



**Universitat Autònoma  
de Barcelona**

# **SGAR - SISTEMA DE GESTIÓN DE ALMACENES POR RADIOFRECUENCIA**

Memòria del projecte  
d'Enginyeria Tècnica en  
Informàtica de Sistemes

realitzat per

*Valeriano de las Morenas Becerro*

i dirigit per

*Marc Talló Sendra*

Escola Universitària d'Informàtica

Sabadell, *Julio*l de 2009

El/la sotasignant, *Marc Talló Sendra*,  
professor/a de l'Escola Universitària d'Informàtica de la UAB,

**CERTIFICA:**

Que el treball al que correspon la present memòria  
ha estat realitzat sota la seva direcció  
per en *Valeriano de las Morenas Becerro*

I per a que consti firma la present.  
Sabadell, *Juliol* de 2009

-----

Signat: *Marc Talló Sendra*

## Resumen

Con la crisis mundial por la que estamos pasando, las empresas buscan reducir costes lo máximo posible. Esta reducción, supone en alguno de los casos, seguir utilizando procesos de gestión de almacenes rudimentarios, como por ejemplo los listados impresos en papel, los cuales ofrecen poca fiabilidad y rapidez, y en consecuencia un mal servicio al cliente que se traduce en un declive de las ventas.

Por ello es necesario crear un software específico, el cual tenga un coste de desarrollo ínfimo y con el que podamos gestionar perfectamente un almacén. Gracias a este bajo coste de desarrollo del software, podemos invertir más dinero en la implantación de un sistema de radiofrecuencia de mayor prestación, con lo que tenemos un punto más a nuestro favor respecto a otras aplicaciones similares. Además de todos estos aspectos, también se implantará un nuevo sistema de trabajo mediante el uso de etiquetas de bulto y ubicación que constan de un código de barras que será legible mediante los dispositivos móviles.

En resumen, este proyecto consiste en el análisis, diseño e implementación de un aplicativo web para gestionar, de una forma rápida y fiable, los procesos que se realizan sobre diversos materiales en un almacén, todo ello bajo un sistema LAMP (Linux, Apache, MySQL y Php), el cual ofrece un sinfín de prestaciones y una alta configurabilidad a un coste de adquisición nulo.

A continuación, encontraremos el documento donde se refleja todo el desarrollo llevado a cabo para diseñar e implementar la aplicación SGAR.

# Índice de contenidos

<b>1. Introducción</b>	<b>Pág.</b>
1.1 Introducción .....	1
1.2 Descripción del problema.....	1
1.3 Objetivos .....	2
1.4 Estado del Arte .....	4
1.5 Motivaciones personales.....	4
1.6 Organización del documento .....	5
<b>2. Estudio de Viabilidad</b>	
2.1 Introducción .....	6
2.2 Objeto .....	6
2.3 Sistema a realizar .....	8
2.4 Planificación del proyecto .....	12
2.5 Conclusiones .....	15
<b>3. Marco teórico</b>	
3.1 Introducción .....	16
3.2 Lenguajes de programación Web .....	16
3.3 Servidor Web .....	18
3.4 Bases de datos .....	20
3.5 Sistema de transmisión de datos mediante Radiofrecuencia.....	22
<b>4. Análisis</b>	
4.1 Análisis de requerimientos.....	25
4.2 Módulos de la aplicación.....	31
4.3 LOPD .....	37
4.4 Tareas previas al diseño de la aplicación .....	38
<b>5. Diseño</b>	
5.1 Arquitectura.....	39
5.2 Estructura de datos.....	40
5.3 Consideraciones previas.....	46
5.4 Diseño de la interfaz .....	48
5.5 Elementos de la interfaz .....	50

<b>6. Implementación</b>	<b>Pág.</b>
6.1 Conexión a la base de datos.....	68
6.2 Autenticación y sesiones de usuarios en el sistema .....	69
6.3 Estructura base de las páginas.....	70
6.4 Informes .....	72
6.5 Validación de formularios mediante Javascript.....	74
6.6 Mensajes de ayuda mediante Ajax .....	75
6.7 Generación de códigos de barras .....	77
6.8 Hojas de estilo para la impresión .....	78
<b>7. Pruebas de funcionamiento y testeo</b>	
7.1 Pruebas efectuadas .....	79
7.2 Posibles ampliaciones .....	80
<b>8. Conclusiones</b>	
8.1 Conclusiones finales .....	82
<b>9. Bibliografía</b>	
9.1 Relación de libros consultados .....	83
9.2 Relación de páginas Web consultadas .....	83
<b>10. Apéndice</b>	
10.1 Índice de figuras.....	85

## Capítulo 1: Introducción

### 1.1 Presentación

El principal objetivo de este proyecto es el de desarrollar una aplicación para gestionar, de una forma más eficiente, un almacén de dimensiones reducidas. Dicha gestión se divide en 3 partes: entrada de material, salida de material y mantenimiento del almacén.

Actualmente, la gran mayoría de empresas, utiliza aplicaciones informáticas para la gestión de su almacén, pero estas no soportan trabajar con radiofrecuencia, lo que implica que los operarios tengan que trabajar con otros soportes, como pueden ser simples hojas de papel. Esto puede provocar errores en los diferentes procedimientos que se realizan, ya que al utilizar este tipo de soporte, al operario no se le impone ningún procedimiento estándar a seguir.

Con la aplicación SGAR, esto quedara solventado, ya que se especificaran una serie de procedimientos estándar, que si no se siguen, no se podrá acabar el trabajo asignado.

La principal filosofía de esta aplicación es el de etiquetar todos los artículos con códigos de barras y utilizar unos dispositivos portátiles equipados con lectores de códigos de barras que nos aseguren de que realmente estamos tratando con el material correcto. Gracias a este método de trabajo, se ganara en rapidez y fiabilidad, dos aspectos muy importantes para una empresa hoy en día.

Todo esto se realizara mediante una aplicación Web, lo que implica que tengamos todos los datos guardados de una forma segura en una base de datos y que podamos gestionar toda esta información de forma clara, ordenada y sencilla.

### 1.2 Descripción del problema

#### 1.2.1 Estado Actual

Los procedimientos que se llevan a cabo actualmente se realizan de la siguiente manera: el encargado de almacén extrae una serie de listados donde se especifica el material que entra (en el caso de entrada de material), el que sale (preparación de pedidos) y la relación de materiales a inventariar (mantenimiento almacén). Estos listados se imprimen en papel y se entregan a los diferentes operarios. Los procedimientos son distintos según el tipo de tarea que se realice, tal y como se puede ver a continuación:

#### Entrada de material

El operario, con su respectivo listado de material que entra, se desplaza a la zona donde se deposita el material que entra en el almacén (lugar que denominaremos playa de entrada) y va seleccionando materiales y comprobando que estén en el listado. Si el material se encuentra en este listado, se cuenta para comprobar que la cantidad sea la correcta y se ubica en un lugar del almacén que este libre. Una vez realizado esto, se devuelve este listado a otro operario para que introduzca la cantidad y la ubicación de dicho material y así quede registrado en el sistema.

### Salida de material

El operario, con su respectivo listado de material que se tiene que preparar para su expedición, selecciona un número de pedido y lo comienza a preparar. En dicha preparación, por cada uno de los materiales, se desplaza hasta la ubicación indicada en el listado, coge el material y lo coloca en un/a caja/palet. Una vez preparada la mercancía se le pega una etiqueta que indica los datos del cliente, la dirección de envío y el número de pedido para más tarde depositarlo en la playa de salidas (lugar donde se prepara todos los pedidos para su posterior transporte). Por último el operario responsable de la preparación del pedido, le devuelve el listado a otro operario para que valide la preparación y pueda sacar el albarán de cliente.

### Mantenimiento de Almacén

El operario, con su respectivo listado de material a inventariar, se desplaza a las diferentes ubicaciones indicadas en el listado y recuenta el material que hay en dicha ubicación. Una vez recontado, compara el stock teórico del listado con el stock real e indica la diferencia. Si el operario encuentra cualquier incidencia (como por ejemplo, la ubicación no contiene el material indicado en el listado, no hay nada en dicha ubicación, etc.) lo especifica en el listado para que pueda ser tratada.

Cuando acaba, le devuelve el listado a otro operario para que arregle los stocks en el sistema y trate las posibles incidencias.

## **1.2.2 Diagnóstico**

Como podemos ver, por cada uno de los procedimientos que se realizan, se necesitan dos operarios, uno que trata con el material y otro para que refleje dichas acciones en el sistema informático. Esto implica que para un único procedimiento se tengan que aplicar el doble de tiempo y de recursos. Además no podemos asegurar que el material tratado sea correcto porque no hay ningún mecanismo que pueda verificarlo, únicamente tenemos la confianza en que el operario que lo haya preparado bien.

Esto claramente supone que el sistema es poco fiable, ineficiente, lento, costoso, en definitiva, una relación de malos adjetivos que implican una mala imagen para la empresa.

## **1.3 Objetivos**

El principal objetivo de SGAR es principalmente que: se solventen las deficiencias que existen y que se apliquen nuevos procedimientos más eficaces además de optimizar el sistema de trabajo para que con ello reduzcamos considerablemente la cantidad de reclamaciones de clientes y así podamos ofrecer un servicio de mejor calidad.

Con estas premisas, implantaremos un sistema basado en radiofrecuencia, una gestión de material mediante códigos de barras y el concepto de “bulto” como unidad de recuento.

Para ello se deberán realizar las siguientes tareas:

- **Definir nuevos procedimientos de trabajo**, utilizando el nuevo concepto de código de bulto + código de barras para cada uno de los artículos existentes en el almacén.
- **Simplificar el sistema de trabajo** de tal forma que un único operario realice todos los procedimientos, evitando así que otro operario tenga que realizar las tareas de introducción de datos en el sistema informático.
- **Asignar tareas a operarios remotamente**, permitiendo el traspaso a otro operario de dicha tarea si el operario inicial no puede acabarla.
- **Asignar prioridades** a las tareas en la asignación a los diferentes operarios.
- Realizar un **estudio de cobertura inalámbrica** lo mas eficiente posible, para utilizar los mínimos dispositivos obteniendo la máxima cobertura.
- Realizar una **correcta elección de los dispositivos inalámbricos**, dispositivos móviles e impresoras de etiquetas de acuerdo con las necesidades que se tengan, optimizando de esta manera el coste total.
- **Incluir nuevas opciones en las tareas de mantenimiento de almacén**, como pueden ser, mover bulto de ubicación, recontar bulto, desmontar bulto e inventario.
- **Registrar toda la información** importante en el tratamiento de un bulto.
- **Gestionar parámetros básicos del sistema** (alta/baja operarios, alta/baja artículos, alta/baja de ubicaciones, etc.)

Y como objetivos específicos tenemos:

- **Diseñar dos interfaces graficas diferentes**, una que se utilizara en los dispositivos móviles y la otra en PC's comunes.
- **Crear una aplicación sencilla y amigable**, con menús de acceso rápido para facilitar la navegación, y con esto conseguir que el aprendizaje de los operarios se reduzca y así el impacto de la implantación sea mínimo.
- **Alta fiabilidad en la preparación de pedidos**, ya que el sistema solicita código de bulto a preparar, lo que nos asegura de que estamos tratando con el material correcto.
- **Optimización de la aplicación** para no sobrecargar los dispositivos móviles, PC's y servidores.
- **Describir correctamente los errores generados** por el programa para que el operario sepa que debe hacer a continuación. Se especificaran las posibles causas que han generado dicho error y su correspondiente solución.
- **Aprender a gestionar un sistema** que esta implementado bajo las tecnologías PHP, MySql, Apache, Ajax, etc. sobre un servidor con sistema operativo Linux.
- **Aplicar todos los conocimientos** adquiridos a lo largo de la carrera.

## 1.4 Estado del Arte

Actualmente existen varias aplicaciones que trabajan con radiofrecuencia, pero estas tienen un coste muy elevado que no todas las empresas pueden cubrir. Además si le sumamos que al implantar este tipo de sistema, debemos de formar a todo los usuarios para que se adapten lo más rápido posible, el coste se eleva aun mas.

Programas como UBICA, IBS o SISLOG, reconocidos mundialmente, son programas muy eficaces y capaces de trabajar con almacenes de grandes dimensiones, pero incorporan muchas funcionalidades que en un almacén de menor envergadura no se le sacaría todo el rendimiento que ofrecen, además del gasto que supone implantar un sistema de estas características. También cabe destacar que cuando se compra un tipo de programa como este, te comprometes al mantenimiento de este con sus actualizaciones, posibles errores, mantenimiento de hardware, etc., lo que supone un gasto excesivo y una dependencia total de la empresa que nos ofrece el software.

Por ello la necesidad de crear un programa más económico y con prestaciones adecuadas a estos almacenes de menor envergadura, sin que aspectos como la fiabilidad o la estabilidad del sistema se vean perjudicados.

## 1.5 Motivaciones personales

La creación de esta aplicación es debido a que, en la empresa donde yo trabajo, existen todas estas deficiencias que implican que no se ofrezca un servicio de calidad y que la reputación de la empresa no sea buena. Debido a esta problemática existente, veo muy interesante realizar un estudio sobre en el campo de la radiofrecuencia y los dispositivos móviles y así poder diseñar una buena aplicación para poder mejorar el servicio que se presta.

A parte de todo esto, siempre me ha gustado mucho el tema de las páginas Web y la verdad es que nunca me había puesto en serio a diseñar una, exceptuando en las practicas de algunas asignaturas de la carrera, y que mejor oportunidad que en el proyecto de fin de carrera. Además veo como valor añadido el realizar dicha aplicación sobre tecnologías que desconozco, como php, JavaScript, Flash, css, Apache, Ajax y Mysql sobre un sistema operativo que empecé a utilizar hace poco tiempo, Linux.

La selección de estas tecnologías viene vinculada a la elección del sistema operativo que he escogido. He podido comprobar personalmente las ventajas y desventajas que ofrece dicho sistema operativo y para mi parecer supera con creces la que me ofrecen otros sistemas. Además de ser un sistema muy estable, más seguro (de momento no se han diseñado virus), mas rápido, lo que supone unos aspecto muy atractivos, es completamente gratuito lo que reduce mucho el coste total del proyecto.

Estas son los principales aspectos que me han impulsado a utilizar estas tecnologías sobre este sistema operativo, y si además le añadimos que llevo tiempo haciendo cursos relacionados con esto, que mejor momento para practicar lo aprendido.

## 1.6 Organización del documento

En este primer capítulo se ha dado una introducción del proyecto, además de indicar otros puntos como la descripción del problema, con su situación actual y el diagnóstico a dicha situación, los objetivos principales y específicos que se plantean y por último las motivaciones que me han llevado a la realización de este proyecto.

En el segundo capítulo encontraremos toda la planificación y desarrollo del proyecto donde encontraremos el estudio de viabilidad, el cual nos sirve para saber si el proyecto es viable o no.

En el tercer capítulo nos encontraremos nociones básicas de los lenguajes y tecnologías utilizadas en el desarrollo del proyecto, para obtener unos mínimos conocimientos de cómo funciona.

El análisis de la aplicación lo encontraremos en el cuarto capítulo, donde tenemos los requerimientos funcionales, no funcionales y donde explicaremos con detalle cada uno de los módulos del aplicativo.

En el quinto capítulo veremos todo lo referente al diseño de las dos interfaces que se utilizarán y las características de cada una de ellas. Además obtendremos una visión global de las funcionalidades que ofrece la aplicación.

El sexto y séptimo capítulo trataremos conceptos sobre el código fuente implementado y las pruebas de testeo que se han realizado sobre este código para comprobar la estabilidad, fiabilidad y consistencia del sistema.

El apartado de las conclusiones finales lo encontraremos en el octavo capítulo donde se especifica lo que se aporta con la aplicación ya finalizada, si se han cumplido todos los objetivos, los problemas que se han ocasionado, entre otros aspectos.

Por último, encontraremos una lista de referencias de la bibliografía utilizada, donde indicaremos los libros y webs consultadas, para llevar a cabo el proyecto además de un apéndice donde hay un índice de figuras para poder localizar las imágenes que hay en este documento más rápidamente.

## Capítulo 2: Estudio de Viabilidad

### 2.1 Introducción

#### 2.1.1 Breve presentación del proyecto

El proyecto consiste en diseñar una aplicación Web para gestionar de una forma más eficiente y rápida un almacén, optimizando así los procedimientos que se siguen actualmente mediante la implantación de un sistema de radiofrecuencia, la utilización de códigos de barras para cada artículo existente en el almacén y el uso de dispositivos móviles que ayuden a los operarios en dichos procedimientos.

#### 2.1.2 Historia del proyecto

Este proyecto nace debido a la cantidad de problemas que se generan cuando se utiliza un soporte físico como es el papel. Al utilizar este soporte no se garantiza la fiabilidad de que el usuario seleccione el producto correcto ya que no existe ningún tipo de control sobre este. Debido a esta problemática se propone implantar una herramienta informática que trabaje sobre un sistema de radiofrecuencia para aumentar la fiabilidad y asegurar un control sobre el operario, además de garantizar una mayor rapidez en la ejecución de los diferentes procedimientos.

### 2.2 Objeto

#### 2.2.1 Descripción del estado actual

Actualmente existe un sistema de trabajo y un soporte físico no muy eficaz y que generan una serie de problemas que no garantizan la buena gestión del almacén. La finalidad de este proyecto es de reconducir esta situación mediante el diseño de nuevos procedimientos más eficientes, la creación de una herramienta Web que organice toda la información en una base de datos y la implantación de la radiofrecuencia para controlar la actividad del almacén.

