 60	0	-	-	
Materias primas	110241	Atcc grano 90	21000	-	-	21000
Materias primas	611364	Atcc polvo 90	0	-	-	
Materias primas	105031	Sulfato de cobre	3000	-	-	3000
Materias primas	110221	DccNa 55 Fino	0	-	-	
Bolsas	104006	Ácido Bórico	3000	-	-	3000
Bolsas	104092	Ácido cianúrico granulado	3000	-	-	3000
Bolsas	103092	Carbonato sódico	5000	-	-	25000
Materias primas	104009	Sulfato de aluminio	2000	-	-	2000
Complementos	210124	Cierres flecha 17mm	20000	-	-	20000
Complementos	210125	Cierres flecha 24mm	7500	-	-	7500
Complementos	210121	Dosificador 100ml	0	-	-	
Complementos	210122	Dosificador 25ml	0	-	-	
Palets	310085	Palet 100x100	0	-	-	
Palets	310086	Palet 80x120 Europalet	0	-	-	
Palets	310087	Palet 100x120 CP1	0	-	-	
Palets	310154	Palet 110x110 CP3	0	-	-	

Movimientos teóricos

Se accede a través de la opción “Movimientos teóricos” del “Menú principal”. Permite crear, editar o eliminar movimientos teóricos.

Para crear un nuevo movimiento teórico seguiremos los siguientes pasos:

1. Seleccionar “Nuevo movimiento teórico”.
2. Seleccionar “Nueva línea”.
3. De la ventana emergente seleccionar el producto, tipo de movimiento (entrada o salida) y la cantidad, y después Aceptar.

Guardar línea movimiento teórico

Buscar: Buscar

Producto:

Tipo de movimiento:

Cantidad:

Aceptar

4. Añadir tantas líneas como sea necesario.
5. Seleccionar Imprimir etiquetas de palet desde la opción disponible en cada línea y para cada producto.
6. En la página “Imprimir etiquetas de palet” introducir la cantidad de producto por palet e imprimir. La aplicación calcula que todos los palets tienen la misma cantidad que se ha introducido y si queda un resto lo pone en el último palet .
7. Imprimir etiquetas de palet.
8. Cumplimentar el formulario de la página “Movimiento teórico” (Fecha, referencia y observaciones) y Aceptar.

Menu Principal > Movimientos teóricos > Movimiento teórico

Fecha
2015-05-09

Referencia
Cajas

Observaciones
Albarán 1894563

Aceptar Cancelar

Líneas

Nueva línea

CÓDIGO	PRODUCTO	CANTIDAD	ACCIONES
210146	Caja cartón 334x253x268 (1 LT)	1500	Imprimir etiquetas palet Editar Eliminar
210147	Caja cartón 391x288x300 (5 LT)	2000	Imprimir etiquetas palet Editar Eliminar

Inventario físico

Se accede a través de la opción “Inventario físico” del “Menú principal”. Permite visualizar el inventario físico de cada producto, es decir, la suma de la cantidad actual de todos los palets. Se muestran los siguientes datos:

- Código
- Producto
- Posición en almacén (si no está posicionado se muestra “Sin posicionar”)
- Palet
- Cantidad

También al principio de cada producto se muestra el sumatorio de todos los palets. Existe la opción de realizar búsquedas por código, producto o palet.

Inventario físico

Menu Principal > Inventario físico

Buscar

CÓDIGO	PRODUCTO	POSICIÓN ALMACÉN	PALET	CANTIDAD
103092	CARBONATO SÓDICO	TOTAL: 25000		
103092	Carbonato sódico	Sin posicionar	P441	1000
103092	Carbonato sódico	Sin posicionar	P442	1000
103092	Carbonato sódico	Sin posicionar	P443	1000
103092	Carbonato sódico	Sin posicionar	P445	1000
103092	Carbonato sódico	Sin posicionar	P441	1000
103092	Carbonato sódico	Sin posicionar	P442	1000
103092	Carbonato sódico	Sin posicionar	P443	1000
103092	Carbonato sódico	Sin posicionar	P445	1000
103092	Carbonato sódico	Sin posicionar	P441	1000
103092	Carbonato sódico	Sin posicionar	P442	1000
103092	Carbonato sódico	Sin posicionar	P443	1000
103092	Carbonato sódico	Sin posicionar	P445	1000
103092	Carbonato sódico	Sin posicionar	P441	1000
103092	Carbonato sódico	Sin posicionar	P442	1000
103092	Carbonato sódico	Sin posicionar	P443	1000
103092	Carbonato sódico	Sin posicionar	P445	1000
103092	Carbonato sódico	Sin posicionar	P441	1000

Movimientos físicos

Se accede a través de la opción “Movimientos físicos” del “Menú principal”. Existen dos tipos de movimiento: movimiento físico de entrada y movimiento físico de salida.

Movimiento físico de entrada

Se puede realizar la entrada mediante la lectura de código de barras o a través del nombre alfanumérico correspondiente, tanto para el campo “Palet” como para el campo “Posición”. Por esto, a cada código de barras siempre le acompaña su valor alfanumérico.

Menu Principal > Movimiento físico > Movimiento físico - ENTRADA

Palet

Posición

Movimiento físico de salida

Se puede realizar la salida mediante la lectura de código de barras o a través del nombre alfanumérico correspondiente para el campo “Palet”, e introduciendo la cantidad a extraer.

Menu Principal > Movimiento físico > Movimiento físico - SALIDA

Palet

Cantidad

22. Proyección a futuro

El tiempo que se dispone para realizar el proyecto elegido es bastante limitado, ya que implica mucha investigación y aprendizaje de nuevos elementos. Debido a esto, en este proyecto se presenta la primera parte concerniente a la gestión de productos, usuarios y almacenes dentro de un proceso productivo. Por esto, para esta primera versión presentada se incluirán mejoras y correcciones de errores.

A continuación se presentan algunas mejoras de la primera versión presentada:

- Mejorar la seguridad en el login.
- Mejorar la entrada de movimientos teóricos.
- Añadir atributos a los usuarios como fotografías, confirmación de contraseña...
- Mejorar las búsquedas y filtros.
- Mayor claridad para la impresión de etiquetas de palet y posiciones.
- Mostrar un calendario a la hora de introducir la fecha
- Control del tiempo en almacén de cada palet.
- Búsqueda de movimientos teóricos por producto.
- Búsqueda de movimientos físicos por producto.
- Controlar movimientos por usuario.

Para próximas versiones, a grandes rasgos se realizarán las siguientes implantaciones:

- Gestión de clientes
- Gestión de proveedores
- Gestión de pedidos
- Gestión de albaranes
- Gestión de facturas
- Gestión de expediciones
- Planificación de la producción
- Gestión de líneas de producción
- Generación hojas de trabajo (*kanbans*)
- Control de producción (tiempos, número empleados...)
- Gráficos de resultados

Finalmente, en principio se está construyendo la aplicación web adaptada a la empresa en la cual actualmente desempeño mi trabajo, pero no descarto que se oriente a una aplicación web genérica con grandes posibilidades de adaptabilidad a diferentes compañías. De momento se continuará con el desarrollo ya que puede resultar bastante útil tanto para mi empresa, como para en un futuro, adaptarse a otros procesos productivos.

23. Conclusión

La realización de este trabajo de fin de grado durante estos últimos cuatro meses ha supuesto un reto personal. Sin embargo, el tiempo limitado tanto por motivos laborales como por la magnitud del proyecto me han obligado a realizar un esfuerzo adicional sin obtener unos resultados 100% satisfactorios. Por esto, no he podido plasmar completamente los conocimientos adquiridos durante todos estos años en el grado multimedia.

El proyecto abarca muchas competencias, desde las áreas de diseño web (arquitectura de la información, usabilidad, accesibilidad, lenguajes y estándares web...), el diseño gráfico, la administración y gestión de proyectos, la programación (análisis y diseño de patrones, bases de datos, programación orientada a objetos...) por nombrar algunas ellas. Por esto, he intentado ceñirme a los contenidos adquiridos.