#### 2.2.2 Perfil del cliente – usuario

Actualmente existen 3 perfiles de usuario, como son el administrativo (se encarga de introducir datos de material de entrada), el encargado del almacén (encargado de asignar tareas a operarios) y el operario (ejecuta las tareas que se le asignan).

Con el nuevo sistema, el operario que gestione el material podrá también introducir los datos en el sistema informático mediante el dispositivo móvil que llevará consigo, con lo que podemos eliminar el perfil del administrativo.

En resumen, en el nuevo sistema existirán tres tipos de perfil:

- **Administrador:** llevara a cabo el mantenimiento de la aplicación así como las diferentes ampliaciones que se realicen sobre el código fuente.
- **Encargado de almacén:** llevara a cabo la asignación de tareas a los diferentes operarios, además del correcto mantenimiento del almacén.
- **Operario:** llevara a cabo la gestión de material a medida que se le asignen diferentes tareas.

Cabe destacar que se realizara una formación especializada a cada perfil de usuario para que se puedan familiarizar con el nuevo sistema y así el impacto de implantación sea lo menor posible.

### 2.2.3 Objetivos

A continuación se indican los principales objetivos del proyecto:

- **Definir nuevos procedimientos de trabajo**, utilizando el nuevo concepto de código de bulto + código de barras para cada uno de los artículos existentes en el almacén.
- **Simplificar el sistema de trabajo** de tal forma que un único operario realice todos los procedimientos, evitando así que otro operario tenga que realizar las tareas de introducción de datos en el sistema informático.
- **Asignar tareas a operarios remotamente**, permitiendo el traspaso a otro operario de dicha tarea si el operario inicial no puede acabarla.
- **Asignar prioridades** a las tareas en la asignación a los diferentes operarios.
- **Alta fiabilidad en la preparación de pedidos**, ya que el sistema solicita código de bulto a preparar, lo que nos asegura de que estamos tratando con el material correcto.
- **Registrar toda la información** importante en el tratamiento de un bulto.
- **Gestionar parámetros básicos del sistema** (alta/baja operarios, alta/baja de ubicaciones, generar e imprimir bultos, etc.)

### 2.2.4 Fuentes de información

Para poder recoger toda la información necesaria se realizara consultas a los siguientes usuarios:

- **Encargado de almacén:** nos informara sobre cómo se realiza la asignación de tareas a los diferentes operarios, como se organiza el almacén, los procedimientos que se llevan a cabo en las diferentes áreas y las posibles mejoras que considere oportunas.
- **Administrativo:** nos informara sobre cómo y cuándo introduce la información en el sistema sobre las tareas llevadas a cabo, como está estructurado el programa que utiliza y como se puede optimizar esta entrada de datos para que sea lo más fácil posible para los operarios.
- **Operario:** nos informara de cómo sigue los diferentes procedimientos de gestión de material, donde y como se producen los fallos o problemas y como cree que debería ser la interface del programa para que sea lo mas personalizada posible.

## 2.3 Sistema a realizar

### 2.3.1 Descripción

#### Requerimientos funcionales:

- Diseñar una aplicación Web capaz de trabajar sobre un sistema de radiofrecuencia que gestione los procedimientos que se producen en un almacén.
- Diseñar una base de datos capaz de almacenar todos los datos necesarios y que garantice un rápido acceso y un orden de estos.
- Realizar un buen estudio de cobertura para un buen funcionamiento.

#### Requerimientos no funcionales:

- La aplicación dispondrá de una interface intuitiva para que sea fácil de utilizar por parte de los usuarios.
- El sistema ha de ser lo más estable posible, para evitar posibles problemas de bloqueos del sistema. Para ello, se realizaran todas las pruebas necesarias.
- El sistema quedara abierto para posibles ampliaciones de dispositivos móviles o PC's de gestión, sin que repercuta en el comportamiento de este.
- Se diseñara de tal forma que se minimice lo máximo posible el costo total del proyecto. Para ello se utilizaran herramientas y lenguajes gratuitos.
- El sistema registrara en cada momento todas las operaciones realizadas sobre los artículos, para en caso necesario, extraer información y resolver posibles incidencias.

### 2.3.2 Modelo de desarrollo

El desarrollo de la aplicación seguirá las siguientes fases:

- Definición de requerimientos funcionales y no funcionales.
- Diseño de las interfaces y la base de datos.
- Codificación de los diferentes módulos, cada uno de ellos se realizara tras el anterior y antes que el siguiente modulo.
- Estudio de cobertura.
- Implementación del programa y del sistema de radiofrecuencia.
- Corrección de posibles errores.
- Codificación de las modificaciones en los correspondientes módulos.
- Formación de usuarios y consolidación del sistema.

Siguiendo estas fases y teniendo en cuenta la estructura del programa en módulos independientes, se seguirá un paradigma de desarrollo **lineal secuencial** (ciclo de vida clásico).

### 2.3.3 Recursos

#### 2.3.3.1 Recursos Software

- Sistema Operativo servidor Ubuntu 8.04 LTS.
- Sistema Operativo pc's Windows Xp Professional SP2.
- Sistema Operativo dispositivos móviles Windows Mobile 2003 o superior.
- Navegador Web Mozilla Firefox 2 o superior.
- Servidor Web Apache.
- Base de datos MySql.

#### 2.3.3.2 Recursos Hardware

- 1 Servidor con la siguiente configuración:
  - Procesador Intel Dual Core 2 o superior.
  - 4 Gb de memoria RAM.
  - Sistema discos duros Raid 1 320 Gb o superior.
  - Tarjeta de red Ethernet Gigabyte.
  - Sistema de alimentación ininterrumpida (SAI).
- 1 PC (según necesidad) para gestión con la siguiente configuración:
  - Procesador Intel Pentium 4 o superior.
  - 1 Gb de memoria RAM.
  - Disco duro de 80 Gb o superior.
  - Tarjeta de red Ethernet 100/1000.
- 1 Dispositivo móvil para cada operario modelo Motorola Symbol MC9090S
- Sistema de radiofrecuencia formado por:
  - El número necesario de antenas (según estudio cobertura) modelo AP-5131 Motorola.
  - 1 dispositivo WS2000 Wireless Switch para interconectar las antenas.
- 1 Impresora de códigos de barras modelo Zebra ZM400:
  - Método de impresión por transferencia térmica.
  - Auto-calibración.
  - Impresión sobre varios tamaños de etiquetas.

### 2.3.3.2 Recursos Humanos

- 1 Analista de sistemas
- 1 Diseñador Web
- 1 Programador
- 1 Técnico de sistemas
- 1 Becario

### 2.3.4 Análisis coste – beneficio

Teniendo en cuenta la utilización de herramientas y lenguajes gratuitos, a continuación se detalla el coste total del proyecto. Además del coste de la aplicación, se incluye el del hardware necesario para la implantación del sistema de radiofrecuencia.

Detalle del coste de diseño de la aplicación:

Tarea	Recurso	Horas Trabajo	Coste
Análisis	Analista (20€/h)	42 h	840 €
Diseño de las interfaces	Diseñador (20€/h)	26 h	520 €
Programación módulos y BBDD	Programador (12€/h)	176 h	2112 €
Instalación del sistema de Radiofrecuencia	Técnico (12€/h)	44 h	528 €
Elaboración documentación	Becario (6€/h)	24 h	144 €
<b>SUBTOTAL</b>			<b>4.144 €</b>

Detalle del coste de hardware utilizado:

Material	Cantidad	Coste Unitario	Coste Total
Amortización PC programador	1	$(900/24)*2$	75 €
Amortización PC diseñador	1	$(1200/24)*2$	100 €
Amortización PC becario/analista	1	$(900/24)*2$	75 €
Amortización Linux Debian	1	0	0 €
Amortización Open Office	1	0	0 €
Amortización Windows XP Prof.	1	100 €	100 €
Amortización MySql - Apache	1	0 €	0 €

Servidor	1	1195 €	1195 €
PC	1	585 €	585 €
Dispositivo Motorola MC9090S	5	1232 €	6160 €
Antena AP-5131 Motorola	5	426 €	2130 €
WS2000 Wireless Switch	1	590 €	590 €
Impresora C.B. Zebra Zm400	1	652 €	652 €
Material montaje Radiofrecuencia	1	250 €	250 €
<b>SUBTOTAL</b>			<b>11.912 €</b>

Suma de los costes de software, hardware y margen de beneficio:

Coste de desarrollo de la aplicación	4.144 €
Coste de hardware utilizado en desarrollo aplic. + Radiofrecuencia	11.912 €
<b>SUBTOTAL</b>	<b>16.056 €</b>
Margen beneficio 25 %	3.939 €
<b>TOTAL</b>	<b>20.070 €</b>

### 2.3.5 Detalle del presupuesto

<p>Aplicación de gestión de almacenes por Radiofrecuencia (<b>SGAR</b>)</p> <ul style="list-style-type: none"> <li>• Gestión de procedimientos mediante dispositivos móviles.</li> <li>• Asignación de tareas remotamente y mediante cola de prioridad.</li> <li>• Control total sobre actividad de operarios.</li> <li>• Actualización de procedimientos para aumentar rendimiento y efectividad.</li> <li>• Formación a usuarios.</li> </ul>	
<p>Material necesario para la implantación del sistema de Radiofrecuencia</p> <ul style="list-style-type: none"> <li>• 5 ud dispositivo movil Motorola MC9090S.</li> <li>• 5 ud antena AP-5131 Motorola.</li> <li>• 1 ud WS2000 Wireless Switch.</li> <li>• 1 ud impresora C.B. Zebra Zm400.</li> </ul>	
<b>TOTAL</b>	<b>20.070 €</b>

Los precios no incluyen el 16% de IVA

Los precios tienen una vigencia de 60 días.

### 2.3.6 Evaluación de riesgos

Al ser un proyecto primerizo y de no grandes dimensiones, los principales riesgos que pueden surgir durante el desarrollo de este pueden ser:

- El tiempo estimado en que se va a realizar el proyecto no corresponde con el tiempo real en que se ha finalizado. Esto puede ser debido a una mala programación de las fases del proyecto que conllevan un retraso en el desarrollo del sistema. Por tanto, se tendría que dar más margen en las fechas programadas para resolver posibles problemas que puedan surgir.
- Una vez comience la programación de los módulos sobre un lenguaje no conocido con anterioridad, nos damos cuenta de que este lenguaje seleccionado no proporciona las herramientas necesarias para la realización de este proyecto. Para ello nos aseguraremos de que dicho lenguaje implementa toda la funcionalidad que necesitamos.
- Ya que el programa está basado en una aplicación Web, es necesario contemplar la compatibilidad entre los navegadores que se utilizarán.
- Otro aspecto importante son los problemas que puedan surgir en la comunicación por radiofrecuencia entre los dispositivos y el servidor. Para ello se deberá estudiar minuciosamente como estructurar esta red para minimizar lo máximo posible estos problemas de transmisión de datos.
- Por último, se ha de contemplar la posibilidad de que exista una ampliación del sistema y se aumente el número de terminales móviles conectados, por tanto el sistema tiene que estar preparado para poder acoger más dispositivos sin ningún tipo de problema.

### 2.3.7 Alternativas

Actualmente existen varios programas similares a SGAR, pero todos ellos utilizan unos lenguajes de programación bajo licencia, lo que supone un incremento considerable en el precio final del programa.

Esto también supone una total dependencia de la empresa responsable del programa a la hora de realizar cualquier modificación o ampliación de este, y en consecuencia el importe sobre dicha modificación que la empresa cree oportuno.

Además, todas estas empresas te obligan a firmar un contrato de mantenimiento, debido a que utilizan hardware específico para su programa.

El conjunto de estos parámetros a tener en cuenta a la hora de comprar un tipo de software de estas características, crea la necesidad de desarrollar un software que posea las funcionalidades necesarias para desarrollar el mismo trabajo que los demás programas pero a un precio de mercado más asequible.

## 2.4 Planificación del proyecto

### 2.4.1 Duración estimada de las fases del proyecto

A continuación se especifica las diferentes fases en que se ha dividido el proyecto y el total de horas necesarias para desarrollar cada una de estas:

Nº	Descripción de la fase	Duración (h)
1	Estudio de Viabilidad	8
2	Análisis	12
3	Entrevista con los diferentes usuarios	16
4	Aprendizaje de los lenguajes no conocidos	40
5	Diseño de la interface grafica	8
6	Diseño e implementación de la BBDD	8
7	Programación del modulo de entrada de material	24
8	Programación del modulo de salida de material	24
9	Programación del modulo de mantenimiento almacén	24
10	Programación del modulo de informes	24
11	Programación del modulo de parámetros del sistema	24
12	Programación de las interfaces web	16
13	Implantación de módulos sobre las interfaces web	16
14	Pruebas de testeo	8
15	Estudio de cobertura*	16
16	Instalación del sistema	32
17	Pruebas sobre el sistema montado en red	8
18	Corrección de posibles errores	8
19	Elaboración de documentación	24
20	Curso de formación a usuarios	16
	<b>TOTAL</b>	<b>356 h</b>

*\*Nota: el estudio de cobertura depende de las dimensiones de cada almacén. El número de horas indicado es estimado. Dicho estudio lo llevara a cabo la empresa que realizara la instalación.*



## 2.5 Conclusiones

Al implantar este sistema se obtienen las siguientes mejoras:

- **Optimización** de procedimientos de manipulación de materiales.
- Completa **organización** del almacén mediante la utilización de códigos de barras para producto y para ubicaciones.
- Gran **simplicidad** a la hora de realizar inventarios de artículos.
- **Efectividad** en la recepción/salida de materiales y mantenimiento del almacén debido al control efectuado por el programa.
- **Asignación remota** de tareas a usuarios.
- **Organización** y **control** sobre los datos almacenados, ya que se encuentran en una base de datos alojada en un servidor, con lo que accedemos de forma inmediata.
- **Resolución inmediata** de posibles incidencias, ya que queda registrado todas las acciones realizadas por un operario en cada manipulación de material.
- **Formación** sobre un método de trabajo más efectivo, rápido y fiable.

## Capítulo 3: Marco Teórico

### 3.1 Introducción

A continuación se exponen una serie de conceptos teóricos sobre los cuales se desarrolla este proyecto, los cuales hemos de tener una pequeña noción para poder entender de una forma más concisa, algunos puntos de la memoria.

### 3.2 Lenguajes de programación Web

Actualmente existen diferentes lenguajes de programación para desarrollar páginas Web, estos han ido surgiendo debido a las tendencias y necesidades de las distintas plataformas que existen.

A continuación se exponen la relación de lenguajes utilizados en el desarrollo del proyecto, de los cuales se muestran sus ventajas y desventajas.

#### 3.2.1 Lenguaje HTML

Desde la expansión de Internet, han sido muchísimos los sitios Web publicados gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de *HyperText Markup Language*, en español *Lenguaje de Marcas Hipertextuales*). Fue desarrollado por el *World Wide Web Consortium* (W3C). Por convención, los archivos de formato HTML usan la extensión *.htm* o *.html*.

#### Ventajas

- Se programa mediante marcas de hipertexto, lo que permite Hipervínculos.
- El código se presenta de forma clara y estructurada.
- Crea archivos de pequeño tamaño, lo que permite una carga rápida.
- Lenguaje de aprendizaje fácil.
- Lo admiten todos los exploradores.

#### Desventajas

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- El diseño es más lento.
- Las etiquetas son muy limitadas.

#### 3.2.2 Lenguaje JavaScript

Se trata de un lenguaje interpretado, con lo que no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Se utiliza principalmente en páginas Web. Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta

se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Actualmente, todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM (en inglés *Document Object Model*, en español *Modelo de Objetos del Documento*).

### **Ventajas**

- Lenguaje de scripting seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código JavaScript se ejecuta en el cliente.

### **Desventajas**

- Código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio Web, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).

### **3.2.3 Lenguaje PHP**

Es el lenguaje de programación más utilizado actualmente para la creación de sitios web dinámicos. PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor*, (inicialmente se llamó *Personal Home Page*). Surgió en 1995, desarrollado por PHP Group.

Generalmente se ejecuta en el servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. PHP no necesita ser compilado para ejecutarse además de que puede ser incrustado dentro de código HTML. Para su funcionamiento se necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Por convención, los archivos en formato PHP usan la extensión *.php*.

#### **Ventajas:**

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida el lenguaje orientado a objetos (clases y herencia).
- Es un lenguaje multiplataforma: Linux, Windows, etc.
- Capacidad de conexión con la mayoría de los gestores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, etc.
- Capacidad de expandir su potencial utilizando módulos.
- Posee de infinidad de documentación detallada.

- Es de código abierto, lo que rebaja los costes de desarrollo.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

**Desventajas:**

- Se necesita instalar un servidor Web.
- Todo el trabajo lo realiza el servidor y no delega en el cliente. Por lo tanto, puede ser más ineficiente a medida que las solicitudes aumenten en número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación.

También hemos de comentar que existen otros lenguajes alternativos a este, como puede ser ASP.net, pero al tratarse de un lenguaje privativo y con un alto coste de implantación, nos obliga a descartarlo ya que no cumplía con una de las políticas del programa SGAR, un desarrollo e implantación de bajo coste frente al software de la competencia.

**3.3 Servidor Web**

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada). Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

1. Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso.
4. Envía el recurso utilizando la misma conexión por la que recibió petición.
5. Vuelve al segundo punto.

Un servidor web que siga el esquema anterior cumplirá todos los requisitos básicos de los servidores HTTP, aunque sólo podrá servir ficheros estáticos.

Pero el aspecto que más nos interesa del servidor web, es el nivel de soporte que ofrece para servir contenido dinámico. Puesto que la mayor parte del contenido web que se sirve no viene de páginas estáticas, sino que se genera de forma dinámica, y esta tendencia se mueve claramente al alza, el soporte para contenido de tipo dinámico que ofrece un servidor web es uno de los puntos críticos en la elección.

La mayor parte de los servidores web ofrecen soporte para CGI (se debe recordar que los CGI son el método más antiguo y sencillo para generar contenido dinámico). Otros muchos ofrecen soporte para algunos lenguajes de programación (normalmente lenguajes interpretados) como PHP, JSP, ASP, etc.

Es muy recomendable que el servidor web que se utilice proporcione soporte para algunos de estos lenguajes, especialmente PHP, sin tener en cuenta JSP, que normalmente requerirá un software externo para funcionar (como un contenedor de Servlets). Un servidor que ofrece soporte a este tipo de lenguajes es Apache.

A continuación se especifican una serie de ventajas que ofrece un servidor Apache:

- Apache es una tecnología gratuita de código fuente libre. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto.
- Corre en una multitud de plataformas, lo que lo hace prácticamente universal.
- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor. Actualmente existen muchos módulos para Apache que son adaptables a este. Otra cosa importante es que cualquier persona que posea una experiencia en programación de C o Perl puede escribir un módulo para realizar una función determinada.
- Apache trabaja con lenguajes Perl, PHP y otros lenguajes de script, como Java y páginas Jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

Pero también se ha de comentar que existe una alternativa a dicho servidor, esta alternativa es Microsoft IIS. Dicho servidor nos ofrece las siguientes ventajas e inconvenientes:

### **Ventajas**

- Fácil de usar
- ASP preparado en la instalación por defecto
- Soporte ODBC integrado
- Configuración gráfica y en línea de comandos

### **Desventajas**

- Multitud de fallos de seguridad
- La mayoría de funcionalidad extra debe ser comprada separadamente
- Sólo funciona en Windows NT/2000.

Otro punto a tener en cuenta, es el grado de utilización de cada uno de los servidores a la hora de servir páginas web en Internet. Como podemos observar en el siguiente grafico, apache es el servidor web mas utilizado en el mundo. Un motivo, que nos impulsa a escoger dicho servidor.

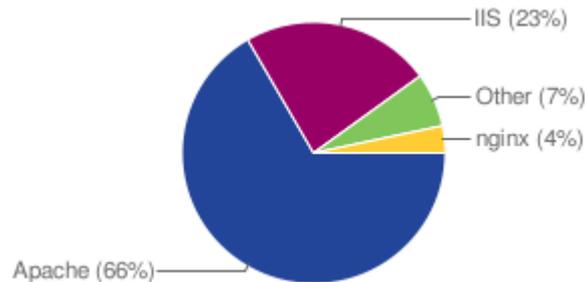


Fig. 1 - Grado de utilización de servidores web en Internet

Dicho grafico se obtenido de un estudio realizado sobre 80 millones de dominios de Internet, dato bastante significativo como para tenerlo en cuenta.

En resumen, teniendo en cuenta las ventajas y desventajas que nos ofrecen ambos servidores, la popularidad de cada uno de ellos, que supone una mayor documentación, y también el lenguaje en que programaremos, está claro que el mas óptimo para llevar a cabo este proyecto es Apache.

### 3.4 Bases de datos

Una base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos. Actualmente, el modelo más utilizado en el diseño de bases de datos es el modelo relacional, el cual se basa en ordenar, de forma lógica, en filas y columnas toda la información almacenada.

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoria.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

Para poder gestionar toda esta información almacenada, se utilizan los *Sistemas de gestión de base de datos* (en inglés *DataBase Management System*) que son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de

definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

A continuación, se especifican una serie de ventajas e inconvenientes de una base de datos contra un sistema de ficheros clásico (papel):

### Ventajas

- Control sobre la redundancia de datos.
- Consistencia de datos.
- Compartir datos.
- Mantenimiento de estándares.
- Mejora en la integridad de datos.
- Mejora en la seguridad frente a usuarios no deseados.
- Mejora en la accesibilidad a los datos.
- Mejora en la productividad.
- Mejora en el mantenimiento.
- Aumento de la concurrencia.
- Mejora en los servicios de copias de seguridad.

### Desventajas

- Complejidad debido a la gran funcionalidad de los SGBD.
- Coste elevado de instalación y mantenimiento.
- Vulnerable a los fallos, debido al estar más centralizado.

En la actualidad podemos encontrar diversos tipos de bases de datos como pueden ser:

- **MySQL:** es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable usar para grandes volúmenes de datos.
- **PostgreSQL y Oracle:** Son sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.
- **Access:** Es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa Access, el cual crea un archivo .mdb.
- **Microsoft SQL Server:** es una base de datos más potente que Access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de informaciones.

Teniendo en cuenta, el lenguaje de programación escogido, el servidor web que se utilizara, está claro que el gestor de BBDD optimo y que ofrece una mayor compatibilidad, es sin duda MySql.

### 3.5 Sistema de transmisión de datos mediante Radiofrecuencia

Los sistemas más desarrollados, en lo que se refiere a gestión de almacenes, se han encontrado siempre con una limitación: el lugar donde se generaba y se solicitaba la información no era un punto fijo, sino móvil, y por tanto la conexión entre dicho punto y el sistema informático se debía hacer utilizando algún tipo de impreso. Esto, que aparentemente puede parecer intrascendente, tiene unas repercusiones importantísimas, ya que impide que la operativa del almacén pueda ser gestionada y optimizada directamente y en tiempo real por un sistema informático. La aparición de los sistemas de radiofrecuencia ha permitido acabar con esta situación.

Para entender qué es un sistema de gestión de almacenes con radiofrecuencia conviene hacer dos aproximaciones: la tecnológica y la logística.

Desde el punto de vista tecnológico, un sistema de gestión de almacenes con radiofrecuencia es un ejemplo característico de integración de tecnologías:

- Red de antenas de transmisión por radiofrecuencia, conectadas al servidor donde se aloja la base de datos mediante un switch específico.
- Dispositivos móviles, equipados con lectores láser de código de barras.
- Implementación de técnicas de optimización operativa en tiempo real.

Estas tecnologías tienen un principal objetivo: la construcción de una herramienta de gestión sencilla y efectiva.

Pero, sin duda la perspectiva más interesante de este tipo de sistemas es la logística. Un sistema de gestión de almacenes por radiofrecuencia, implementa un modelo logístico completamente nuevo y avanzado respecto al tradicional sistema basado en impresos. Dicho sistema se basa en la planificación continua, en el seguimiento de la actividad de los operarios, el inventario en tiempo real y la optimización de la actividad además de la desaparición de las actividades administrativas.

Los principales beneficios obtenidos con la implantación de un sistema de gestión de almacenes con radiofrecuencia son:

- **Reducción de trabajos administrativos**, la eliminación de documentos de trabajo (listados de preparación, reposición, etc.), reduce los trabajos administrativos al desaparecer las tareas de emisión - distribución de los mismos.

- **Disminución de errores por:**
  - Claridad en las órdenes.
  - Control total de cada tarea (chequeo con lector láser de palet - ubicación).
  - Seguimiento pormenorizado de la actividad de cada operario.
  - Reducción de las anomalías y trabajos generados por errores en la ubicación de referencias.
  - Reducción en el coste de supervisión de pedidos.
  - Decremento de las reclamaciones de clientes.
  
- **Aumento de la productividad**, debido a distintos factores: optimización de recorridos, optimización de la asignación de trabajos, El aumento de la productividad en almacenes donde se ha instalado radiofrecuencia se encuentra en torno al 15%.
  
- **Control de productividad y seguimiento de la actividad del almacén en tiempo real** y directamente de los generados por el ordenador sin necesidad de la participación en esta tarea de ningún otro recurso (humano o técnico).
  
- **Disminución del tiempo de respuesta del almacén.** Se elimina la emisión y confirmación de listados de trabajo y además, se posibilita la emisión del albarán o factura en el mismo instante que se termina de preparar el pedido.
  
- **Rotación de existencias.** En cada reposición o salida de palet completo es el ordenador quien elige el palet más antiguo, garantizándose de esta forma la necesaria rotación de las existencias.
  
- **Simplificación de la comprobación de inventario.** utilizando terminales de radiofrecuencia la comprobación de inventario se puede realizar de una forma rápida y sencilla, sin ningún soporte escrito. Para cada ubicación, el terminal indica el tipo de referencia y la cantidad de unidades que el ordenador tiene registradas. Si hay discrepancias la corrección de inventario se realiza desde el mismo terminal.
  
- **Conocimiento del stock de situación del almacén en tiempo real.** Los terminales de radiofrecuencia informan de cada movimiento en el mismo momento en que éste se produce, por tanto los datos contenidos en el ordenador referentes a inventario, ubicación de referencias, situación de pedidos, son un reflejo preciso del estado real del almacén. Esto posibilita reducir stocks, responder rápidamente a cualquier demanda de información (estado de un pedido, por ejemplo) y facilita el mantenimiento del mapa de almacén.

- **Aumento de la capacidad del almacén.** El desfase entre la realización de movimientos y la actualización de ubicaciones en el ordenador provoca una disminución de la capacidad del almacén por considerar ocupados huecos que están vacíos. Al eliminar este desfase vía actualizaciones en tiempo real se consigue aumentar la capacidad disponible del almacén.
- **Mejora del control de incidencias.** El sistema debe controlar automáticamente ciertas incidencias (palet mal ubicado, hueco ocupado, etc.) y debe permitir que los operarios de almacén informen de posibles incidencias. La centralización de todas estas, garantiza en todo momento la fiabilidad de la información.
- **Mejora en la elección de ubicaciones en las entradas.** El conocimiento de la ubicación de las referencias en tiempo real (huecos ocupados y vacíos) permite que para cada entrada se pueda elegir la mejor de las ubicaciones.
- **Mejora del puesto de trabajo.** El operario realiza su trabajo dialogando con el "cerebro del almacén". Esto dignifica el puesto de trabajo y es motivo de satisfacción para el operario.

Todas estas ventajas se pueden sintetizar en tres:

- ✓ Incremento de la productividad.
- ✓ Incremento de la calidad de servicio.
- ✓ Flexibilidad para responder a comportamientos imprevisibles de la demanda y a la futura dimensión de la empresa.

## Capítulo 4: Análisis

### 4.1 Análisis de requerimientos

A continuación se muestran el conjunto de características en las cuales nos basaremos para realizar el análisis de la aplicación. Dichas características son: los tipos de usuarios que utilizarán la aplicación y los dos tipos de requerimientos a analizar, requerimientos funcionales y no funcionales. Para entender un poco mejor que definen cada uno de los requerimientos, realizaremos una pequeña definición de cada uno.

Los requerimientos funcionales son aquellos que definen el comportamiento interno del sistema, es decir, las transformaciones que realiza el programa sobre los datos de entrada para producir los datos de salida. Estas transformaciones pueden ser cálculos, detalles técnicos, manipulación de datos y otras funcionalidades que muestran como los casos de uso son implementados en la realidad.

Los requerimientos no funcionales, en cambio, son aquellos aspectos visibles para el usuario, que no están relacionados de forma directa con la funcionalidad del sistema. Dichos aspectos pueden ser: interfaz del usuario, documentación, rendimiento del hardware, características de ejecución, gestión de errores, fiabilidad, robustez, seguridad, etc.

#### 4.1.1 Usuarios

Al definir los diferentes tipos de usuarios, hemos de tener en cuenta que cada tipo tendrá unos privilegios y restricciones diferentes a los demás grupos, ya que todos comparten la misma información pero acceden a ella de diferente forma. Esto viene definido según el nivel de privilegio, las restricciones y la utilidad que se va a dar a la información, por cada uno de los grupos de usuario.

Existirán 3 tipos de grupos de usuario, que son:

- **Administrador de la aplicación**
  - Será el responsable de la administración de todo el sistema, a nivel de software como de hardware.
  - Deberá poseer altos conocimientos de programación web y gestión de redes, además del tratamiento de incidencias comunes.
- **Encargado o responsable de área**
  - Será el encargado de la asignación de tareas a los diferentes operarios según la cola de trabajo de cada uno de ellos. También se encargará de generar informes para el correcto mantenimiento del almacén.
- **Operario**
  - Son los encargados de gestionar materiales para diversos fines.

### **4.1.2 Requerimientos funcionales**

Para obtener una visión más detallada de los requerimientos funcionales, clasificaremos estos según el grupo de usuario. De esta forma obtendremos cada uno de los requerimientos necesarios para cada tipo de usuario, ya que cada uno tiene unos específicos.

#### **Administradores**

Será el encargado de la administración, mantenimiento y desarrollo de la aplicación, además de la gestión del hardware específico para la radiofrecuencia. No tendrá ninguna restricción sobre el programa y podrá acceder a todos los módulos, además de total acceso a la base de datos para su gestión y acceso al programa de gestión de la red de radiofrecuencia.

Podrá ejercer las siguientes funcionalidades:

- Modificación del código fuente para añadir o modificar funcionalidades.
- Mantenimiento del hardware del sistema de radiofrecuencia.
- Gestión de incidencias con el software o hardware del sistema.
- Acceso total a todas las funcionalidades de la aplicación.

#### **Responsable de área**

Será el responsable de controlar la carga de trabajo de los operarios, así como de la asignación de tareas a los mismos, además del mantener en correcto orden el almacén. Tendrá acceso a todos los módulos, pero no podrá acceder a la base de datos para realizar operaciones de mantenimiento, tan solo podrá consultarla mediante el programa SGAR. No tendrá acceso al programa de control de la red de radiofrecuencia.

Podrá ejercer las siguientes funcionalidades:

- Asignación de tareas a operarios.
- Alta o modificación de parámetros como usuarios, ubicaciones y bultos.
- Acceso a diversos informes del programa.
- Acceso total a todas las funcionalidades de la aplicación, excepto la parte restringida al administrador.

#### **Operario**

Será el encargado de gestionar materiales a partir de la asignación remota. Podrá gestionar materiales de entrada, material de salida y realizar operaciones de mantenimiento sobre materiales del almacén. Tendrá restricciones sobre algunas partes de los módulos.

Podrá ejercer las siguientes funcionalidades:

- Gestión completa de tareas, entrada y salida de material.
- Gestión de almacén mediante modulo de mantenimiento de almacén.

#### 4.1.3 Requerimientos no funcionales

Tal como hemos definido anteriormente, los requerimientos no funcionales son aquellos que no están vinculados con la funcionalidad interna del sistema. En nuestro caso son:

- **Interfaz adecuada a cada dispositivo**

Al trabajar con dispositivos móviles, deberemos adecuar la resolución a dichos dispositivos para una correcta visualización. Además de diseñara con colores suaves para contrarrestar el cansancio visual. Para el ordenador de control se utilizara una resolución de 1024x768 para que la información sea pueda leer de forma rápida ya que si implementamos una resolución mayor, nos tendremos que fijar mas y esto puede agudizar el cansancio visual.

- **Navegación rápida y clara**

El programa debe ser fácil de utilizar, ya que si complicamos mucha la navegación, el impacto de implantación para los operarios seria más costoso. Para ello, en el menú principal de operario tan solo se pondrán los accesos que más se utilizaran, y mediante el uso de iconos facilitaremos la interpretación de los diferentes módulos. Deberemos mostrar también, de una forma clara y explícita, cada uno de los errores que se generen para una correcta interpretación.

- **Estabilidad de la radiofrecuencia**

Para minimizar el tiempo de recepción de material, preparación de pedidos y mantenimiento de materiales, deberemos de implementar una red de radiofrecuencia con un tiempo de respuesta muy bajo, ya que esto nos ayudara a rebajar el tiempo que emplea cada operario en realizar una operación. Para ello deberos realizar un estudio de cobertura optimo, asegurándonos de que tengamos un nivel de señal optimo en cada rincón del almacén.

- **Escalabilidad del sistema**

Debemos de tener en cuenta que la empresa pueda tener un periodo de trabajo mayor de lo habitual y esto nos provoca un incremento en el numero de terminales trabajando con radiofrecuencia y el programa.

- **Desarrollo en módulos**

Al programar en módulos independientes, nos facilita la incorporación o modificación de nuevos parámetros o nuevos módulos, minimizando el coste de implantación, además de que si se produce un fallo en uno de ellos, no afectara a los demás módulos.

- **Máxima velocidad en el acceso a los datos**

El acceso de los operarios al programa se realizara mediante una sesión de cliente remoto contra el servidor para minimizar la transferencia de datos entre el dispositivo y el servidor. Con esto conseguimos simplificar muchísimo la configuración de los dispositivos, ya que con un simple programa de Terminal Server pueden trabajar.