Soy consciente que si hubiera utilizado APIs como Bootstrap, jquery u otros, la tarea podría haber sido más sencilla. No me gusta carecer del control de una función sin saber cómo trabaja, y sabía que no disponía de tiempo suficiente para comprender completamente este tipo de APIs. Seguramente ya tendré tiempo de aprender y utilizar estas APIs, pero pienso que primero es comprender como funcionan los lenguajes de programación.

Por esto, a parte de otras competencias, quería aprender a programar en un lenguaje orientado a objetos y en MVC. Es decir, crear modelos, vistas y controladores con sus funciones, y por supuesto, buscando soluciones a la multitud de diferentes problemas que se planteaban.

Por último, ahora me doy cuenta que el proyecto que en principio tenía en mente era bastante ambicioso. Por esto, en este trabajo de fin de grado se presenta la primera versión de la aplicación web a la que aún le faltan mejoras y corrección de errores. Y aunque no estoy totalmente satisfecho del trabajo realizado, estoy bastante contento ya que he puesto en marcha un proyecto desde cero y creo que según vaya adquiriendo más experiencia puede avanzar o llevarme hacia otros proyectos personales ilusionantes.

Finalmente agradecer a todos mis familiares, amigos y compañeros, así como la ayuda que me ha proporcionado la UOC durante la realización de este trabajo final de grado.

Anexo 1. Entregables del proyecto

- Memoria del proyecto (pdf)
- Proyecto (zip)

Incluye todos los archivos que forman el proyecto:

- Modelos
 - Vistas
 - Controladores
 - Librerías
 - CSS
 - Javascript
 - Fuentes
 - Imágenes
- Autoinforme (pdf)
 - Vídeo (mp4)
 - Presentación (Prezi) (pdf)
http://prezi.com/lobix1swv5qe/?utm_campaign=share&utm_medium=copy
 - Enlace web: Enlace a la aplicación web en funcionamiento: <http://www.processcore.com>

Anexo 2. Código fuente (extractos)

A continuación se van a exponer algunos extractos de código fuente. La aplicación se ha desarrollado mediante el MVC (Modelo, vista y controlador). Los modelos representan las clases de la aplicación, las vistas se encargan de imprimir cada pantalla y recoger las acciones que los usuarios solicitan, y los controladores se encargan de ejecutar las diferentes funciones. Cada controlador se define, de manera que cada vista tiene la función de comprobar que si no está definido el controlador no se ejecuta.

Página de acceso a la aplicación donde el usuario debe introducir los datos para acceder (vista y controlador).

index.php

Controlador

```
<?php
session_start();
define('CONTROLADOR', TRUE);

require_once 'modelos/Conexion.php';

if(isset($_POST['submit'])){
    $errMsg = '';
    //usuario y password que se envían desde el formulario
    $email = trim($_POST['email']);
    $pass = trim($_POST['pass']);

    if($email == '')
        $errMsg .= 'Debe introducir el email<br>';

    if($pass == '')
        $errMsg .= 'Debe introducir el password<br>';

    if($errMsg == ''){
        $conexion = new Conexion();
        $consulta = $conexion->prepare('SELECT * FROM usuario
                                        WHERE email = :email');
        $consulta->bindParam(':email', $email);
        $consulta->execute();
        $registro = $consulta->fetch(PDO::FETCH_ASSOC);
        if(count($registro) > 0 && $pass==$registro['pass']){
            $_SESSION['email'] = $registro['email'];
            $_SESSION['nombre'] = $registro['nombre'];
            $_SESSION['apellido'] = $registro['apellido'];
            $_SESSION['id_tipo_usuario'] = $registro['id_tipo_usuario'];
            header('location:processcore.php');
            exit;
        }else{
            $errMsg .= 'Usuario y password incorrectas<br>';
        }
    }
}

include 'vistas/index.php';

if(isset($errMsg)){
    echo '<div style="color:#FF0000;text-align:center;font-size:12px;">'.$errMsg.'</div>';
}
?>
```

Figura 40: Código Login (controlador)

Vista

```

<?php if (!defined('CONTROLADOR')) exit; ?>

<[DOCTYPE html]
<html lang="es"
<head>
  <title>ProcessCore</title>
  <meta charset="utf-8"/>
  <meta name="description" content="ProcessCore - Sistema de gestión para producciones industriales"/>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <link rel="shortcut icon" type="image/x-icon" href="imagenes/favicon_processcore.ico"/>
  <link rel="author" type="text/plain" href="otros/humans.txt"/>
  <link rel="stylesheet" type="text/css" media="all" href="css/estilo_index.css"/>
  <!--[if !IE 9]>
    <script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
</head>
<body>
  <form id="login" action="index.php" method="post">
    <h1>Log In</h1>
    <fieldset id="inputs">
      <input id="email" name="email" type="text" placeholder="Email" value="<?php if (isset($_COOKIE['email'])) echo $_COOKIE['email'];?>" autofocus required>
      <input id="pass" name="pass" type="password" placeholder="Password" required>
    </fieldset>
    <fieldset id="actions">
      <input type="submit" id="submit" name="submit" value="Log in">
    </fieldset>
  </form>
</body>
</html>

```

Figura 41: Código Login (vista)

Se llama al controlador index.php, el cual comprueba si existe algún envío de información vía \$_POST y muestra la vista index.php. Si existe envío vía \$_POST el controlador comprueba que los datos obtenidos concuerdan con los de acceso, y si son correctos redirecciona a la página principal de la aplicación processcore.php. Si son erróneos devuelve un mensaje de error en la misma página del login.

Clase Conexión.php

Modelo

```

<?php

if (!defined('CONTROLADOR')) exit;

class Conexion extends PDO {
  private $tipo_de_base = 'mysql';
  private $host = 'localhost';
  private $nombre_de_base = 'pc_bbdd';
  private $usuario = 'root';
  private $contrasena = '';

  public function __construct() {

    // Sobreescribo el método constructor de la clase PDO.
    try{
      parent::__construct($this->tipo_de_base, 'host='.$this->host.', dbname='.$this->nombre_de_base, $this->usuario, $this->contrasena);
    }catch(PDOException $e){
      echo "Se ha producido un error y no se puede conectar a la base de datos. Detalle: " . $e->getMessage();
      exit;
    }
  }
}
?>

```

Figura 42: Código Conexión (modelo)

La clase conexión va a servir para conectarnos a la base de datos. Se puede observar los atributos privados que albergan los datos necesarios para conectar a la base datos: el tipo de *bbdd*, el *host*, el nombre de la *bbdd*, el usuario y la contraseña. Y finalmente la función constructora para llevar a cabo la conexión. Esta clase será requerida por varios controladores para poder trabajar con la base de datos. También se ha añadido una excepción en el caso de que no se pueda conectar con la *bbdd*.

Página principal de la aplicación web (controlador y vista)

processcore.php

Controlador

```
<?php
define('CONTROLADOR', TRUE);
session_start();

if($_SESSION['id_tipo_usuario']==1)
{
$menu_principal= "menu_principal_adm.php";
}
elseif($_SESSION['id_tipo_usuario']==2)
{
$menu_principal= "menu_principal_ges.php";
}
elseif($_SESSION['id_tipo_usuario']==3)
{
$menu_principal= "menu_principal_op.php";
}

require_once 'vistas/processscore.php';
?>
```

Figura 43: Código Página principal (controlador)

Comprueba si la sesión existe, y según el tipo de usuario carga un menú u otro.