#### **4.1.4 Tecnologías seleccionadas para el desarrollo de la aplicación**

Para el desarrollo de este proyecto, se ha decidido utilizar lenguaje HTML y JavaScript para la programación de la Web, php como lenguaje de acceso a la BBDD, MySql como gestor de dicha BBDD y un servidor web basado en Apache. En lo que se refiere a sistemas operativos, se utilizara un sistema Linux Ubuntu LTS para el servidor principal, Windows XP para el ordenador del responsable de almacén y Windows Mobile para los dispositivos móviles. La decisión de utilizar estas tecnologías se basa en el mero hecho de intentar reducir lo máximo posibles los costes globales del desarrollo. El simple hecho de reducir costes, no supone reducir las prestaciones ya que estas tecnologías ofrecen un sinfín de características que nada tienen que envidiar a las tecnologías de pago.

##### **4.1.4.1 Lenguajes de programación**

Uno de los principales objetivos de este proyecto es, como ya se ha comentado anteriormente, la de desarrollar un software de bajo coste sin perder prestaciones como la estabilidad, velocidad de ejecución, escalabilidad, entre otros. Esto lo conseguimos mediante la implementación de lenguajes como Html en combinación con php, JavaScript y Ajax.

Podemos decir que php es el lenguaje que cumple con todas nuestras necesidades. Además este lenguaje nos proporciona, respecto a otros lenguajes similares como puede ser ASP.net, las siguientes ventajas:

- Se trata de un lenguaje multiplataforma, con lo que puede trabajar sobre la gran mayoría de sistemas operativos.
- Gran compatibilidad con la mayoría de gestores de BBDD actuales.
- Posee una gran documentación.
- Se trata de software libre, con lo que reducimos costes.
- Permite técnicas de programación orienta a objetos.
- Permite ampliar su potencial mediante la implementación de módulos externos (extensiones).
- Fácil aprendizaje.

Como podemos ver, el abanico de ventajas que nos ofrece este lenguaje cumple con las necesidades que se nos presentan, lo que indica que se trata del lenguaje ideal para el desarrollo del proyecto.

Como desventajas podemos decir que php tiene muy pocas, pero cabe destacar que php consume más recursos que otros lenguajes, lo que provoca un lentitud en el acceso a los datos mediante las llamadas y ejecución de funciones. Esto normalmente se agrava sobretodo en paginas colgadas en Internet, pero en nuestro caso será casi inapreciable, ya que se trata de una web alojada en un servidor web de una red interna.

#### **4.1.4.2 Base de datos y Servidor Web**

La elección del programa gestor de la BBDD va muy ligado con la elección del lenguaje de programación, php, ya que se intenta que la compatibilidad entre ambos sea la mayor posible. El gestor con una mayor compatibilidad con dicho lenguaje, en estos momentos, es sin duda MySQL.

Pero también hemos de tener en cuenta la compatibilidad entre el programa gestor de la base de datos y el servidor web, ya que mediante la web accederemos a los datos almacenados en la base de datos. Para obtener una máxima compatibilidad utilizaremos un servidor web Apache. La implantación, en un único servidor, de MySQL y Apache nos garantiza una máxima compenetración entre ambos, que sin duda notaremos a la hora de trabajar con el programa.

La versión que utilizaremos para cada uno de los programas es:

- MySQL versión 5.0.51
- Apache versión 2.2.8

También se ha tenido en cuenta, la posibilidad de utilizar otro programa gestor de bases de datos como Microsoft SQL Server 2005, PostgreSQL 8, InterBase SMP 2009, entre otros, pero el incremento del coste total que implica la instalación de estos gestores no es compatible con intención de reducir costes del programa SGAR.

Se estudio, por último, la posible implantación de Oracle Database 10g Express Edition, ya que al tratarse un gestor gratuito cumple con la condición ya comentada anteriormente, pero las limitaciones que imponen a la hora de gestionar grandes volúmenes de datos, nos obligo a descartarla.

En lo que se refiere a las alternativas de servidores web, se miro la posibilidad de instalar un servidor Microsoft IIS 6.0, ya que es gratuito al igual que Apache, pero al no ofrecer las misma prestaciones que Apache se descarto. Apache es mucho más potente y configurable que IIS y además posee un sinfín de información.

#### **4.1.4.3 Sistemas Operativos**

En lo que se refiere a los sistemas operativos, utilizaremos la distribución Ubuntu 8.04 LTS (nombre en clave "Hardy Heron") para el servidor, el sistema operativo Microsoft Windows XP Profesional para el ordenador de gestión de tareas y por ultimo Windows Mobile 5 para los dispositivos móviles.

La elección de la versión 8.04 de Ubuntu, viene determinada por la compatibilidad que ofrece este sistema operativo con los programas MySQL y Apache. De hecho, con un solo comando podemos instalar los dos programas, ya que esta versión contiene los paquetes para la instalación de dichos programas. Además, se ha de decir, que al implantar un servidor basado en Linux, además de aprender nuevas tecnologías, conseguimos una alta fiabilidad ya que diariamente se revisan y actualizan paquetes que pueden ocasionar algún tipo de problema. Por último se ha de decir que se trata de una versión LTS (Long Term Support), en la que se da soporte durante 3 años y con lo que nos aseguramos una versión suficientemente estable y actualizada como para instalarla en un servidor.

Al elegir Windows XP Professional para el ordenador de gestión de tareas, se pensó en que si se instalaba una versión Linux, el usuario tendría una dificultad añadida ya que al no conocer cómo funciona el sistema operativo, debería dedicar un tiempo al aprendizaje de este, lo que supone un incremento del coste total para la empresa y un retraso en la implantación del sistema.

Por último, en la elección del sistema operativo para los dispositivos móviles, podemos decir que Windows Mobile 5 es el sistema operativo que llevan todos los dispositivos, o casi todos. De hecho, el dispositivo que se adquirirá para los operarios, lleva instalado Windows Mobile 5, con lo cual no podemos cambiarlo y se tendrá que utilizar este. Podemos decir también que actualmente no hay una gran variedad de sistemas operativos para dispositivos móviles, y las alternativas que hay a Windows Mobile no ofrecen el soporte que ofrece este.

### 4.2 Módulos de la aplicación

En el siguiente diagrama se muestra el proceso que sigue un material desde que entra al almacén hasta que sale:

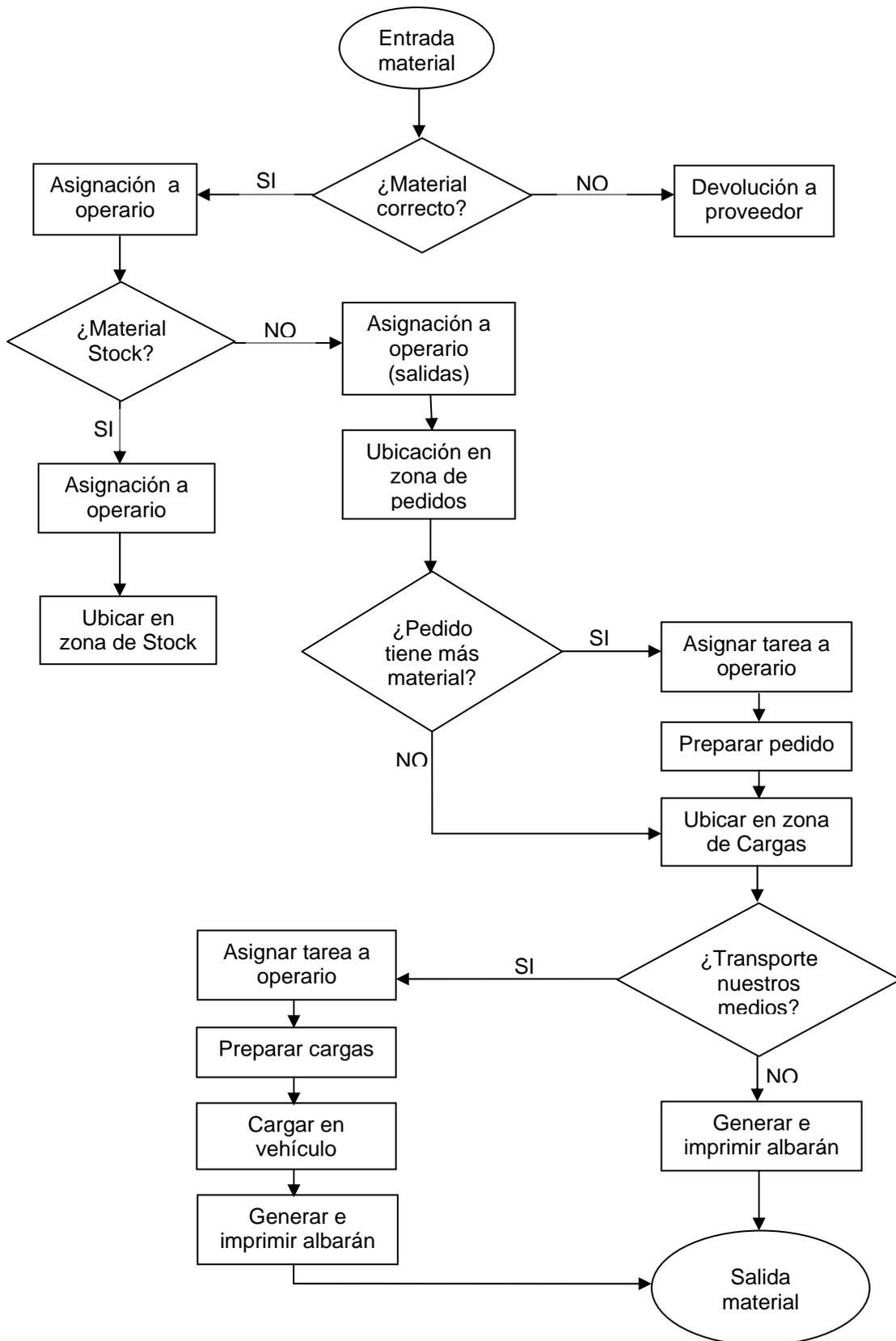


Fig. 2 - Diagrama de flujo de los procesos efectuados sobre los materiales

A continuación, se muestra un diagrama con los posibles estados en que pueden encontrarse los materiales:

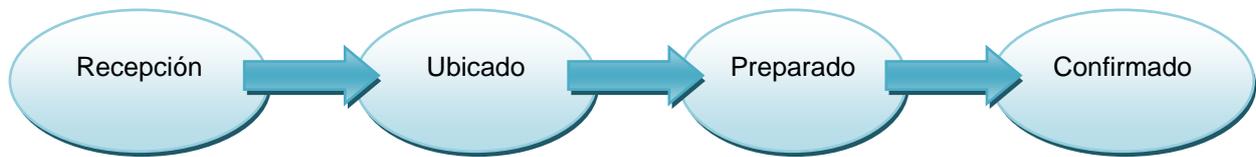


Fig. 3 – Estados posibles de un material

## 4.2.1 Módulo de entrada de material

### 1. Asignación de tarea a operario

Una vez se hayan introducido los albaranes de compra en el sistema, mediante el programa específico de la empresa, el encargado de almacén podrá asignar a un usuario determinado una serie de materiales a recontar según la prioridad que crea oportuna.

Al entrar a dicha opción, se genera un listado de material a recontar que mediante un check, iremos marcando los material que queremos asignar. Una vez marcados todos los que se quieren, seleccionaremos el operario que realizara la tarea y le asignaremos una prioridad.

Una vez asignada la tarea, será visible solamente por el usuario asignado y solo se podrá acceder mediante su sesión. Las tareas se ordenaran según la prioridad de mayor a menor.

### 2. Recuento de material de entrada

Para realizar el recuento de material de entrada, accederemos a dicha opción, y se nos mostrara, mediante un listado, los materiales asignados a nuestro usuario. Estos se mostraran por orden de prioridad y accederemos al recuento de cada uno de ellos mediante la opción de “Ubicar”.

Cuando se haya seleccionado uno, aparecerá otra pantalla donde se pide introducir los siguientes datos:

- Proveedor y número de albarán del material de entrada.
- Descripción de dicho material.
- Cantidad teórica, según albarán de entrada, y cantidad física real.
- Numero de bulto, si ya existe más material, el sistema rellenara este campo con el numero de bulto correspondiente.
- Ubicación donde hay más material y ubicación donde realmente se ubicara el material.
- Opción para marcar si la ubicación donde hemos ubicado el material ya no se puede ubicar nada más porque está completa.

Al introducir estos datos y validarlos, se nos ubicara el material y ya estará disponible como stock en el almacén. Una vez ubicado, el material ya no será visible, ni el listado de asignación de tareas ni en el de reasignación.

### 3. Reasignación de tarea a operario

Existe la posibilidad de repartir la carga de trabajo de un operario a otro. Si se observa que la cola de trabajo de un operario no disminuye, por cualquier motivo, mediante la opción de reasignación, podremos traspasar esa tarea a otro operario para que la lleve a cabo.

Esta opción es muy similar a la de asignación de tareas, con la diferencia de que se nos muestra el usuario que tiene esa tarea asignada y la prioridad de esta. Una vez seleccionada las tareas a reasignar, indicaremos el nuevo usuario que las llevara a cabo y la prioridad.

## **4.2.2 Módulo de salida de material**

### 1. Asignación de tarea a operario

Una vez se hayan introducido los pedidos de clientes, se mostrara mediante un listado, la relación de pedidos a preparar. El funcionamiento de dicho listado, es muy similar al de asignación de entradas de material con la diferencia de que se muestran materiales que se han de preparar para pedidos. Una vez marcados los pedidos a preparar, mediante el check de “Preparar”, seleccionaremos el operario que realizara la tarea y la prioridad de esta.

### 2. Preparar pedido

Una vez asignados los pedidos a preparar, cada usuario podrá visualizar los que se le han asignado mediante la validación de su sesión. Una vez validada, podrá acceder a la opción de preparar pedido, donde se le mostrara un listado con los pedidos que se le han asignado y en cada uno de ellos el material a preparar. Al lado de cada línea de material le aparecerá un botón “Preparar” y al hacer clic en él, nos remitirá a otra pantalla donde se nos muestra la siguiente información:

- Nº de pedido y proveedor del material.
- Descripción del material y cantidad a preparar.
- Ubicación del material en el almacén y stock disponible.
- Numero de bulto del material de stock.
- Una casilla donde introduciremos la cantidad que hemos preparado.
- Otra casilla donde indicaremos la ubicación donde hemos depositado el material.
- Por último, una casilla donde introduciremos el número de bulto del pedido.

Una vez introducidos los campos necesarios y que ya hayan sido validados, dicha línea de material pasara a estado de “Preparado”. Hasta que no hayamos acabado todas la líneas de material del pedido asignado, el pedido no estará listo

para que sea recogido o enviado al o por el cliente. Por último, dicho pedido desaparecerá del listado de asignación y reasignación de tareas.

### 3. Reasignación de tarea a operario

Como en el módulo de entradas, también existe la posibilidad de reasignar una tarea de preparación de pedido desde un operario a otro. Para ello, accederemos a dicha opción, donde se nos mostrara un listado con los pedidos asignados a todos los operarios en el cual marcaremos los pedidos a reasignar e indicaremos el nuevo operario y la prioridad.

## **4.2.3 Módulo de mantenimiento del almacén**

### 1. Recuento de bulto

Para poder visualizar el contenido de un bulto, accederemos a esta funcionalidad desde el menú principal e introduciremos el número de bulto en el formulario inicial. Una vez introducido y validado en el sistema, nos aparecerá un listado con el contenido de dicho bulto indicándonos si se trata de un bulto de stock o de pedido.

En el informe nos aparecerá la siguiente información:

- Número de albarán de entrada o de pedido, según la finalidad del material.
- Descripción del material.
- Cantidad.
- Ubicación.

### 2. Regulación de stock

Teniendo en cuenta de la necesidad de realizar como mínimo una vez al año, un inventario general de almacén, existe la opción de realizar regulaciones de stock de materiales que componen un bulto. Se ha de comentar, que las regulaciones de stock tan solo se pueden realizar sobre material de stock ya que no tendría sentido realizarlo sobre pedidos, ya que ese material no cuenta como stock.

Para ello, introduciremos el número de bulto, y nos aparecerá la relación de material que posee dicho bulto. A continuación introduciremos la cantidad real existente para cada línea de material y regularemos uno por uno.

### 3. Mover bulto

Si tenemos la necesidad de mover un bulto de una ubicación a otra, existe la posibilidad de realizarlo mediante la funcionalidad de mover bulto que ofrece el programa. Para ello, deberemos introducir el número de bulto a mover junto con la antigua ubicación y la nueva ubicación y aceptamos para que quede reflejado el movimiento en el sistema.

#### **4.2.4 Modulo de informes**

##### 1. Informe de ubicaciones

El primer informe que encontramos, es el informe de ubicaciones. Dicho informe nos ofrece un listado del material que hay en las ubicaciones que hemos filtrado en la selección. Para ello, realizaremos en una pantalla previa al visor de informes una selección de las ubicaciones que queremos filtrar y marcaremos si queremos la opción de “Inventario”. Si marcamos dicha opción, no tendrá en cuenta los materiales que han sido preparados para algún pedido.

A continuación nos aparecerá el visor de informes que nos mostrara la siguiente información:

- Ubicación, nº de bulto, descripción del material, cantidad y nº de albarán.

##### 2. Informe histórico

Para obtener información sobre quién y cuándo un operario preparo o ubico un material, accederemos al informe histórico, en el cual, introduciremos un intervalo de fechas para acotar la selección. Tenemos también la opción de incluir en la consulta las salidas de material que se han realizado sobre esa fecha.

Una vez introducida dicha información, el visor de informes nos mostrara la siguiente información:

- Fecha en que se realizo la acción, nº de bulto del material, descripción de este, cantidad, ubicación donde se encontraba el material y operario que realizo la operación de entrada o salida dependiendo de si marcamos la opción anteriormente comentada.