Vista (1ª parte)

```

<!DOCTYPE html>
<html lang="es">
<head>
  <title>ProcessCore</title>
  <meta charset="utf-8"/>
  <meta name="description" content="ProcessCore - Sistema de gestión para producciones industriales"/>
  <meta name="viewport" content="width=device-width, initial-scale=1"/>
  <link rel="shortcut icon" type="image/x-icon" href="imagenes/favicon_processcore.ico"/>
  <link rel="author" type="text/plain" href="qtros/humans.txt"/>
  <link rel="stylesheet" type="text/css" media="all" href="css/estilo_general.css"/>
  <script src="js/tiempo.js"></script>
  <!--[if lt IE 9]>
  <script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
</head>
<body onload="tiempo()">
  <header>
    <div id="cabecera">
      <figure>
        <a href="processcore.php">
          
        </a>
      </figure>
      <h1>
        ProcessCore <br /> Sistema de gestión para producciones industriales
      </h1>
      <section id="tiempo">
        <p>| HORA</p>
        <p id="hora" class="hora"></p>
        <p>| AÑO / MES</p>
        <p id="ano_mes" class="fecha"></p>
        <p>| FECHA / DIA</p>
        <p id="dia" class="fecha"></p>
      </section>
      <section id="login">
        <ul>
          <li><?php echo 'Bienvenido ' . $_SESSION['nombre'] . ' ' . $_SESSION['apellido']; ?></li>
          <li><a href='logout.php'>Logout</a></li>
        </ul>
      </section>
    </div>
  </header>

```

Figura 44: Código Página principal (vista) 1ª parte

Por un lado, en esta primera parte se ve representado el head de la página donde se ubica el título de la página, los *meta* que entre otros permiten definir la descripción de la página o el icono de favoritos, y los *link rel* que permiten importar la hoja de estilos y un archivo de *javascript* para mostrar la fecha y hora.

Por otro lado, se encuentra el inicio del *body*, el cual al cargarse llama a una función *javascript* tiempo() que permite ver en el *header* la fecha y hora. En el inicio del *body* se puede observar el *header* que contiene el logo de la aplicación, el título *h1*, y la representación gráfica de la mencionada fecha y hora. Por último, se muestra el nombre del usuario que ha entrado en el sistema y la opción del *logout*.

Vista (2ª parte)

```

<div id="contenedor">
  <div id="contenedor_principal">
    <?php
      //Si no se ha especificado ninguna página se cargará la página por defecto
      $default_pag='procesacore.php';
      $pag= isset($_GET["pag"]) && !empty($_GET["pag"]) ? $_GET["pag"] : $default_pag;
      //ruta al directorio de los ficheros
      $path="/";
      //nombre del fichero de contenido
      $pagina=$pag.".php";
      //Test. Existe el fichero de contenido?
      $file=$path.$pagina;
      if(file_exists($file))
        include_once($pagina);
      else
        include $menu_principal;
    ?>
  </div>

  <div id="contenedor_secundario">
    <section id="contenido_superior">
      <h4>Accesos directos</h4>
      <ul>
        <li><a href="procesacore.php?pag=nuevo_movimiento_teorico" title="Ir a Nueva movimiento teórico">
          
        </a>
        <a href="procesacore.php?pag=nuevo_movimiento_teorico" title="Ir a Nueva movimiento teórico">Nueva movimiento teórico.</a>
        </li>
        <li>
          <a href="procesacore.php?pag=movimiento_fisico_entrada" title="Ir a Nueva entrada física">
            
          </a>
          <a href="procesacore.php?pag=movimiento_fisico_entrada" title="Ir a Nueva entrada física">Nueva entrada física.</a>
        </li>
        <li>
          <a href="procesacore.php?pag=movimiento_fisico_salida" title="Ir a Nueva salida física">
            
          </a>
          <a href="procesacore.php?pag=movimiento_fisico_salida" title="Ir a Nueva salida física">Nueva salida física.</a>
        </li>
        <li>
          <a href="procesacore.php?pag=gestion_almacenes" title="Ver almacenes">
            
          </a>
          <a href="procesacore.php?pag=gestion_almacenes" title="Ver almacenes">Ver almacenes.</a>
        </li>
      </ul>
    </section>
    <section id="contenido_inferior">
      <h4>Últimos movimientos teóricos</h4>
      <?php include "ultimos_movimientos_teoricos.php";?>
    </section>
  </div>
</div>

```

Figura 45: Código Página principal (vista) 2ª parte

Aquí se puede observar un contenedor que alberga un contenedor principal y un contenedor secundario. En el contenedor principal, para una mejor comprensión se ha colocado las funciones que harán que se carguen las diferentes páginas en este div que se envían mediante \$_GET. De esta manera se cargará por defecto el menú principal, y si existe la página llamada se cargará en este div, manteniendo el resto de la página invariable.

En el contenedor secundario existen dos secciones, una superior donde se alojan cuatro accesos directos, y otro inferior donde se alojan los cinco últimos movimientos teóricos (se importa a través de un include).

Vista (3ª parte)

```

<footer>
  <div id="pie">
    <h4>Explora ProcessCore</h4>
    <ul>
      <li>
        <h5>Contacto</h5>
        <a href="processcore.php?pag=contacto" title="Ir a contacto">Enviar consulta</a>
      </li>
      <li>
        <h5>Soporte</h5>
        <a href="#" title="Ayuda">Consulta la ayuda</a>
      </li>
      <li>
        <h5>ProcessCore</h5>
        <a href="#" title="Ir a Sdora">Visualizar versiones</a>
      </li>
      <li>
        <h5>Estándares web</h5>
        <a href="http://validator.w3.org/check?uri=referer">
      </li>
      <li>
        <h5>Descubre</h5>
        <a href="https://www.facebook.com/processcore" title="Ir a Facebook ProcessCore" target="_blank">
      </li>
      <li>
        <p>© 2015 ProcessCore</p>
      </li>
    </ul>
  </div>
</footer>
</body>
</html>

```

Figura 46: Código Página principal (vista) 3ª parte

Por último se muestra el *footer* donde una lista muestra las diferentes opciones que se enumeran a continuación: enviar una consulta, consultar la ayuda, visualizar las versiones, comprobar los estándares web W3C y en que redes sociales se encuentra ProcessCore.

Clase Producto.php

Modelo (1ª parte)

```

<?php
if (!defined('CONTROLADOR')) exit;
require_once 'Controlador.php';

class Producto {

    private $id;
    private $id_categoria;
    private $descripcion;
    private $codigo;
    private $punto_pedido;
    const TABLA = 'productos';

    public function __construct ($id_categoria=null, $descripcion=null, $codigo=null, $punto_pedido=null, $id=null){
        $this->id_categoria=$id_categoria;
        $this->descripcion=$descripcion;
        $this->codigo=$codigo;
        $this->punto_pedido=$punto_pedido;
        $this->id=$id;
    }

    public function getId(){
        return $this->id;
    }

    public function getId_categoria(){
        return $this->id_categoria;
    }

    public function getDescripcion(){
        return $this->descripcion;
    }

    public function getCodigo(){
        return $this->codigo;
    }

    public function getPunto_pedido(){
        return $this->punto_pedido;
    }

    public function setId_categoria($id_categoria){
        $this->id_categoria = $id_categoria;
    }

    public function setDescripcion($descripcion){
        $this->descripcion = $descripcion;
    }

    public function setCodigo($codigo){
        $this->codigo = $codigo;
    }

    public function setPunto_pedido($punto_pedido){
        $this->punto_pedido = $punto_pedido;
    }
}

```

Figura 47: Código Producto (modelo) 1ª parte

La clase producto donde se observa los atributos, la función constructora y los getters y setters de acceso a los atributos.

Modelo (2ª parte)

```

public function guardar(){
    $conexion = new Conexion();
    if($this->id){
        $consulta = $conexion->prepare('UPDATE `self::TABLA` SET id_categoria = :id_categoria, descripcion = :descripcion, codigo = :codigo, punto_pedido = :punto_pedido WHERE id = :id');
        $consulta->bindParam(':id_categoria', $this->id_categoria);
        $consulta->bindParam(':descripcion', $this->descripcion);
        $consulta->bindParam(':codigo', $this->codigo);
        $consulta->bindParam(':punto_pedido', $this->punto_pedido);
        $consulta->bindParam(':id', $this->id);
        $consulta->execute();
    }else{
        $consulta = $conexion->prepare('INSERT INTO `self::TABLA` (id_categoria, descripcion, codigo, punto_pedido) VALUES(:id_categoria, :descripcion, :codigo, :punto_pedido)');
        $consulta->bindParam(':id_categoria', $this->id_categoria);
        $consulta->bindParam(':descripcion', $this->descripcion);
        $consulta->bindParam(':codigo', $this->codigo);
        $consulta->bindParam(':punto_pedido', $this->punto_pedido);
        $consulta->execute();
        $this->id = $conexion->lastInsertId();
    }
    $conexion = null;
}

public function eliminar(){
    $conexion = new Conexion();
    $consulta = $conexion->prepare('DELETE FROM `self::TABLA` WHERE id = :id');
    $consulta->bindParam(':id', $this->id);
    $consulta->execute();
    $conexion = null;
}

public static function mostrarTodos(){
    $conexion = new Conexion();
    $consulta = $conexion->prepare('SELECT id, id_categoria, descripcion, codigo, punto_pedido FROM `self::TABLA` ORDER BY id_categoria');
    $consulta->execute();
    $registros = $consulta->fetchAll();
    return $registros;
}

public static function buscarPorId($id) {
    $conexion = new Conexion();
    $consulta = $conexion->prepare('SELECT id_categoria, descripcion, codigo, punto_pedido FROM `self::TABLA` WHERE id = :id');
    $consulta->bindParam(':id', $id);
    $consulta->execute();
    $registro = $consulta->fetch();
    $conexion = null;
    if ($registro) {
        return new self($registro['id_categoria'], $registro['descripcion'], $registro['codigo'], $registro['punto_pedido'], $id);
    } else {
        return false;
    }
}
}

```

Figura 48: Código Producto (modelo) 2ª parte

Aquí se muestra la función guardar() que incluye actualizar o insertar un producto en la base de datos, la función eliminar() que borra el producto, la función mostrarTodos() que devuelve todos los productos y la función buscarPorId() que devuelve el producto que sea igual al parámetro pasado a dicha función.

Modelo (3ª parte)

```

//función para mostrar productos con su categoría en inventario teórico
public static function mostrarProductos(){
    $conexion = new Conexion();
    $consulta = $conexion->prepare('SELECT producto.id, producto.descripcion, producto.codigo, producto.punto_pedido, categoria.nombre, linea_movimiento_teorico.id_tipo_movimiento,
SUM(IF(linea_movimiento_teorico.id_tipo_movimiento=1,linea_movimiento_teorico.cantidad,linea_movimiento_teorico.cantidad*(-1)) AS total FROM producto
INNER JOIN categoria ON producto.id_categoria=categoria.id
LEFT JOIN linea_movimiento_teorico ON producto.id=linea_movimiento_teorico.id_producto GROUP BY producto.id
');
    $consulta->execute();
    $registros = $consulta->fetchAll();
    return $registros;
}

//función para mostrar productos según la búsqueda
public static function mostrarBusqueda($busqueda){
    $conexion = new Conexion();
    $consulta = $conexion->prepare('SELECT producto.id, producto.descripcion, producto.codigo, producto.punto_pedido, categoria.nombre, linea_movimiento_teorico.id_tipo_movimiento,
SUM(IF(linea_movimiento_teorico.id_tipo_movimiento=1,linea_movimiento_teorico.cantidad,linea_movimiento_teorico.cantidad*(-1)) AS total FROM producto
INNER JOIN categoria ON producto.id_categoria=categoria.id
LEFT JOIN linea_movimiento_teorico ON producto.id=linea_movimiento_teorico.id_producto
WHERE producto.codigo like :busqueda or producto.descripcion like :busqueda
GROUP BY producto.id
');
    $busqueda = "%".$busqueda."%";
    $consulta->bindParam(':busqueda', $busqueda);
    $consulta->execute();
    $registros = $consulta->fetchAll();
    return $registros;
}
}

```

```

//función para mostrar productos según la categoría
public static function mostrarVisualizar($visualizar){
    $conexion = new Conexion();
    $consulta = $conexion->prepare('SELECT producto.id, producto.descripcion, producto.codigo, producto.punto_pedido, categoria.nombre, linea_movimiento_teorico.id_tipo_movimiento,
SUM(IF(linea_movimiento_teorico.id_tipo_movimiento=1,linea_movimiento_teorico.cantidad,linea_movimiento_teorico.cantidad*(-1))) AS total FROM producto
INNER JOIN categoria ON producto.id_categoria=categoria.id
LEFT JOIN linea_movimiento_teorico ON producto.id=linea_movimiento_teorico.id_producto
WHERE categoria.id_categoria = :visualizar
GROUP BY producto-id
');
    $consulta->bindParam(':visualizar', $visualizar);
    $consulta->execute();
    $registros = $consulta->fetchAll();
    return $registros;
}

//función para buscar productos por código
public static function buscarPorCodigo($codigo) {
    $conexion = new Conexion();
    $consulta = $conexion->prepare('SELECT codigo FROM ' . self::$TABLA . ' WHERE codigo = :codigo');
    $consulta->bindParam(':codigo', $codigo);
    $consulta->execute();
    $registro = $consulta->fetch();
    $conexion = null;
    if ($registro) {
        return true;
    } else {
        return false;
    }
}

```

Figura 49: Código Producto (modelo) 3ª parte

La función `mostrarProductos()` devuelve los productos con su categoría para su visualización en el inventario teórico, ya que se han combinado las tablas categoría y producto con `INNER JOIN` y a su vez con la tabla `línea_movimiento_teorico` a través de `LEFT JOIN` para poder calcular y mostrar el stock teórico de cada producto finalmente agrupado por `GROUP BY`.

La función `mostrarBusqueda($busqueda)`, además de lo anteriormente citado se encarga de realizar la búsqueda solicitada por el usuario y devolver los datos filtrados por producto o código a través de la orden `WHERE` y `like`.

La función `mostrarVisualizar($visualizar)`, realiza la misma función que la anterior pero en este caso permite en el inventario teórico filtrar por tipo de categoría.

Por último, la función `buscarPorCodigo($codigo)` devuelve `true` si existe y `false` si no existe, y de esta manera al crear un nuevo producto comprobará la existencia del código o no antes de insertarlo en la base de datos.

Inventario teórico (controlador y vista).

inventario_teorico.php

Controlador

```

<?php
define('CONTROLADOR_INVENTARIO_TEORICO', TRUE);

//Para mostrar los productos
require_once 'modelos/Producto.php';
require_once 'modelos/Palet.php';
require_once 'modelos/Categoria.php';

$categorias=Categoria::mostrarTodos();

$busqueda=(isset($_REQUEST['busqueda'])) ? $_REQUEST['busqueda'] : null;
$visualizar=(isset($_REQUEST['visualizar'])) ? $_REQUEST['visualizar'] : null;
$visualiza="";
$busca="";

if ($busqueda) {
    $productos = Producto::mostrarBusqueda($busqueda);
} else {
    $productos = Producto::mostrarProductos();
}

if ($visualizar){
    $productos = Producto::mostrarVisualizar($visualizar);
    $visualiza="Se muestran los productos de la categoría: ".Categoria::devuelveCategoria($visualizar);
}

require_once 'modelos/LineaMovimientoTeorico.php';

include 'vistas/inventario_teorico.php';
?>

```

Figura 50: Código Inventario teórico (controlador)

Se define primeramente el controlador y se incluyen las clases necesarias (Producto, Palet y Categoría). Se almacena en la variable `$categorias` los datos de las categorías para ser mostradas en el listado y poder filtrar. En las variables `$busqueda` y `$visualizar` se almacena la búsqueda o filtro realizada por el usuario, sino su valor es `null`. Si disponen de datos la variable `$busqueda` o `$visualizar` se ejecuta una sentencia u otra y se almacenan en la variable `$productos`. Esta variable por defecto muestra todas los productos del inventario teórico.