##### 3. Informe artículos

Por último, podremos realizar una búsqueda por palabra clave de artículo. Al introducir el nombre del artículo, podremos marcar la opción de buscar también en las salidas de material.

Una vez introducida la palabra clave, el sistema realizara una búsqueda y mostrara todos los artículos que contienen esa palabra, para más tarde mostrarlo en el visor de informes con la siguiente información:

- Ubicación del material, nº de bulto de este, descripción, cantidad y fecha de entrada o de salida, teniendo en cuenta si hemos marcado la opción de tener en cuenta las salidas.

#### **4.2.5 Modulo de parámetros del Sistema**

##### 1. Alta y modificación de usuarios

El sistema permite crear nuevos usuarios y modificar ciertos parámetros de los ya existentes. Dicha acción tan solo la podrá realizar el encargado de almacén, que será el responsable de gestionar a los usuarios.

A la hora de realizar el alta de un usuario se deberán introducir los siguientes datos:

- Nombre y apellidos del operario.
- Nombre de usuario para validar la sesión y acceso al programa.
- Password de acceso a este.
- Cargo que ocupara en la empresa.

Por último, a la hora de modificar algún parámetro de usuario, podremos cambiar los campos anteriormente comentados o incluso si se desea dar de baja al usuario de forma indefinida.

## 2. Alta y modificación de ubicaciones

Otro parámetro que podrá modificar o crear el encargado de almacén, son las ubicaciones. Al igual que con los usuarios, se podrán modificar varios parámetros o incluso eliminar la ubicación.

A la hora de realizar el alta de una ubicación se deberán introducir los siguientes datos:

- Calle, pila y altura de la nueva ubicación.
- Familia de productos que almacenara.

En cambio, a la hora de realizar alguna modificación, tan solo podremos modificar los campos *Familia de productos* y *Ubicación completa*, ya que no tendría sentido modificar la propia ubicación, si fue creada con dichos datos que cumplen un estándar de codificación de ubicaciones.

## 3. Generar e imprimir códigos de bulto

Una de las funcionalidades más importantes del programa es la gestión de las etiquetas de bulto. Dichas etiquetas son uno de los fundamentos en que se basa este sistema de trabajo. Para ello se han diseñado dos funcionalidades que gestionan estas etiquetas.

Con la opción de generar bulto, podemos ir generando, conforme vayamos necesitando, números de bultos para que puedan ser utilizados. En la pantalla de generación se muestra el ultimo bulto que se genero y el numero de bultos que queremos generar a partir de este, para que se siga un correcto orden y no se desorden los bultos en la base de datos.

Una vez generado los números, podremos imprimir las etiquetas que irán pegadas en palets, ubicaciones, cajas, etc. Para ello, podremos imprimir un rango de etiquetas, desde un número inicial hasta uno final, o imprimir un número en concreto. Al generarse la vista previa, podremos imprimir dichas etiquetas mediante el icono de impresión, el cual nos abrirá el menú de impresión en el cual seleccionaremos la impresora de etiquetas correspondiente.

### 4.3 LOPD

La Ley Orgánica de Protección de Datos (LOPD), es una ley española que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar.

Su objetivo principal es regular el tratamiento de los datos y ficheros, de carácter personal, independientemente del soporte en el cual sean tratados, los derechos de los ciudadanos sobre ellos y las obligaciones de aquellos que los crean o tratan.

A continuación se citan brevemente las obligaciones básicas que impone dicha ley:

- Inscripción de los ficheros en el Registro General de la Protección de Datos
- Redacción del documento de seguridad.
- Redacción de cláusulas de protección de datos.
- Auditoría de Protección de Datos.
- Adoptar medidas técnicas y organizativas.
- Adaptación a futuras acciones.

La ley clasifica 3 niveles de seguridad en función de los datos que se almacén. Estos niveles son:

- **Bajo:** Nombre, apellidos, direcciones de contacto, teléfonos y datos bancarios.
- **Medio:** Comisión infracciones penales, Comisión infracciones administrativas, Información de Hacienda Pública, Información de servicios financieros.
- **Alto:** Ideología, Religión, Creencias, Origen racial, Salud, entre otros.

En principio, la aplicación estaría clasificada como nivel bajo, ya que solo almacena datos básicos de operarios y proveedores. De todas formas, esto no quiere decir que no se tengan que seguir una serie de procedimientos para que se cumpla correctamente dicha ley.

A grandes rasgos, los procedimientos a seguir serian los siguientes:

- Inscripción de los ficheros de datos en la AEPD.
- Creación de avisos de seguridad para la web.
- Creación del documento de seguridad.
- Desarrollo de un Informe de Recomendaciones, para que pueda implementar todas las medidas de seguridad para cumplir con la legislación de la manera más eficaz posible.
- Elaboración de una guía para la correcta adaptación a la LOPD.

Si no se siguen estos procesos, nos abstenemos a las siguientes sanciones:

- **LEVES:** Multa de 601,01 € a 60.101,21 €
- **GRAVES:** Multa de 60.101,21 € a 300.506,05 €
- **MUY GRAVES:** Multa de 300.506,05 € a 601.012,10 €

Para más información, se puede consultar la página web de la LOPD que encontraremos en la bibliografía de este documento.

#### 4.4 Tareas previas al diseño de la aplicación

Para finalizar este capítulo, se describe las tareas a seguir después de haber realizado el análisis de la aplicación:

- Aprender los lenguajes en que se programara la aplicación (php, JavaScript y Ajax)
- Montaje de un servidor virtual donde realizar la implementación y codificación del aplicativo:
  - Montar el sistema operativo sobre el que trabajaremos.
  - Montar el servidor Apache.
  - Montar el gestor de la BBDD (MySql).
- Diseñar e implementar la base de datos en el gestor.
- Realizar el diseño de la aplicación.
- Empezar a programar los módulos correspondientes.

## Capítulo 5: Diseño

### 5.1 Arquitectura

La aplicación se basa en una arquitectura cliente-servidor LAMP. En dicha arquitectura se siguen los siguientes procesos:

- Cliente Web: Solicitud de conversión del nombre de una máquina al Servidor DNS.
- Servidor DNS: Recepción y tratamiento de la solicitud. Una vez recibida la petición realiza las consultas necesarias para resolver y obtener la dirección IP.
- Servidor DNS: Devuelve la dirección IP que corresponde al Servidor Web al navegador Web.
- Cliente Web: Realiza la solicitud de información mediante una URL (Método GET) o un formulario (Método POST). Dicha solicitud incluye: la dirección IP del servidor web, el puerto del servidor web, URL y parámetros POST.
- Servidor Web: Control de Acceso, Análisis de la petición y localización del recurso.
- Servidor Web: Selección del módulo de PHP para interpretar el fichero PHP elegido.
- Módulo de PHP: Ejecución del programa PHP elegido.
- Aplicación PHP: Procesado de los parámetros GET o POST.
- Aplicación PHP: Realización de las llamadas SQL al Servidor de SGBD.
- SGBD: Recogida y procesado de la petición SQL.
- SGBD: Retorno de los datos solicitados.
- Aplicación PHP: Procesado de los datos devueltos por el SGBD y Generación del HTML resultante.
- Modulo PHP: Paso del HTML resultante a Apache.
- Apache: Devuelve el HTML al cliente Web.
- Cliente Web: Presentación del HTML resultante, CSS y elementos relacionados y ejecución del Javascript.

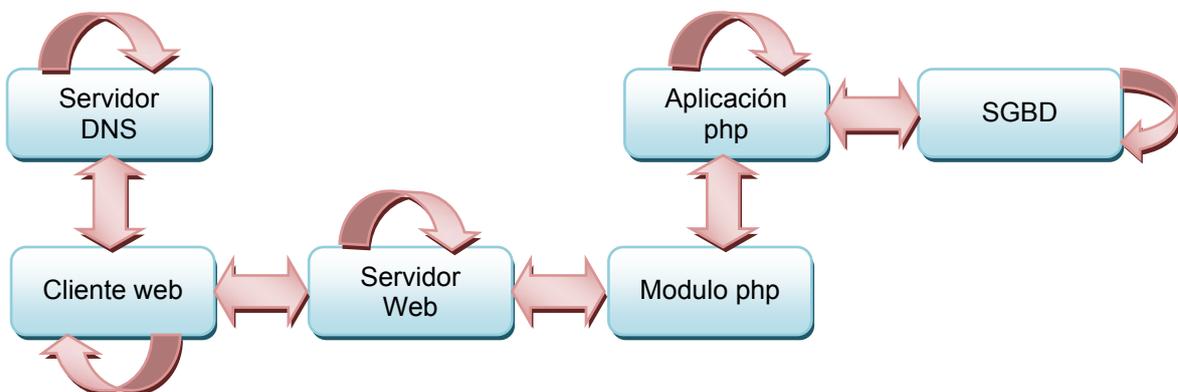


Fig. 4 – Diagrama arquitectura LAMP

### 5.2 Estructura de datos

La estructura de datos nos permite definir las tablas que se utilizarán en la aplicación y los tipos de datos que almacenan estas, indicando las relaciones entre ellas. La aplicación tendrá la siguiente estructura de datos:

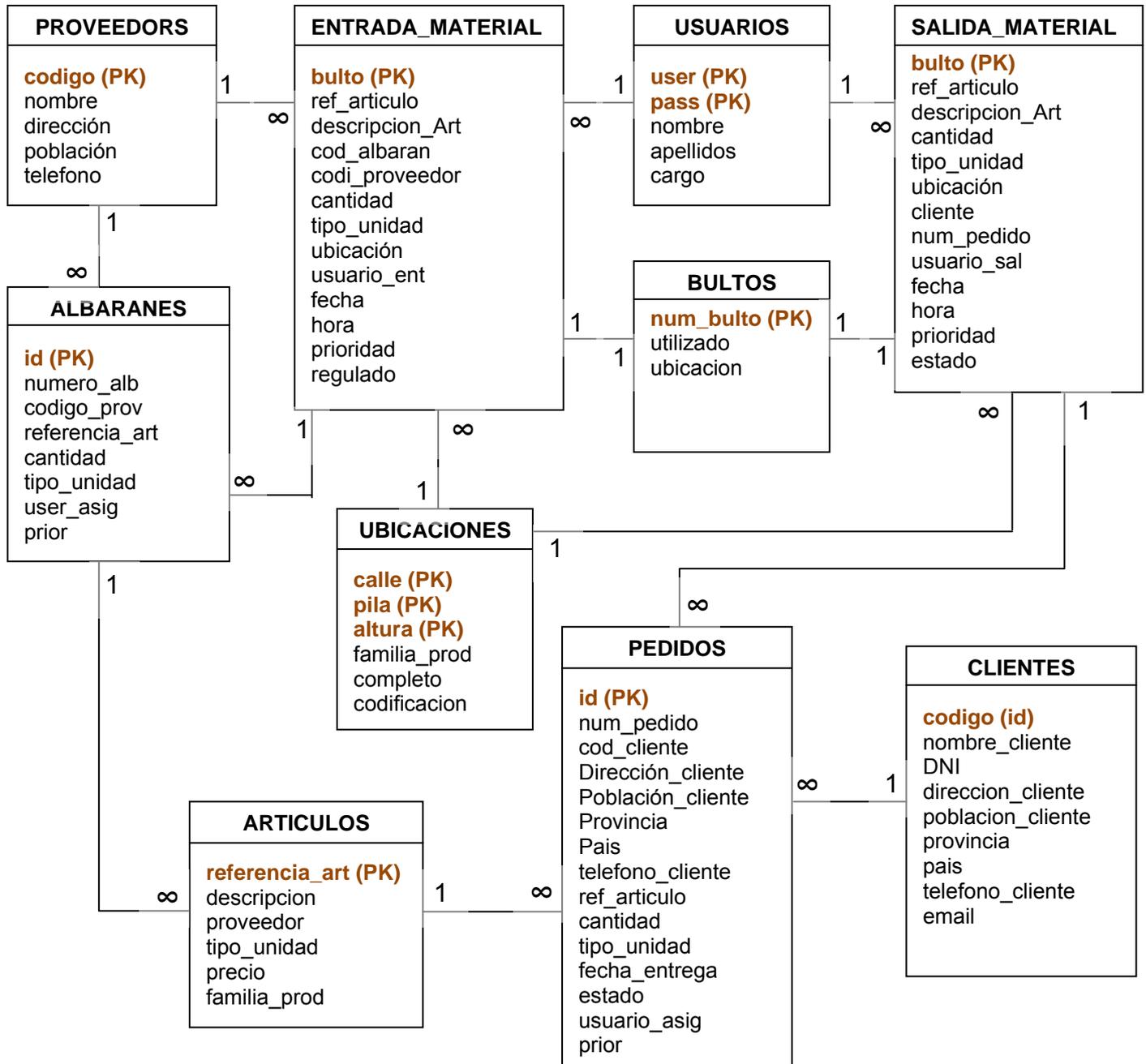


Fig. 5 – Diagrama de la base de datos

Para tener un concepto más claro de cada una se expone, a continuación, de manera más detallada la estructura de datos de cada tabla, así como otras características:

Nombre tabla: Albaranes						
<b>Descripción:</b>		Relación de albaranes de entrada de material				
<b>Relación con:</b>		Entrada_material, Artículos y Proveedores				
PK	Nombre	Tipo	Long.	Valor	Null?	Descripción
X	id	integer	11	-	No	Numero identificador de entrada
	numero_alb	varchar	10	-	Sí	Numero de albarán de entrada
	codigo_prov	varchar	10	-	Sí	Código del proveedor del material
	referencia_art	varchar	10	-	Sí	Referencia del articulo
	cantidad	interger	10	-	Sí	Cantidad de articulo
	tipo_unidad	varchar	2	-	Sí	Tipo de unidad de artículo: UD, M2, LT, KG, etc.
	contado	varchar	1	0	Sí	Indica el estado en que se encuentra el material. Valor 0 es entrado pero no asignado a operario, valor 1 es asignado pero no recontado y valor 2 es ubicado.
	user_asig	varchar	10	-	Sí	Nombre del operario al que se le ha asignado la tarea
	prior	integer	1	-	Sí	Prioridad de la tarea (5→1, donde 5 es la prioridad más baja)

Nombre tabla: Artículos						
<b>Descripción:</b>		Relación de artículos dados de alta en la aplicación				
<b>Relación con:</b>		Albaranes y Pedidos				
PK	Nombre	Tipo	Long.	Valor	Null?	Descripción
X	referencia_art	varchar	10	-	No	Referencia del articulo
	descripcion	varchar	50	-	Sí	Descripción del articulo
	proveedor	varchar	20	-	Sí	Nombre del proveedor
	tipo_unidad	varchar	2	-	Sí	Tipo de unidad de artículo: UD, M2, LT, KG, etc.
	precio	decimal	-	-	Sí	PVP del articulo
	familia_prod	varchar	10	-	Sí	Familia de productos: cerámica, sanitario, menaje, grifería, ferretería, cementos, etc.

Nombre tabla: Bultos						
<b>Descripción:</b>		Relación de números de bultos dados de alta en la aplicación				
<b>Relación con:</b>		Entrada_material y Salida_material				
PK	Nombre	Tipo	Long.	Valor	Null?	Descripción
X	num_bulto	varchar	10	-	No	Numero de bulto. Formato: AAAA000000 incremental
	utilizado	binary	1	0	Sí	Indica si el bulto ha sido utilizado, valor 0 es no utilizado y valor 1 es utilizado.
	ubicacion	varchar	10	-	Sí	Ubicación codificada donde se encuentra el bulto.