Vista

```

<?php if (!defined('CONTROLADOR_INVENTARIO_TEORICO')) exit; ?>

<h2>Inventario teórico</h2>

<nav class="breadcrumb">
<ul>
<li><a href="procesocore.php">Menu Principal </a></li>
<li><p>Inventario teórico</p></li>
</ul>
</nav>

<form method="post" action="procesocore.php?page=inventario_teorico" class="seleccion">
<label for="visualizar"> Visualizar </label>
<?php if (count($categorias) > 0): ?>
<select name="visualizar">
<option value="1">Seleccionar</option>
<?php foreach ($categorias as $item): ?>
<option value="<?php echo $item['id']?>"><?php echo $item['nombre']?></option>
</select>
<?php endif; ?>
<input type="submit" class="submit" value="Actualizar" />
</form>

<form method="post" action="procesocore.php?page=inventario_teorico" class="busqueda">
<label for="buscar"> Buscar </label>
<input type="text" name="busqueda" id="busqueda" value="" placeholder="Escriba aquí el código o producto"
pattern="[a-zA-Z0-9 ÆÁÏÏÏ]([3,20])$" title="Solamente valores alfanuméricos. Min:3 Max:20" autofocus required />
<input type="submit" class="submit" value="Aceptar" />
</form>

```

```

<p><?php echo $visualiza; ?></p>
<p><?php echo $busca; ?></p>

<?php if (count($productos) > 0): ?>
<table id="inventario_teorico">
  <thead>
    <tr>
      <th scope="col">CATEGORIA</th>
      <th scope="col">CÓDIGO</th>
      <th scope="col">PRODUCTO</th>
      <th scope="col">STOCK TOTAL</th>
      <th scope="col">STOCK DISPONIBLE</th>
      <th scope="col">STOCK CURSO</th>
      <th scope="col">STOCK FÍSICO</th>
    </tr>
  </thead>
  <tbody>
    <?php foreach ($productos as $item): ?>
      <tr>
        <td><?php echo $item['categoria']; ?></td>
        <td><?php echo $item['codigo']; ?></td>
        <td><?php echo $item['producto']; ?></td>
        <td><?php
          if($item['total']==null){
            echo "0";
          }else{
            echo $item['total'];
          }
        ?></td>
        <td></td>
        <td></td>
        <td>
          <?php
            $cantidad_palets= Palet::cantidadTotal($item['codigo']);
            foreach ($cantidad_palets as $item2):
              echo $item2['total'];
            endforeach;
          ?>
        </td>
      </tr>
    </tbody>
  </table>

```

Figura 51: Código Inventario teórico (vista)

Primeramente comprueba que el controlador está definido, y después muestra el título *h2*, el navegador *breadcrumbs*, el formulario para filtrar por categoría, el formulario de búsqueda e imprime el valor buscado o filtrado si existe. Por último se muestra la tabla que contiene todos los datos, y que mediante un *foreach* recuperamos de la variable *\$productos* que contiene la información. Dentro del *foreach* se ha añadido un condicional por si la variable devuelve *null* que imprima 0, y otro *foreach* para recuperar el stock físico actual de cada producto.

Imprimir etiquetas de palet (controlador).

pdf_palets.php

Controlador (1ª parte)

```
<?php
define('CONTROLADOR', TRUE);

require_once 'modelos/Palet.php';
require_once 'modelos/Producto.php';
require_once 'librerias/fpdf.php';
require_once 'librerias/code128.php';

$palet_id = (isset($_REQUEST['palet_id']) ? $_REQUEST['palet_id'] : null);
$cantidad = (isset($_REQUEST['cantidad']) ? $_REQUEST['cantidad'] : null);
$cantidad_palet = (isset($_REQUEST['cantidad_palet']) ? $_REQUEST['cantidad_palet'] : null);
$id_producto = (isset($_REQUEST['id_producto']) ? $_REQUEST['id_producto'] : null);
$id_posicion = (isset($_REQUEST['id_posicion']) ? $_REQUEST['id_posicion'] : null);
$ultima_id = (isset($_REQUEST['ultima_id']) ? $_REQUEST['ultima_id'] : null);
$linea_movimiento_teorico_id = (isset($_REQUEST['linea_movimiento_teorico_id']) ? $_REQUEST['linea_movimiento_teorico_id'] : null);

$nombre="P".$ultima_id."".$linea_movimiento_teorico_id;

if($palet_id){
    $palet = Palet::buscarPorId($palet_id);
}else{
    $palet = new Palet();
}

$producto=Producto::buscarPorId($id_producto);
$nombre_producto=$producto->getDescripcion();
$codigo_producto=$producto->getCodigo();

$i=1;
$pdf = new PDF_Code128('P','mm','A4');
```

Figura 52: Código Imprimir etiquetas de palet (controlador) 1ª parte

Se incluyen las clases Palet y Producto, así como las librerías “fpdf” y “code128”. Se recogen los datos enviados a través de \$_GET y se almacenan en las diferentes variables.

En la variable \$nombre se genera el nombre principal del palet con la combinación de strings y variables, y después a cada palet se le añadirá un número consecutivo con el fin de obtener un único identificador para cada palet, y que se almacenará en la variable \$nombre1.

Se crea un nuevo PDF (\$pdf) que permita la generación de códigos de barra, en formato vertical y tamaño A4.

Controlador (2ª parte)

```

//para los palets por cantidad y los introduce en la bdd
while ($cantidad>0){

    if ($cantidad >= $cantidad_palet){

        $nombre=$nombre."".$i;
        $i++;

        if ($i%2== 0) {
            $pdf->AddPage();
            $pdf->SetFont('Arial','B',24);
            $pdf->Image('imagenes/logo_processcore_2.png',12,12,22);
            $pdf->Cell(0,50,utf8_decode($nombre_producto),1,1,'C');
            $pdf->SetFont('Arial','B',16);
            $pdf->Cell(60,20,'Codigo: '.$codigo_producto,0,1,'L');
            $pdf->SetFont('Arial','B',12);
            $pdf->Cell(60,20,'Cantidad inicial: '.$cantidad_palet.' uds-',0,1,'L');
            $pdf->Cell(60,20,'Palet: '.$nombre,0,1,'L');
            //Para código de barras
            $pdf->Code128(120,85,$nombre,80,20);
            $pdf->SetXY(120,110);
            $pdf->Write(5,$nombre);
        } else {
            $pdf->Ln(45);
            $pdf->SetFont('Arial','B',24);
            $pdf->Image('imagenes/logo_processcore_2.png',12,157,22);
            $pdf->Cell(0,50,utf8_decode($nombre_producto),1,1,'C');
            $pdf->SetFont('Arial','B',16);
            $pdf->Cell(60,20,'Codigo: '.$codigo_producto,0,1,'L');
            $pdf->SetFont('Arial','B',12);
            $pdf->Cell(60,20,'Cantidad inicial: '.$cantidad_palet.' uds-',0,1,'L');
            $pdf->Cell(60,20,'Palet: '.$nombre,0,1,'L');
            //Para código de barras
            $pdf->Code128(120,225,$nombre,80,20);
            $pdf->SetXY(120,250);
            $pdf->Write(5,$nombre);
        }

        $palet = new Palet();
        $palet->setId_producto($id_producto);
        $palet->setId_posicion($id_posicion);
        $palet->setNombre($nombre);
        $palet->setCantidad($cantidad_palet);
        $palet->guardar();

        $cantidad=$cantidad-$cantidad_palet;
    } else {

        $nombre=$nombre."".$i;
        $i++;
    }
}

```

```

if ($i%2== 0) {
    $pdf->AddPage();
    $pdf->SetFont('Arial','B',24);
    $pdf->Image('imagenes/logo_processcore_2.png',12,12,22);
    $pdf->Cell(0,50,utf8_decode($nombre_producto),1,1,'C');
    $pdf->SetFont('Arial','B',16);
    $pdf->Cell(60,20,'Codigo: '.$codigo_producto,0,1,'L');
    $pdf->SetFont('Arial','B',12);
    $pdf->Cell(60,20,'Cantidad inicial: '.$cantidad.' uds-',0,1,'L');
    $pdf->Cell(60,20,'Palet: '.$nombre,0,1,'L');
    //Para código de barras
    $pdf->Code128(120,85,$nombre,80,20);
    $pdf->SetXY(120,110);
    $pdf->Write(5,$nombre);
} else {
    $pdf->Ln(45);
    $pdf->SetFont('Arial','B',24);
    $pdf->Image('imagenes/logo_processcore_2.png',12,157,22);
    $pdf->Cell(0,50,utf8_decode($nombre_producto),1,1,'C');
    $pdf->SetFont('Arial','B',16);
    $pdf->Cell(60,20,'Codigo: '.$codigo_producto,0,1,'L');
    $pdf->SetFont('Arial','B',12);
    $pdf->Cell(60,20,'Cantidad inicial: '.$cantidad.' uds-',0,1,'L');
    $pdf->Cell(60,20,'Palet: '.$nombre,0,1,'L');
    //Para código de barras
    $pdf->Code128(120,225,$nombre,80,20);
    $pdf->SetXY(120,250);
    $pdf->Write(5,$nombre);
}

$palet = new Palet();
$palet->setId_producto($id_producto);
$palet->setId_posicion($id_posicion);
$palet->setNombre($nombre);
$palet->setCantidad($cantidad);
$palet->guardar();
$cantidad=$cantidad-$cantidad;
}

$pdf->Output();
}

```

Figura 53: Código Imprimir etiquetas de palet (controlador) 2ª parte

En cada página A4 van expuestas dos etiquetas.

Primeramente se puede observar a través de *while* que mientras la cantidad total sea mayor que 0 se ejecutará lo siguiente:

A través de la variable \$i y \$nombre se irán definiendo los nombres de cada palet en la variable \$nombre1. Después se incrementará la variable \$i en 1.

A. Primera sentencia (if)

- 1) Si la cantidad total es mayor o igual que la cantidad por palet, se generará el primer palet y si el resto de la variable $\$i \% 2 == 0$ se añadirá una página nueva y se imprimirá la primera etiqueta, si no es cero se imprimirá la segunda etiqueta debajo de la primera y en la misma página.
- 2) Se guardará el palet en la base de datos
- 3) Se restará la cantidad del palet a la cantidad total

B. Segunda sentencia

- 1) Si la cantidad total es menor que la cantidad por palet, se generará el último palet y se imprimirá de la misma manera que se ha citado anteriormente.
- 2) Finalmente se genera el pdf llamando a la siguiente función `$pdf->Output();`

Anexo 3. Capturas de pantalla

A continuación se muestran algunas pantallas representativas de la aplicación:

Pantalla (escritorio-tablet) Menú principal

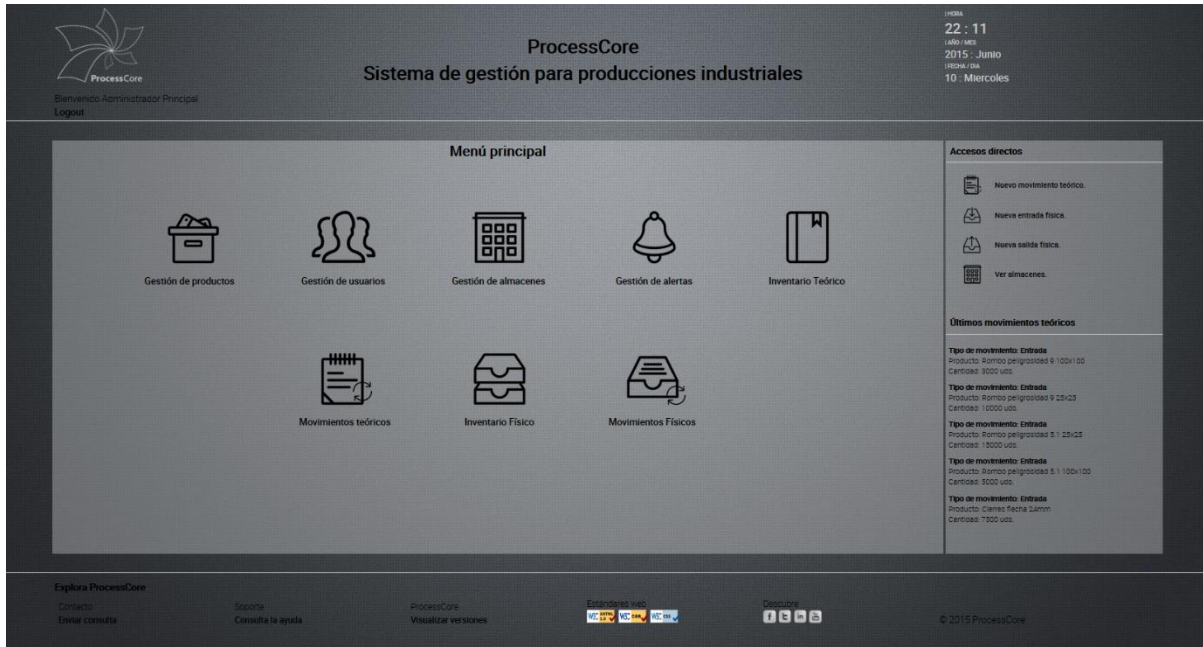


Figura 54: Pantalla menú principal

Pantalla (escritorio-tablet) Nuevo producto

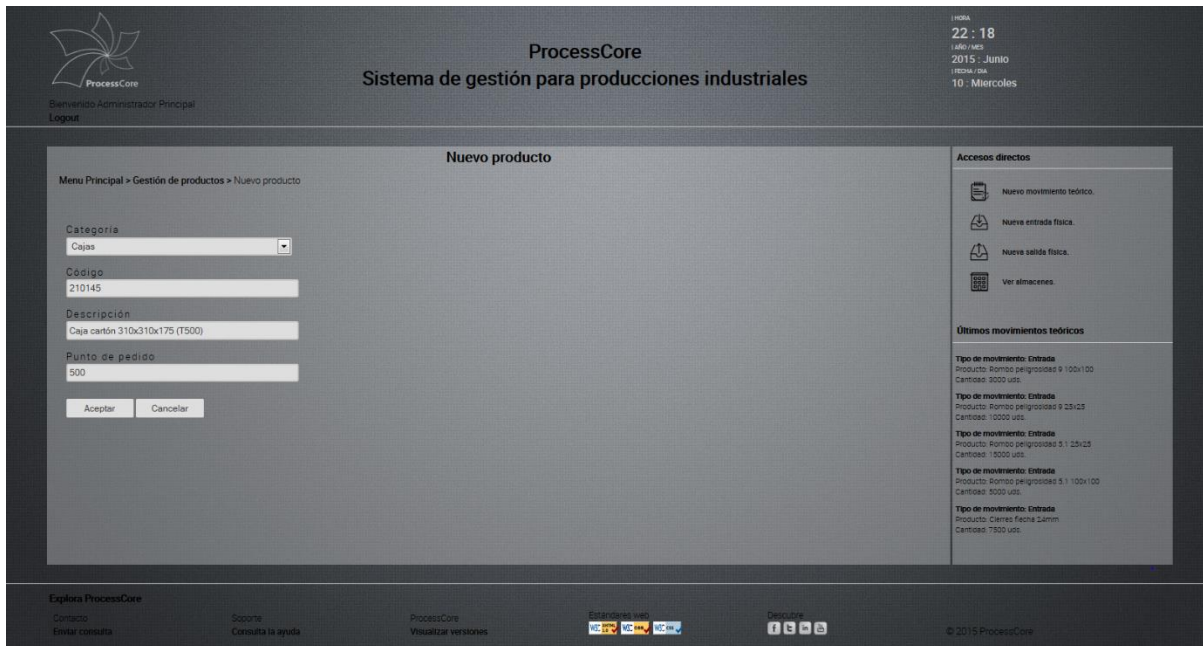


Figura 55: Pantalla nuevo producto

Pantalla (escritorio-tablet) Inventario teórico

ProcessCore
Sistema de gestión para producciones industriales

BIENVENIDO ADMINISTRADOR PRINCIPAL
Logout

INVENTARIO TEÓRICO

Visualizar: Seleccionar Actualizar

Buscar: búsqueda por código o producto Aceptar

CATEGORÍA	CÓDIGO	PRODUCTO	STOCK TOTAL	STOCK DISPONIBLE	STOCK CURSO	STOCK FISICO
Materias primas	611413	DcoNa 55	0	-	-	-
Materias primas	611421	DcoNa 60	0	-	-	-
Materias primas	110241	Atoc grano 90	21000	-	-	21000
Materias primas	611364	Atoc poko 90	0	-	-	-
Materias primas	105031	Sulfato de cobre	3000	-	-	3000
Materias primas	110221	DcoNa 55 Fino	0	-	-	-
Bolsas	104006	Acido Borico	3000	-	-	3000
Bolsas	104092	Acido cianúrico granulado	3000	-	-	3000
Bolsas	103092	Carbonato sódico	5000	-	-	25000
Materias primas	104009	Sulfato de aluminio	2000	-	-	2000
Complementos	210124	Cierres flecha 17mm	20000	-	-	20000
Complementos	210125	Cierres flecha 24mm	7500	-	-	7500
Complementos	210121	Dosificador 100ml	0	-	-	-
Complementos	210122	Dosificador 25ml	0	-	-	-
Palets	310085	Palet 100x100	0	-	-	-
Palets	310086	Palet 80x120 Europalet	0	-	-	-
Palets	310087	Palet 100x120 CP1	0	-	-	-
Palets	310154	Palet 110x110 CP3	0	-	-	-

Accesos directos:
 - Nuevo movimiento teórico.
 - Nueva entrada física.
 - Nueva salida física.
 - Ver almacenes.

Últimos movimientos teóricos:
 Tipo de movimiento: Entrada
 Producto: Rombo perforado 9 100x100
 Cantidad: 3000 uds.
 Tipo de movimiento: Entrada
 Producto: Rombo perforado 9 23x23
 Cantidad: 10000 uds.
 Tipo de movimiento: Entrada
 Producto: Rombo perforado 5 1 100x100
 Cantidad: 3000 uds.
 Tipo de movimiento: Entrada
 Producto: Cierres flecha 24mm
 Cantidad: 7500 uds.

Explora ProcessCore
 Contacto: Enviar consulta | Ayuda: Consulta la ayuda | ProcessCore: Visualizar versiones | Estándares web: NC 12V, NC 48V, NC 60V, NC 80V | Descubre: f, t, g, +

© 2015 ProcessCore

Figura 56: Pantalla inventario teórico

Pantalla (escritorio-tablet) Nuevo movimiento teórico

ProcessCore
Sistema de gestión para producciones industriales

BIENVENIDO ADMINISTRADOR PRINCIPAL
Logout

MOVIMIENTO TEÓRICO

Fecha: 2015-05-25

Referencia: Envases superiores 5kg

Observaciones: Albarán 387562

Aceptar Cancelar

LINEAS

CÓDIGO	PRODUCTO	CANTIDAD	ACCIONES
210142	Cubo 5.9 ltr Superfos 222mm	8000	imprimir etiquetas palet, editar, eliminar