Nombre tabla: Clientes						
<b>Descripción:</b>		Relación de clientes de la empresa				
<b>Relación con:</b>		Pedidos				
PK	Nombre	Tipo	Long.	Valor	Null?	Descripción
X	codigo	varchar	5	-	No	Código identificador de cliente
	nombre_cliente	varchar	50	-	Sí	Nombre del cliente
	DNI	varchar	9	-	Sí	DNI o NIF del cliente
	direccion_cliente	varchar	50	-	Sí	Dirección fiscal del cliente
	poblacion_cliente	varchar	35	-	Sí	Población de la dirección fiscal
	provincia	varchar	35	-	Sí	Provincia de la dirección fiscal
	pais	varchar	35	-	Sí	País de la dirección fiscal
	telefono_cliente	varchar	9	-	Sí	Teléfono fijo o móvil del cliente
	email	varchar	35	-	Sí	E-mail de contacto del cliente

Nombre tabla: Entrada_material						
<b>Descripción:</b>		Relación del material de stock del almacén				
<b>Relación con:</b>		Bultos, Usuarios, Ubicaciones, Albaranes y Proveedores				
PK	Nombre	Tipo	Long.	Valor	Null?	Descripción
X	bulto	varchar	10	-	No	Bulto que identifica el material

	ref_articulo	varchar	10	-	Sí	Referencia del material de stock
	descripcion_art	varchar	30	-	Sí	Descripción del material de stock
	cod_albaran	varchar	10	-	Sí	Código del albarán de entrada
	codi_proveedor	varchar	10	-	Sí	Código del proveedor del material
	cantidad	decimal	-	-	Sí	Cantidad de material en stock
	tipo_unidad	varchar	2	-	Sí	Tipo de unidad de artículo: UD, M2, LT, KG, etc.
	ubicacion	varchar	10	-	Sí	Ubicación del material en el almacén
	usuario_ent	varchar	10	-	Sí	Usuario que realizo la ubicación
	fecha	varchar	10	-	Sí	Fecha en que se ubico el material
	hora	varchar	5	-	Sí	Hora en que se ubico el material
	prioridad	varchar	1	-	Sí	Prioridad asignada para ubicar material

<b>Nombre tabla:</b> Pedidos						
<b>Descripción:</b> Relación de pedidos de clientes						
<b>Relación con:</b> Artículos, Salida_material y Clientes						
PK	Nombre	Tipo	Long.	Valor	Null?	Descripción
X	id	integer	11	-	No	Numero identificador de pedido
	num_pedido	varchar	10	-	Sí	Numero de pedido de cliente
	cod_cliente	varchar	10	-	Sí	Código de cliente
	direccion_cliente	varchar	35	-	Sí	Dirección de entrega al cliente
	poblacion_cliente	varchar	35	-	Sí	Población de entrega
	provincia	varchar	25	-	Sí	Provincia de la población
	pais	varchar	25	-	Sí	País de la dirección de entrega
	telefono_cliente	varchar	9	-	Sí	Teléfono de contacto
	ref_articulo	varchar	10	-	Sí	Referencia del articulo deseado
	cantidad	integer	11	-	Sí	Cantidad de articulo a preparar
	tipo_unidad	varchar	2	-	Sí	Tipo de unidad de artículo: UD, M2, LT, KG, etc.

	fecha_entrega	varchar	10	-	Sí	Fecha estimada de entrega
	estado	varchar	1	0	Sí	Indica el estado en que se encuentra el pedido. Valor 0 es pedido no asignado a operario, valor 1 es asignado pero no preparado y valor 2 es preparado.
	usuario_asig	varchar	10	-	Sí	Usuario asignado a la preparación del pedido
	prior	varchar	1	-	Sí	Prioridad de la tarea asignada

<b>Nombre tabla:</b> Proveedores						
<b>Descripción:</b> Relación de proveedores de materiales						
<b>Relación con:</b> Albaranes y Entrada_material						
PK	Nombre	Tipo	Long.	Valor	Null?	Descripción
X	codigo	varchar	5	-	No	Código identificador de proveedor
	nombre	varchar	25	-	Sí	Nombre del proveedor
	direccion	varchar	50	-	Sí	Dirección fiscal del proveedor
	poblacion	varchar	30	-	Sí	Población de la dirección fiscal
	telefono	varchar	9	-	Sí	Teléfono de contacto del proveedor

<b>Nombre tabla:</b> Salida_material						
<b>Descripción:</b> Relación de materiales preparados para pedidos						
<b>Relación con:</b> Bultos, Usuarios, Ubicaciones, Pedidos y Clientes						
PK	Nombre	Tipo	Long.	Valor	Null?	Descripción
X	bulto	varchar	10	-	No	Bulto que identifica el material
	ref_articulo	varchar	10	-	No	Referencia del material pedido
	descripcion_art	varchar	30	-	Sí	Descripción del material pedido
	cantidad	integer	11	-	Sí	Cantidad de material a preparar
	tipo_unidad	varchar	2	-	Sí	Tipo de unidad de artículo: UD, M2, LT, KG, etc.
	ubicacion	varchar	10	-	Sí	Ubicación del pedido preparado en el almacén.

	cliente	varchar	10	-	Sí	Nombre del cliente del pedido
	num_pedido	varchar	10	-	Sí	Numero de pedido de cliente
	usuario_sal	varchar	15	-	Sí	Usuario que realizo la preparación del pedido
	fecha	varchar	10	-	Sí	Fecha en que se preparo el pedido
	hora	varchar	5	-	Sí	Hora en que se preparo el pedido
	prioridad	varchar	1	-	Sí	Prioridad asignada para realizar la tarea de preparar el pedido

<b>Nombre tabla:</b>		<b>Ubicaciones</b>				
<b>Descripción:</b>		Relación de ubicaciones que existen en el almacén				
<b>Relación con:</b>		Entrada_material y Salida_material				
PK	Nombre	Tipo	Long.	Valor	Null?	Descripción
X	calle	varchar	4	-	No	Código de calle del almacén
X	pila	varchar	3	-	No	Código de pila de la calle
X	altura	varchar	1	-	No	Código de altura de la pila
	familia_prod	varchar	10	-	Sí	Familia de productos: cerámica, sanitario, menaje, grifería, ferretería, cementos, etc.
	completo	tinyint	1	-	Sí	Indica si la ubicación esta completa o no. Valor 0 indica que está vacía y valor 1 indica que esta completa

<b>Nombre tabla:</b>		<b>Usuarios</b>				
<b>Descripción:</b>		Relación de usuarios dados de alta en la aplicación				
<b>Relación con:</b>		Entrada_material y Salida_material				
PK	Nombre	Tipo	Long.	Valor	Null?	Descripción
X	user	varchar	15	-	No	Nombre de usuario para validarse
X	pass	varchar	9	-	No	Password de usuario para validarse
	nombre	varchar	30	-	Sí	Nombre del operario o responsable
	apellidos	varchar	30	-	Sí	Apellidos del operario o responsable
	cargo	varchar	15	-	Sí	Cargo que ocupa: operario o responsable

### 5.3 Consideraciones previas

Antes de volcarnos plenamente en el diseño deberemos de tomar un conjunto de decisiones tales como, el árbol de navegación, la estructura de las páginas del aplicativo y la accesibilidad de cada una de estas.

A continuación se muestra el árbol de navegación principal para obtener una visión general de las opciones que ofrece el programa.

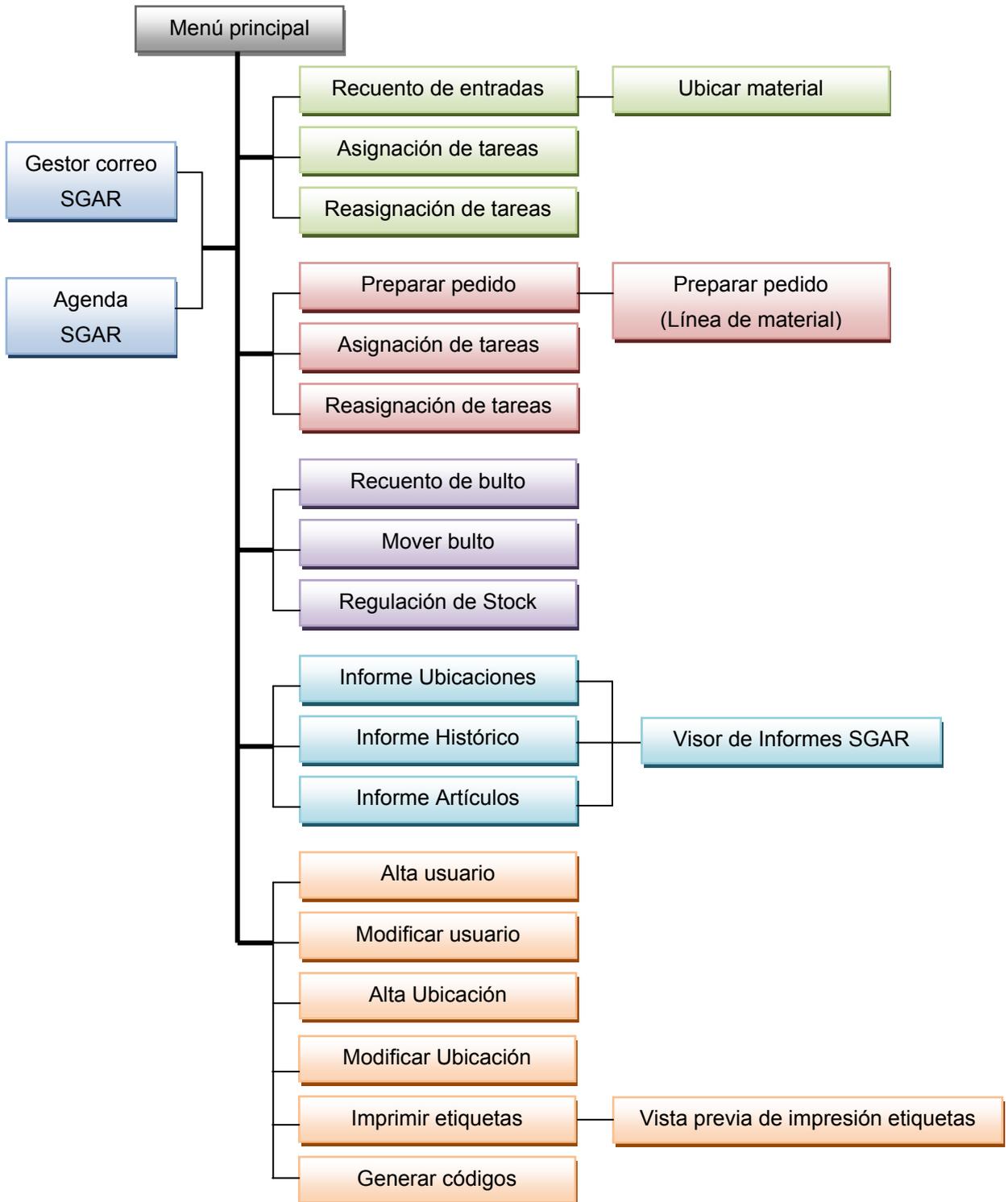


Fig. 6 - Árbol de navegación

La estructura de las páginas del aplicativo es un punto muy importante ya que hemos de realizar una buena estructuración para poder acceder a cada módulo con el menor número posible de pasos. Para ello, se diseñará de forma que no tengamos que utilizar la barra de desplazamiento, de esta forma el acceso a las opciones será rápido.

La estructura se basará en una página central que se distribuirá de la siguiente forma:

- En la parte superior, se mostrará el logotipo de la aplicación junto con la fecha y hora actual, además del nombre del usuario que se ha validado y la opción de “cerrar sesión”. También se mostrará unos accesos directos al gestor de correo del programa, a un calendario y si no nos encontramos en la página principal se colocará un acceso a dicha página. Esta parte es común a todas las páginas de la aplicación.
- La parte central será diferente en función de donde nos hallemos:
  - Si nos encontramos en la página principal encontraremos el menú que estará dividido en cinco módulos, estos son: entrada de material, salida de material, informes, parámetros del sistema y mantenimiento del almacén.
  - Si nos encontramos en cualquier otra página, se mostrará el formulario respectivo a cada opción.

En cambio, en el visor de informes la estructura también se basa en una página principal pero en esta, la parte superior es diferente, ya que nos encontramos directamente el resultado del informe junto con las opciones de impresión y un enlace a la página anterior. Esta página está diseñada que para cuando realicemos un impresión tan solo muestre el resultado de la consulta y no se imprima tal y como esta visualizado en el navegador. Esto se consigue mediante la utilización de css específicos para la impresión, de tal forma que únicamente se utilicen en la vista previa del documento.

Por otro lado, en la impresión de las etiquetas de bulto no utilizaremos la estructura de página central, ya que ha de seguir un formato en concreto para que la impresora de etiquetas la pueda imprimir correctamente. Se visualizará sobre una página en blanco sin ningún tipo de estructura.

Para que la aplicación sea más accesible, deberemos de tener en cuenta que esta aplicación se visualizará en PC's comunes y en dispositivos móviles. Para ello tendremos en cuenta cada resolución de los diferentes dispositivos donde se visualizará la aplicación:

En el PC de gestión la resolución de la pantalla deberá de ser de 1024 x 768 con un mínimo de 16 millones de colores. En cambio, en los dispositivos móviles la resolución deberá de ser de 640 x 240 y 16 millones de colores.

Por último, al tratarse de una aplicación que utiliza formularios validados con JavaScript deberemos de tener en cuenta que el navegador, en nuestro caso Mozilla firefox, tenga JavaScript habilitado.

## 5.4 Diseño de la interfaz

### 5.4.1 Estilo de las páginas

Para seguir con la tendencia que existe actualmente en el diseño web, se utilizaran hojas de estilo (css) para la apariencia de las páginas de la aplicación. Mediante la utilización de estas hojas de estilo, podremos controlar todo el diseño de la web, parámetros tales como el tipo de fuente, las imágenes que se utilizan, el tamaño de las capas de diseño, el color de estas, la posición en la página web, entre muchísimos más parámetros.

Las ventajas que nos ofrece este tipo de diseño son las siguientes:

- Separación del contenido y presentación.
- Flexibilidad.
- Unificación del diseño de las páginas del sitio.
- Optimización de los tiempos de carga y de tráfico en el servidor.
- Precisión o elasticidad.
- Accesibilidad y estructuración.
- Limpieza del código fuente.
- Compatibilidad y continuidad.
- Estandarización frente a especificaciones propietarias.
- Permite la diferenciación de estilos para imprimir / visualizar en pantalla.

El principal inconveniente de este sistema es, que no todos los navegadores visualizan de la misma forma la hoja de estilo. Pero en nuestro caso no es un problema, ya que tan solo utilizaremos Mozilla Firefox, ya que en Linux no está disponible Internet Explorer.

### 5.4.2 Colores y fuentes utilizados

Los colores y fuentes utilizados deben ofrecer una correcta visualización del contenido. Además nos abstendremos de utilizar colores estridentes para no provocar fatiga visual, ya que al tratarse de un programa que se utilizara durante bastantes horas, colores muy llamativos provocan cansancio visual que a su vez provocaría desconcentración y dolores de cabeza.

En lo que se refiere a las fuentes, se decidió utilizar una fuente que fuese clara y lo mas legible posible para una mejor comprensión. Como en Linux existe la posibilidad de instalar fuentes TrueType, se decidió que una fuente que cumpliese estas características y que además fuese considerada fuente de web segura, era sin duda Arial.

Otro tema relacionado con las fuentes, era la generación de los códigos de bulto. En principio se pensó en utilizar una fuente que generase códigos EAN13, pero este tipo de códigos a una cierta distancia no son legibles por un lector de código de barras. Para ello, se decidió escoger una fuente que generase un código de mayor tamaño, Code128, el cual sí que es legible a más distancia.

En resumen, utilizaremos los siguientes colores y fuentes para la aplicación:

#### Colores

- Como fondo de la aplicación → #B0D1F0
- Para los bordes de la tabla del menú principal → #BEBEBE
- Mensaje de error de validación o generación etiq. → #F84040
- Para el borde la opción cerrar sesión → #1DA4DB
- Para la cabecera del visor de informes → #575252
- Para el fondo del visor de informes → #FFFFFF

#### Fuentes

- Para botones y títulos → Arial, 15px en negrita
- Para contenido de formularios → Arial, 15px
- Para accesorios directos del menú principal → Arial, 12px
- Para contenido de informes → Arial, 13px
- Para mostrar mensajes de error → Arial, 16px en color rojo
- Códigos de barras → Code128

### **5.4.3 Imágenes e iconos**

En lo que se refiere a imágenes e icono deberemos tener las siguientes consideraciones:

- Al tratarse de una página web, las imágenes deberán ocupar el mínimo espacio posible. Para ello, utilizaremos imagen con formato JPG y PNG. Las que más utilizaremos serán las PNG ya que son mejores para el tema de las transparencias, y como la aplicación tenemos un fondo en rayas, nos interesa que las imágenes nos muestren dicho fondo y no un fondo en blanco como muestran los formatos JPG.
- Utilizaremos una resolución de 48x48 pixeles para los iconos de acceso directo a la agenda, gestor de correo, home y impresión. Para el fondo de cada uno de los módulos del menú principal utilizaremos imágenes con una resolución de 250x180 pixeles exceptuando en el menú de parámetros que utilizaremos una imagen de 582x180 pixeles. El logotipo de la aplicación tendrá una resolución de 144x166 pixeles.

Todas las imágenes, excepto el logotipo, provienen del tema de iconos llamado Mashup de Gnome. Todas tienen formato PNG, ya que es el estándar de iconos de Linux.

- Todas las imágenes utilizadas las podremos encontrar en la carpeta “images” en el directorio donde se encuentra la web.

## 5.5 Elementos de la interfaz

A continuación se comentará de una forma más detallada cada una de las dos interfaces que existirán. Se explicara con detalle todas las partes que forman la aplicación, entre ellas los menús, formularios, mensajes de advertencia o error y demás elementos que contienen las páginas que forman el aplicativo.

### 5.5.1 Interfaz para PC

#### 5.5.1.1 Menú de acceso

En esta pantalla nos encontraremos una película de flash y un formulario de validación donde introduciremos el nombre de usuario y password para acceder al menú principal.

Una vez introducidos, los datos serán validados en la base de datos, y si son correctos la aplicación nos remitirá al menú principal. Si los datos son incorrectos, nos aparecerá una pantalla mostrando un error de validación donde se nos indica que el usuario o el password introducidos son incorrectos, además de un vínculo a la página de validación.

En caso de no recordar ni el usuario ni el password, nos pondremos en contacto con el administrador del sistema para que nos lo facilite.

#### 5.5.1.2 Menú principal

En el menú principal nos encontraremos los cinco módulos que componen esta aplicación, tal y como podemos observar en la siguiente imagen:



Fig. 7 – Menú principal de la aplicación

Cada uno de los módulos, junto con sus vínculos, nos dará acceso a las diferentes funcionalidades que se explican a continuación.

### 5.5.1.3 Entradas material

Dentro de este módulo nos encontramos tres opciones: recuento de material, asignación a operario y reasignación a operario.

#### Recuento de material

Se nos mostrará un listado con los materiales a contar que el responsable de almacén ha asignado a nuestro usuario.