210142T	Tapa cubo 5.9 ltr Superfos 222mm	8000	imprimir etiquetas palet, editar, eliminar

Accesos directos:
 - Nuevo movimiento teórico.
 - Nueva entrada física.
 - Nueva salida física.
 - Ver almacenes.

Últimos movimientos teóricos:
 Tipo de movimiento: Entrada
 Producto: Rombo perforado 9 100x100
 Cantidad: 3000 uds.
 Tipo de movimiento: Entrada
 Producto: Rombo perforado 9 23x23
 Cantidad: 10000 uds.
 Tipo de movimiento: Entrada
 Producto: Rombo perforado 5 1 100x100
 Cantidad: 3000 uds.
 Tipo de movimiento: Entrada
 Producto: Cierres flecha 24mm
 Cantidad: 7500 uds.

Explora ProcessCore
 Contacto: Enviar consulta | Ayuda: Consulta la ayuda | ProcessCore: Visualizar versiones | Estándares web: NC 12V, NC 48V, NC 60V, NC 80V | Descubre: f, t, g, +

© 2015 ProcessCore

Figura 57: Pantalla movimiento teórico

Pantalla (escritorio-tablet) Inventario físico

ProcessCore
Sistema de gestión para producciones industriales

BIENVENIDO ADMINISTRADOR PRINCIPAL
Logout

22 : 13
16/06/2015 - Junio
10 : Miércoles

Menu Principal > Inventario físico

Buscar
Búsqueda por código, producto o pallet

CÓDIGO	PRODUCTO	POSICIÓN ALMACÉN	PALET	CANTIDAD
103092	Carbonato sódico	Sin posicionar	P443	1000
103092	Carbonato sódico	Sin posicionar	P445	1000
103092	Carbonato sódico	AL1E4C1F1	P444	1000
103092	Carbonato sódico	AL1E4C1F1	P444	1000
103092	Carbonato sódico	AL1E4C1F1	P444	1000
103092	Carbonato sódico	AL1E4C1F1	P444	1000
103092	Carbonato sódico	AL1E4C1F1	P444	1000
104006	ACIDO BORICO	TOTAL: 3000		
104006	Acido Borico	Sin posicionar	P471	1000
104006	Acido Borico	Sin posicionar	P472	1000
104006	Acido Borico	Sin posicionar	P473	1000
104009	SULFATO DE ALUMINIO	TOTAL: 2000		
104009	Sulfato de aluminio	Sin posicionar	P451	1000
104009	Sulfato de aluminio	Sin posicionar	P452	1000
104092	ACIDO CIANURICO GRANULADO	TOTAL: 3000		
104092	Acido cianurico granulado	Sin posicionar	P461	1000
104092	Acido cianurico granulado	Sin posicionar	P462	1000
104092	Acido cianurico granulado	Sin posicionar	P463	1000

Accesos directos

- Nuevo movimiento teórico.
- Nueva entrada física.
- Nueva salida física.
- Ver almacenes.

Últimos movimientos teóricos

Tipo de movimiento: Entrada
Producto: Rombo perigrinado 9 100x100
Cantidad: 10000 usd.

Tipo de movimiento: Entrada
Producto: Rombo perigrinado 9 25x25
Cantidad: 10000 usd.

Tipo de movimiento: Entrada
Producto: Rombo perigrinado 5 1 25x25
Cantidad: 18000 usd.

Tipo de movimiento: Entrada
Producto: Rombo perigrinado 5 1 100x100
Cantidad: 8000 usd.

Tipo de movimiento: Entrada
Producto: Cierres Fecha 24mm
Cantidad: 7500 usd.

Explora ProcessCore
Contacto Enviar consulta Soporte Consulta la ayuda ProcessCore Visualizar versiones

© 2015 ProcessCore

Figura 58: Pantalla inventario físico

Pantalla (escritorio-tablet) Gestión de estanterías

ProcessCore
Sistema de gestión para producciones industriales

BIENVENIDO ADMINISTRADOR PRINCIPAL
Logout

22 : 15
16/06/2015 - Junio
10 : Miércoles

Menu Principal > Gestión de almacenes > Gestión de estanterías

ALMACÉN	ESTANTERÍA	CÓDIGO	FILAS	COLUMNAS	ACCIONES
1	Zona Produccion	1	4	5	Imprimir posiciones
1	Zona etiquetado	2	3	4	Editar Eliminar Imprimir posiciones

Accesos directos

- Nuevo movimiento teórico.
- Nueva entrada física.
- Nueva salida física.
- Ver almacenes.

Últimos movimientos teóricos

Tipo de movimiento: Entrada
Producto: Rombo perigrinado 9 100x100
Cantidad: 3000 usd.

Tipo de movimiento: Entrada
Producto: Rombo perigrinado 9 25x25
Cantidad: 10000 usd.

Tipo de movimiento: Entrada
Producto: Rombo perigrinado 5 1 25x25
Cantidad: 18000 usd.

Tipo de movimiento: Entrada
Producto: Rombo perigrinado 5 1 100x100
Cantidad: 8000 usd.

Tipo de movimiento: Entrada
Producto: Cierres Fecha 24mm
Cantidad: 7500 usd.

Explora ProcessCore
Contacto Enviar consulta Soporte Consulta la ayuda ProcessCore Visualizar versiones

© 2015 ProcessCore

Figura 59: Pantalla gestión de estanterías

Etiquetas de palet



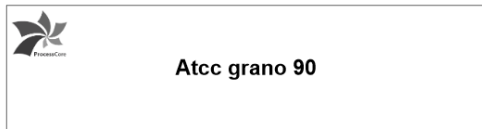
Codigo: 110241

Cantidad inicial: 1000 uds.

Palet: P111



P111



Codigo: 110241

Cantidad inicial: 1000 uds.

Palet: P112



P112

Figura 60: Etiquetas de palet

Etiquetas para estanterías



Zona Produccion

Columna Fila
1 1



AL1E4C1F1

Zona Produccion

Columna Fila
1 2



AL1E4C1F2

Zona Produccion

Columna Fila
1 3



AL1E4C1F3

Zona Produccion

Columna Fila
1 4



AL1E4C1F4

Figura 61: Etiquetas de estantería

Smartphone – Menú principal



Figura 62: Pantalla Smartphone menú principal

Smartphone – Inventario físico



Figura 63: Pantalla Smartphone inventario físico

Smartphone – Movimiento físico – ENTRADA

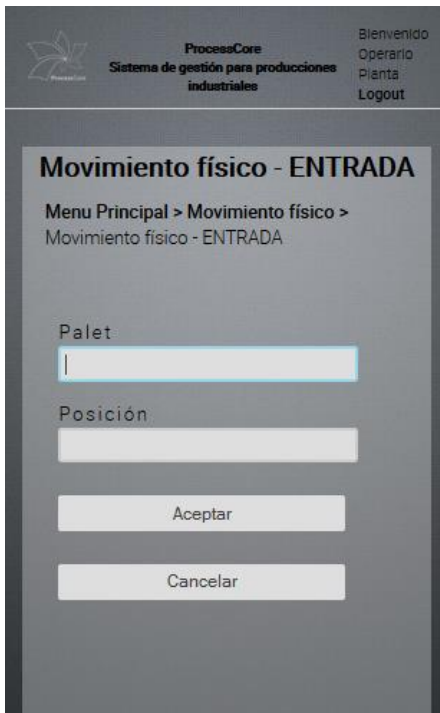


Figura 64: Pantalla Smartphone movimiento entrada

Smartphone – Movimiento físico – SALIDA



Figura 65: Pantalla Smartphone movimiento salida

Anexo 4. Libro de estilo

Logotipo sobre fondo claro



Figura 66: Logo sobre fondo blanco

Color:  #434142

Fuente: Myriad Pro Bold & Regular

Logotipo sobre fondo oscuro



Figura 67: Logo sobre fondo oscuro

Color:  #c3c4c9

Fuente: Myriad Pro Bold & Regular

Web



Fuentes:

Títulos h1, h2, h3, h4: Roboto Regular

Enlaces: Roboto Medium

Texto general y títulos h5: Roboto Light

Sección tiempo: Roboto Medium

Color:  #000000  #CCCCCC

Iconos:

License: <http://www.1001freedownloads.com/free-vector/lynnny-icons-full>



Figura 68: Iconos web

Lynny Icons - Full by [1001FreeDownloads.com](http://www.1001freedownloads.com)

Anexo 5. Bibliografía

MORVILLE, Peter & ROSENFELD, Louis. **Arquitectura de la información para la World Wide Web**. Primera edición. Barcelona: Ediciones Eureka Media, SL. UOC, 2010. Depósito legal: B-8.149-2010.

Eva Patricia Gil, Eva de Lera Tatjer y Antònia Monjo Palau. **Usuarios y sistemas interactivos**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2009. Depósito legal: B-20.845-2011

Jordi Cabot Sagrera, Isabel Guitart Hormigo, Jordi Pradel i Miquel, José Antonio Raya Martos. **Análisis y diseño con patrones**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2010. Depósito legal: B-28.197-2010

Maria Josep Cuenca Ordinyana, Francesca Nicolau Fuster, Àngels Campos González, Josep Ribera i Condomina, Maria Jesús Marco Galindo, Ramon Segret i Sala. **Competencia comunicativa para profesionales de las TIC**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2010. Depósito legal: B-1.316-2010

Piero Berni Millet, Dídac Gil de la Iglesia. **Diseño de bases de datos**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2010. Depósito legal: B-1.330-2010

Joan Arnedo Moreno, Daniel Riera i Terrén. **Diseño y programación orientada a objetos**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2010. Depósito legal: B-27.192-2010

José Ramón Rodríguez, Pere Mariné Jové. **Gestión de proyectos**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2014. Depósito legal: B-14.722-2014

Jordi Pradel Miquel, Jose Raya Martos. **Ingeniería del software**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2011. Depósito legal: B-4.230-2011

M. Jesús Marco Galindo, Jordi Àlvarez Canal, Josep Vilaplana Pastó. **Programación**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2010. Depósito legal: B-3.154-2010

Carlos Casado Martínez, Guillem Garcia Brustenga, Vicent Moncho Mas, Gemma Subirana Grau. **Programación web**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2010. Depósito legal: B-2.429-2010

Vicent Moncho Mas, Francisco Ortega Belmonte, Jordi Sánchez Cano. **Programación web avanzada**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2012. Depósito legal: B-19.372-2012

Sílvia Pujalte Piñán, Xavier Bonet. **Tratamiento y publicación de imagen y vídeo**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2011. Depósito legal: B-28.327-2011

Antoni Marín Amatller. **Vídeo**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2010. Depósito legal: B-10.057-2010

Roser Beneito Montagut, Jordi Alberich, Albert Corral, David Gómez Fontarills, Alba Ferrer Franquesa. **Diseño gráfico**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2010. Depósito legal: B-8.153-2010

Tona Monjo, Cristina Fàbregas. **Diseño de interfaces multimedia**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2011.

Eva Bruballa, Pere Carreras, César Córcoles, Jordi Lagares, Octavi López, David Mata, Jordi Mompert. **Física para multimedia**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2011. Depósito legal: B-19.863-2011

Pere Cruells, Germán Sáez, Joan Trias. **Matemáticas para multimedia I**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2010. Depósito legal: B-7.928-2010

Pere Cruells, Germán Sáez. **Matemáticas para multimedia II**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2011. Depósito legal: 8.511-2010

Teresa Marín García. **Arte, creatividad y diseño**. Barcelona: Ediciones Eureka Media, SL. FUOC, 2011. Depósito legal: B-1.441-2011

VV.AA. **PHP**. Official web php. [en línea] < Información <http://www.php.net/>>

VV.AA. **Forsdelweb**. Comunidad de desarrollo web en lengua hispana. [en línea] < <http://www.forsdelweb.com/>>

VV.AA. **Stackoverflow**. Comunidad de desarrollo web en lengua inglesa. [en línea] < <http://www.stackoverflow.com/>>

VV.AA. **Ganttter**. Herramienta de desarrollo diagramas ganttter en línea. [en línea] < <http://www.ganttter.com> >

VV.AA. **Moqups**. Herramienta de desarrollo wireframes en línea. [en línea] < <http://www.moqups.com>>

VV.AA. **FPDF Library**. Librería para crear pdfs. [en línea] < <http://www.fpdf.org/> >

VV.AA. **PHPMailer**. Librería para el envío de emails. [en línea] < <http://phpmailer.worxware.com/> >

VV.AA. **Teclado escáner Demo**. Aplicación Android para permitir al dispositivo móvil disponer de un teclado escáner. [en línea] < <https://play.google.com/store/apps/details?id=com.tecit.android.barcodekbd.demo&hl=es> >

VV.AA. **Wikipedia**. La enciclopedia libre. [en línea] < <https://es.wikipedia.org/wiki/Wikipedia:Portada> >

VV.AA. **1001 Free Downloads**. Web descarga de fotos, iconos, fuentes, vectores... [en línea] < <http://www.1001freedownloads.com/> >

VV.AA. **HD Wallpapers**. Web descarga de wallpapers. [en línea] <www.princeton.edu/communications/services/docs/IAquide2.pdf >