ALBARAN	PROVEEDOR	MATERIAL	CANTIDAD	
101	CEMEX, S.A.	HORMIGON SECO 25 KG	25 UD	<b>Ubicar</b>
102	LAND, S.A.	CEMENTO PORTLAND 35KG	100 UD	<b>Ubicar</b>
103	GROHE, S.A.	MONOMANDO LAVABO STYLE CR	15 UD	<b>Ubicar</b>
104	CEMEX, S.A.	MORTERO DE CAL 25 KG	56 UD	<b>Ubicar</b>
105	CEMENTS MOLINS, S.A.	ARENA FINA 25 KG	56 UD	<b>Ubicar</b>

Fig. 8 – Listado de materiales a contar por operario

Si accedemos a la opción ubicar de cada una de las líneas, esta nos remitirá a un formulario donde se ha auto-rellenado con la información de dicha línea.

Proveedor: CEMEX, S.A.  
 Numero Albaran: 101  
 Descripción: HORMIGON SECO 25 KG  
 Cantidad Teorica: 25 UD ?  
 Cantidad Real:  ?  
 Numero de Bulto:  ?  
 Ubicacion Propuesta: 1000/100/2 ?  
 Ubicacion Real:  ?  
 Ubicacion completa ?

**Ubicar** **Atras**

Fig. 9 – Formulario de ubicación de material

Si no rellenamos ninguno de los campos en blanco, se mostrará un mensaje indicando que campos faltan por rellenar:

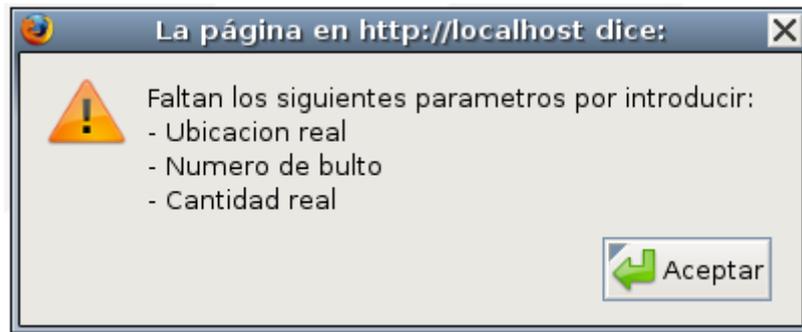


Fig. 10 – Mensaje de aviso de campos necesarios para ubicar material

Como ayuda al usuario, al lado de cada campo hay un icono de ayuda que al poner el puntero del ratón encima nos aparecerá un recuadro, que se ha programado en Ajax, donde se explica de forma detallada, el significado de este campo.



Fig. 11 – Mensaje de ayuda de formulario de ubicar material

Una vez introducidos y validados todos los datos serán guardados en la base de datos y se nos remitirá de nuevo a la página de recuento de entradas, donde no estará la línea de material que acabamos de ubicar.

Si al usuario no se le ha asignado ningún material a recontar, en esta página nos aparecerá un mensaje de advertencia indicándonos que no se han asignado materiales a este usuario y además un vínculo a la página anterior (recuento de entradas).



Fig. 12 – Mensaje de aviso de no asignación de tareas de recuento

### Asignación de tareas

En esta página nos aparecerá un listado de los materiales que han sido introducidos mediante un albarán de entrada.

Para asignar tareas a un operario, seleccionaremos cada línea de material deseada e indicaremos a que usuario se le va asignar y con qué prioridad.

ALBARAN	PROVEEDOR	MATERIAL	CANTIDAD	
101	CEMEX, S.A.	HORMIGON SECO 25 KG	25 UD	<input type="checkbox"/> Asigna
102	LAND, S.A.	CEMENTO PORTLAND 35KG	100 UD	<input type="checkbox"/> Asigna
103	GROHE, S.A.	MONOMANDO LAVABO STYLE CR	15 UD	<input type="checkbox"/> Asigna
104	CEMEX, S.A.	MORTERO DE CAL 25 KG	56 UD	<input type="checkbox"/> Asigna
105	CEMENTS MOLINS, S.A.	ARENA FINA 25 KG	56 UD	<input type="checkbox"/> Asigna
106	BELLOTA, S.A.	MARTILLO DESENCOFRANTE BELLOTA	5 UD	<input type="checkbox"/> Asigna
108	MIRAGE ITALY, S.C.L.	MOSAICO LIFESTONE MALLA 31x31	25 M2	<input type="checkbox"/> Asigna
109	HENKEL, S.A.	JABON LIQUIDO ACTIVO	50 LT	<input type="checkbox"/> Asigna
110	ROCA, S.A.	LAVABO VICTORIA 58x45	10 UD	<input type="checkbox"/> Asigna
110	ROCA, S.A.	INODORO VICTORIA S/H BLANCO	10 UD	<input type="checkbox"/> Asigna
111	BELLOTA, S.A.	JUEGO LLAVES ALLEN - TOR	5 UD	<input type="checkbox"/> Asigna

Operario:  | Prioridad:

**Asignar**

Fig. 13 – Asignación de tareas de recuento de entradas a operarios

Si seleccionamos una tarea y asignamos directamente sin indicar el nombre de usuario o la prioridad, el sistema nos mostrará un mensaje indicándonos que campos faltan para asignar la tarea:

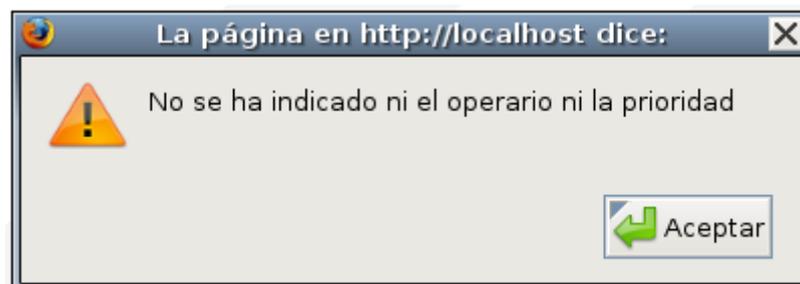


Fig. 14 - Mensaje de error por falta de parámetros de asignación

Este mensaje es dinámico ya que indica que campos han sido introducidos y cuáles no.

### Reasignación de tareas

El diseño de esta pantalla es muy similar a la de asignación de tareas, tan solo cambia la información del listado. Al igual que la asignación de tareas, seleccionaremos los materiales que queremos cambiar de operario, seleccionaremos el nuevo operario y la nueva prioridad y asignaremos de nuevo.

ALBARAN	MATERIAL	CANTIDAD	OPERARIO	PRIOR.	
101	HORMIGON SECO 25 KG	25 UD	Luis	4	<input type="checkbox"/> Reasigna
102	CEMENTO PORTLAND 35KG	100 UD	Valeriano	5	<input type="checkbox"/> Reasigna
103	MONOMANDO LAVABO STYLE CR	15 UD	Josep	4	<input type="checkbox"/> Reasigna

Operario:  ▼ Prioridad:  ▼

Fig. 15 – Reasignación de tareas de recuento de entrada operarios

Dicha pagina muestra los mismos errores por falta de parámetros de asignación.

### 5.5.1.4 Salida material

Dentro de este módulo nos encontramos tres opciones: preparar pedido, asignación de tareas y reasignación de tareas.

#### Preparar pedido

En esta opción nos aparecerá un listado de pedidos asignados a nuestro usuario. Cada pedido tendrá una relación de material a preparar y en cada línea de este material tendremos un botón de acceso al formulario de preparación de pedidos.

**PEDIDO:** 00001  
**CLIENTE:** FCD CONSTRUCCIONES, S.A.

MATERIAL	CANTIDAD	PRIOR.	
YESO RAPIDO 25KG	5 UD	5	<input type="button" value="Preparar"/>
HORMIGON SECO 25 KG	18 UD	5	<input type="button" value="Preparar"/>

**PEDIDO:** 00002  
**CLIENTE:** CONSTRUCCIONES YESERO, S.A.

MATERIAL	CANTIDAD	PRIOR.	
CEMENTO PORTLAND 35KG	1 UD	5	<input type="button" value="Preparar"/>

Fig. 16 – Listado de pedidos a preparar asignados

Al seleccionar la opción de preparar de una línea de material se cargará un formulario donde se nos autorrellenará con la información de dicho pedido.

Pedido: 00001

Proveedor: CEMEX, S.A.

Material: YESO RAPIDO 25KG

Cantidad pedido: 5 UD

Ubicacion material: 1000/100/2

Stock: 10 UD

Bulto material: 2009000001

Confirmar Bulto material:  ?

Cantidad preparada:  ?

Ubicacion depositada:  ?

Bulto pedido:  ?

Fig. 17 – Formulario preparar pedido

Tal y como comentamos en la opción de ubicar material, en esta pantalla también tendremos ayuda mediante los iconos con una interrogación que tendremos al lado de diferentes campos del formulario.

Si confirmamos la preparación antes de rellenar los campos en blanco, nos aparecerá un mensaje de advertencia indicándonos que campos nos faltan por introducir.

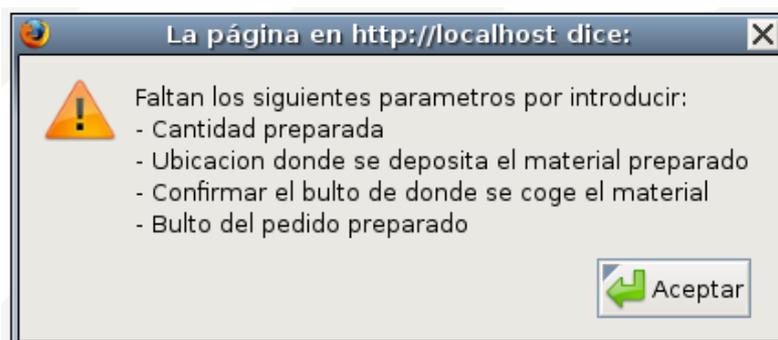


Fig. 18 - Mensaje de advertencia por falta de información

### Asignación de tareas

El diseño de esta pantalla es igual que a la asignación de tareas del módulo de entrada, con la diferencia de la información que se muestra en el listado, es en relación con los pedidos a preparar.

PEDIDO	CLIENTE	FECHA ENTREGA	
00004	REFORMAS E INSTALACIONES ROMERO, S.L.	18/07/2009	<input type="checkbox"/> Asigna
00005	LABORATORIOS ESTEVE, S.A.	22/07/2009	<input type="checkbox"/> Asigna
00006	SACYR VALLHERMOSO, S.A.	22/07/2009	<input type="checkbox"/> Asigna
00007	MARTA LLORENTE VAZQUEZ	23/07/2009	<input type="checkbox"/> Asigna
00008	LACADOS DEL VALLES, S.A.	25/07/2009	<input type="checkbox"/> Asigna

Operario:  | ▼      Prioridad:  | ▼

**Asignar**

Fig. 19 – Asignación de tareas de preparación de pedidos

Reasignación de tareas

Al igual que en la anterior opción, el diseño de esta página es muy similar a la reasignación de tareas al módulo de entrada, con la diferencia que muestra información sobre los pedidos a preparar.

PEDIDO	CLIENTE	OPERARIO	PRIOR.	
00001	FCD CONSTRUCCIONES, S.A.	Valeriano	5	<input type="checkbox"/> Reasigna
00002	CONSTRUCCIONES YESERO, S.A.	Valeriano	5	<input type="checkbox"/> Reasigna
00003	GRAFICAS ROJAS, S.L.	Valeriano	5	<input type="checkbox"/> Reasigna

Operario:  | ▼      Prioridad:  | ▼

**Asignar**

Fig. 20 – Reasignación de tareas de preparación de pedidos

**5.5.1.5 Informes**

Dentro de este módulo existen tres tipos de informes: informe de ubicación, informe histórico e informe de artículos.

Informe de ubicación

Encontraremos un formulario donde indicaremos el intervalo de ubicaciones que queremos listar. Una vez seleccionado, aceptaremos la selección y si el sistema valida correctamente la información introducida, nos remitirá al visor de informes visualizando la información requerida.

Fig. 21 – Formulario de informe de ubicación

Si por el contrario, no introducimos algún valor necesario en el formulario, el sistema nos mostrara un mensaje de advertencia indicándonos que campos faltan por introducir.

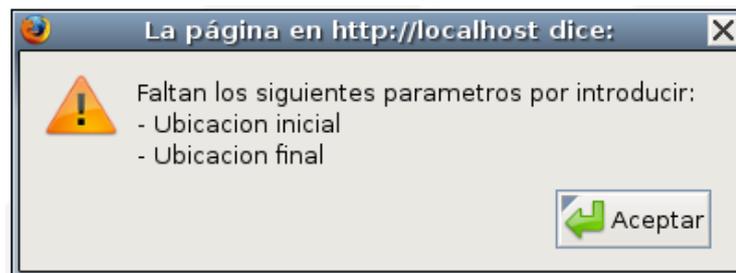


Fig. 22 – Mensaje de advertencia en informe de ubicación

Informe histórico

Consta de un formulario para indicarle el intervalo de fechas que queremos listar. Una vez introducido, aceptaremos la selección y si el sistema valida correctamente la información introducida, nos remitirá al visor de informes visualizando la información requerida. Cabe destacar que el formulario de fechas, tiene que seguir una estructura en concreto, que se encuentra definida debajo de cada uno de los campos de fecha que hay.

Fig. 23 – Formulario de informe histórico

Si no introducimos algún valor necesario en el formulario, el sistema nos mostrara un mensaje de advertencia indicándonos que campos faltan por introducir.

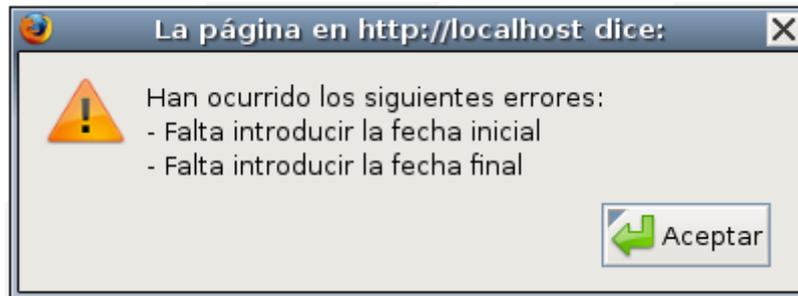


Fig. 24 – Mensaje de advertencia en informe histórico

### Informe de artículos

Encontraremos un formulario donde introduciremos la palabra clave del artículo que queremos listar. Una vez introducido, aceptaremos la selección y el sistema validara la información introducida. Si es correcta, nos remitirá al visor de informes visualizando la información requerida.

Un formulario web con el campo de texto "Nombre clave del artículo:" que contiene un cuadro de entrada vacío. Debajo de este campo hay un checkbox con el texto "Incluir salidas de material". En la parte inferior del formulario hay un botón con el texto "Buscar".

Fig. 25 – Formulario de informe de artículos

Si no introducimos la palabra clave en el formulario, el sistema nos mostrara un mensaje de advertencia indicándonos que falta este campo por introducir.

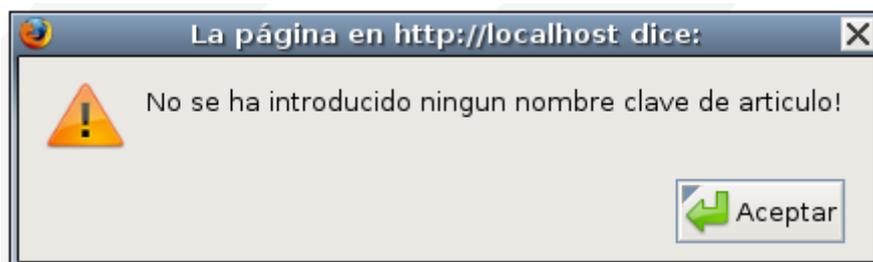


Fig. 26 – Mensaje de advertencia en informe de artículos

### Visor de informes

El diseño de este visor es muy sencillo. Tal y como hemos comentado anteriormente, consta de una página central donde se visualizara la información

solicitada desde cada uno de los diferentes informes, junto con un icono de impresión y un botón para volver a la página de informe desde donde se ha solicitado la información.

VISOR DE INFORMES - SGAR					
 Imprimir <input type="button" value="Atras"/>	INFORME UBICACIONES COMPLETO				1000/100/1 - 1000/100/6
	<b>STOCK</b>				
	<b>UBICACION</b>	<b>BULTO</b>	<b>DESCRIPCION</b>	<b>CANTIDAD</b>	<b>ALBARAN</b>
	1000/100/2	2009000001	YESO RAPIDO 25KG	5 UD	101
<b>PEDIDOS</b>					
<b>UBICACION</b>	<b>BULTO</b>	<b>DESCRIPCION</b>	<b>CANTIDAD</b>	<b>PEDIDO</b>	
	1000/100/3	2009000005	YESO RAPIDO 25KG	5 UD	00001

Fig. 27 – Visor de informes

Si los parámetros enviados por algún informe, no existen en la base de datos o son incorrectos, se mostrara un mensaje de error indicando el fallo ocurrido.

VISOR DE INFORMES - SGAR	
 Imprimir <input type="button" value="Atras"/>	ERROR: No se ha podido generar el informe debido a que los datos introducidos no son correctos o no existe ningun resultado con esos valores!

Fig. 28 – Error al generar la vista de informe

### 5.5.1.6 Mantenimiento almacén

Dentro de este módulo nos encontramos tres opciones: recuento de bulto, mover bulto y regulación de stock.

#### Recuento de bulto

Encontraremos un pequeño formulario donde deberemos introducir el número de bulto para visualizar su contenido. Una vez introducido este número de bulto, el sistema nos mostrará mediante un pequeño listado ubicado en la parte inferior de este formulario, con la relación de material que contiene este número de bulto.

Numero bulto:

**Mostrar contenido**

MATERIAL DE: **STOCK**

NUM. ALBARAN	DESCRIPCION	CANTIDAD - UBICACION
101	YESO RAPIDO 25KG	5 UD 1000/100/2

Fig. 29 – Formulario y listado de recuento de bulto

Si en el formulario no introducimos ningún número de bulto, el sistema nos mostrará un mensaje de advertencia indicándonos el error producido.

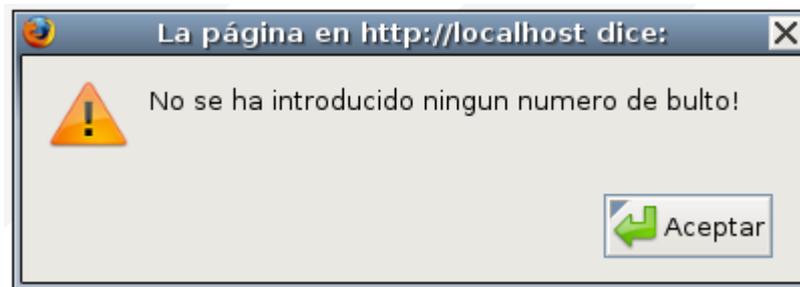


Fig. 30 – Mensaje de advertencia en recuento de bulto

Mover bulto

Se nos mostrara un formulario donde deberemos introducir la información necesaria para llevar a cabo el movimiento de bulto. Un vez introducida, el sistema nos imprimirá en la misma página un mensaje donde indicara si se ha producido correctamente o ha habido algún error en el movimiento de bulto.

Numero bulto:

Ubicacion actual:

Nueva ubicacion:

**Cambiar ubic.**

Fig. 31 – Formulario de movimiento de bulto

Si en el formulario no introducimos ningún algún parámetro necesario, el sistema nos mostrará un mensaje de advertencia indicándonos los campos que faltan por introducir para que se realice el movimiento

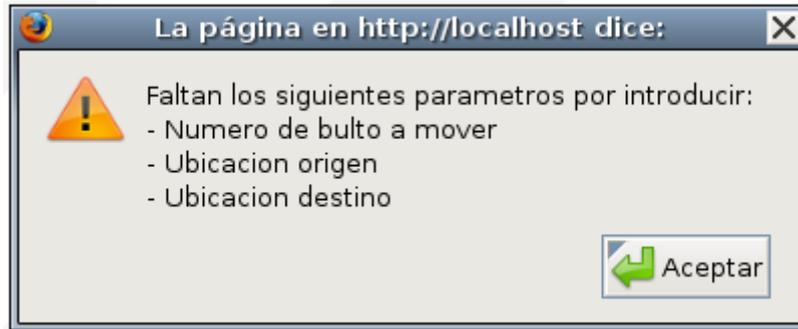


Fig. 32 – Mensaje de error por falta de parámetros en mover bulto

Regulación de stock

En esta opción encontraremos, al igual que en la opción de recuento de bulto, un pequeño formulario donde deberemos introducir el numero de bulto al que deseamos realizar una regularización de stock.

Una vez introducido y validado el bulto, se nos mostrara un listado con la relación de material de este, y en cada una de las líneas de material aparecerá un campo con la cantidad de material que consta en la base de datos. Es en este campo, donde introduciremos la cantidad de material que hemos contado en inventario, y una vez introducido podremos actualizar dicha cantidad mediante el botón “regular”.



Fig. 33 – Regularizaciones de stock

Al igual que en las demás opciones, si no introducimos un numero de bulto, el sistema nos mostrara una pantalla de advertencia indicándonos de la falta de esta información. Además, si el bulto no existe en la BBDD o ya ha sido regulado, se nos mostrara en la misma página un mensaje de error.

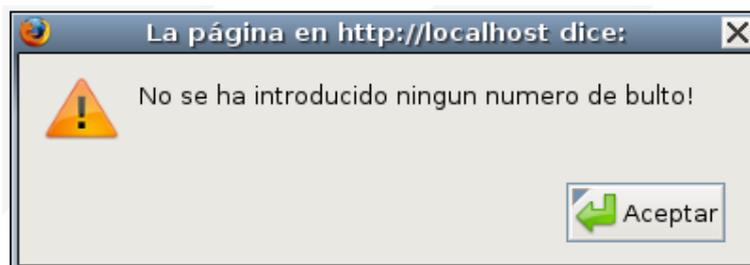


Fig. 34 – Mensaje de advertencia en regulación de stock



Fig. 35 – Mensaje de error en regulación de stock

### 5.5.1.7 Parámetros del sistema

Dentro de este módulo nos encontramos seis opciones: alta usuario, modificar usuario, alta ubicación, modificar ubicación, generar códigos e imprimir etiqueta.

#### Alta usuario

Consta de un formulario que deberemos rellenar para dar de alta al nuevo usuario en el sistema. Una vez introducidos todos los datos, el sistema validará esta información y la insertará en la base de datos.

A screenshot of a user registration form. The form contains the following fields: 'Nombre:' with the value 'Valeriano'; 'Apellidos:' with the value 'de las Morenas'; 'Nombre Usuario:' with the value 'vale'; 'Password:' with three black dots; and 'Cargo:' with a dropdown menu showing 'Encargado'. At the bottom of the form is a button labeled 'Crear'.

Fig. 36 – Formulario de alta de nuevo usuario

Como en los demás formularios, si no introducimos algún campo necesario, el sistema nos mostrará un mensaje de advertencia indicándonos que campos faltan por introducir.

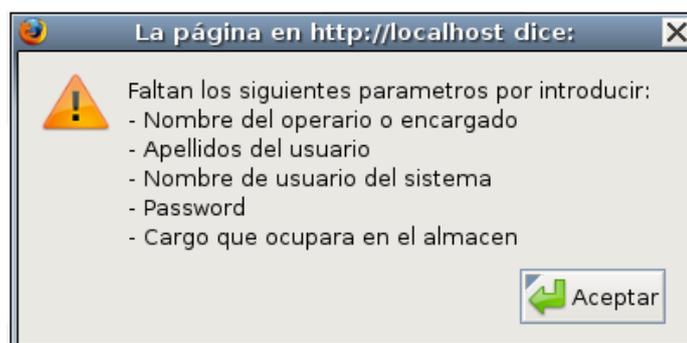


Fig. 37 – Mensaje de advertencia de formulario alta usuario

Si el usuario y password introducidos ya existen en la base datos, el programa mostrara en la misma pantalla un mensaje indicando el error.

No se ha podido crear el usuario, es posible que ya exista en la BBDD!

Fig. 38 – Mensaje de error en el formulario alta usuario

Modificar usuario

Consta de un selector de usuarios dados de alta en el sistema, donde escogeremos uno y aceptaremos. Una vez seleccionado, nos aparecerá el mismo formulario de altas de usuario, pero con los campos rellenos con la información del usuario seleccionado anteriormente.

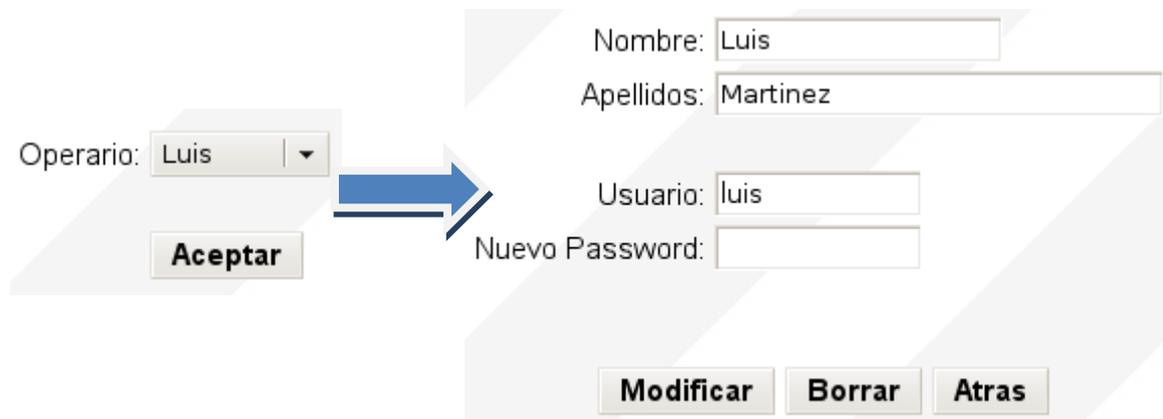


Fig. 39 – Selección de usuario y modificación de parámetros

Como en los demás formularios, si no introducimos algún campo necesario, el sistema nos mostrara un mensaje de advertencia indicándonos que campos faltan por introducir.

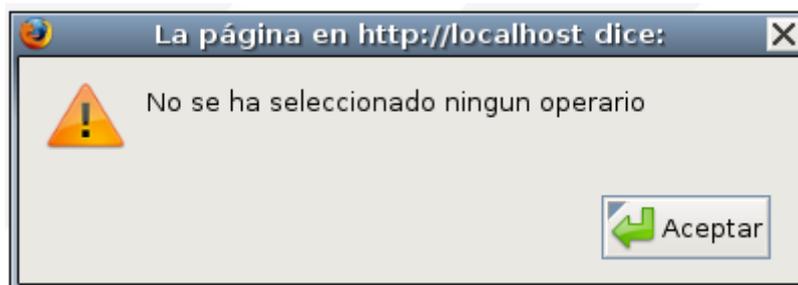


Fig. 40 – Mensaje de advertencia de formulario modificar usuario

Alta ubicación

Consta de un formulario que deberemos rellenar para dar de alta la nueva ubicación en el sistema. Una vez introducidos todos los datos, el sistema validara esta información y la insertara en la base de datos.



Formulario de alta de nueva ubicación con los siguientes campos:

- Calle: 1000
- Pila: 100
- Altura: 1
- Familia productos: ARIDOS

Botón: Crear

Fig. 41 – Formulario de alta de nueva ubicación

Como en los demás formularios, si no introducimos algún campo necesario, el sistema nos mostrara un mensaje de advertencia indicándonos que campos faltan por introducir.

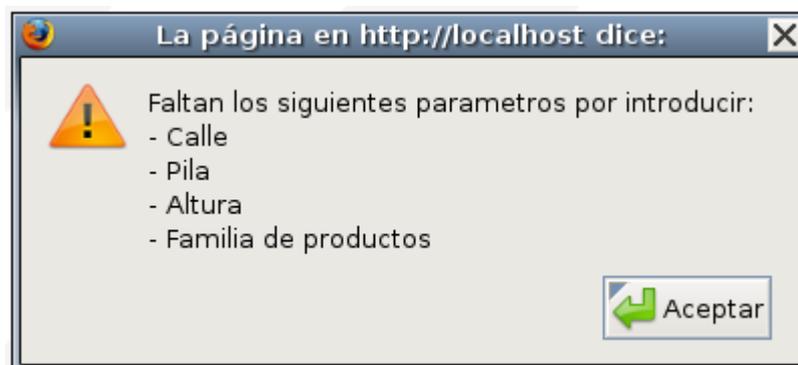


Fig. 42 – Mensaje de advertencia de formulario alta ubicación

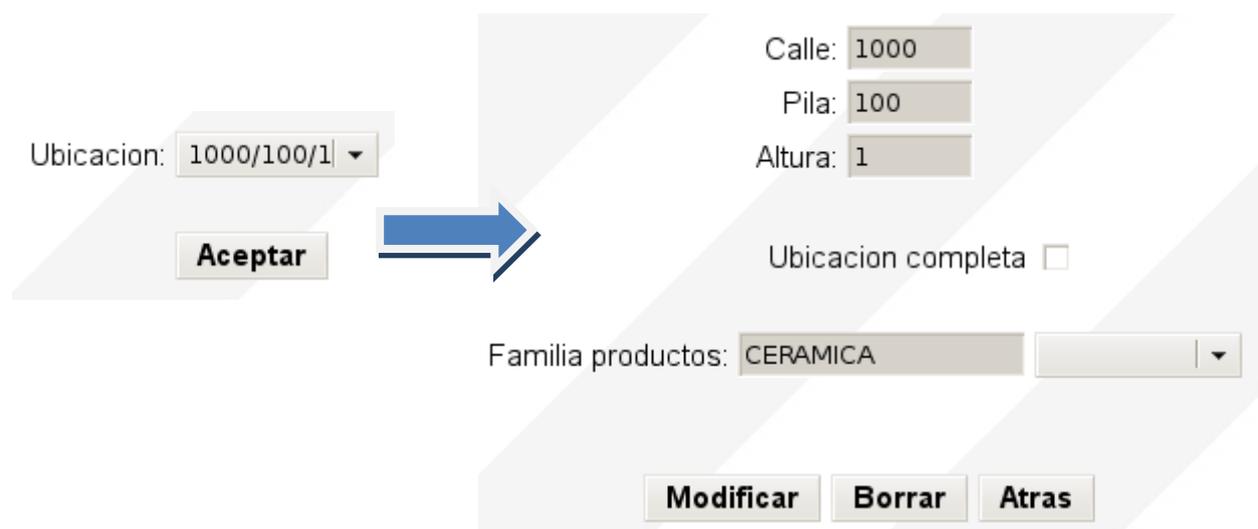
Si calle, pila y altura introducidas ya existen en la base datos, el programa mostrara en la misma pantalla un mensaje indicando el error.

La ubicacion introducida ya existe en la BBDD!

Fig. 43 – Mensaje de error en el formulario alta ubicación

### Modificar ubicación

Al igual que en la modificación de usuarios, consta de un selector de ubicaciones dadas de alta en el sistema, donde escogeremos una y aceptaremos. Una vez seleccionada, nos aparecerá el mismo formulario de altas de ubicación, pero con los campos rellenados con la información de la ubicación seleccionada anteriormente.



The image shows a web form interface. On the left, there is a dropdown menu labeled 'Ubicacion:' with the value '1000/100/1' and a button labeled 'Aceptar'. A blue arrow points from this section to the right. On the right, there are three input fields: 'Calle:' with the value '1000', 'Pila:' with the value '100', and 'Altura:' with the value '1'. Below these is a checkbox labeled 'Ubicacion completa' which is unchecked. Further down is a dropdown menu labeled 'Familia productos:' with the value 'CERAMICA'. At the bottom, there are three buttons: 'Modificar', 'Borrar', and 'Atras'.

Fig. 44 – Selección de ubicación y modificación de parámetros

Como en los demás formularios, si no introducimos algún campo necesario, el sistema nos mostrara un mensaje de advertencia indicándonos que campos faltan por introducir.

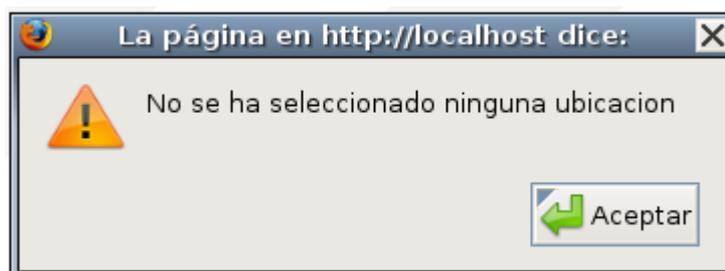


Fig. 45 – Mensaje de advertencia de formulario modificar ubicación

### Generar códigos

Consta de un pequeño formulario donde indicaremos el número de bultos que deseamos generar. Al generarse, el sistema nos informara que como ha ido el proceso.



The image shows a form with two input fields. The first is labeled 'Ultimo bulto generado:' and contains the value '2009000006'. The second is labeled 'Numero de bultos a generar:' and is empty. Below the fields is a button labeled 'Generar'.

Fig. 46 – Formulario de generación de bultos

Si no introducimos el numero de bulto que deseamos generar, el nos mostrara un mensaje de advertencia indicándonos de que falta por introducir este campo.

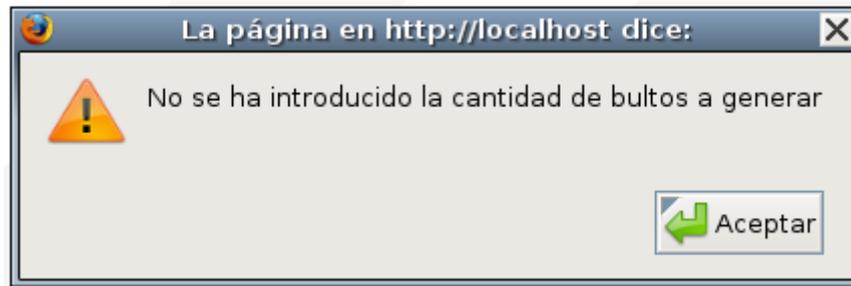


Fig. 47 – Mensaje de advertencia de formulario generar códigos

Al generarse los códigos correctamente, el sistema nos informara mediante un mensaje en esta misma página

Los bultos se han creado correctamente!

Fig. 48 – Mensaje de generación códigos de formulario generar códigos

### Imprimir etiqueta

Consta de un formulario donde deberemos introducir el intervalo de bultos o un único bulto que queramos imprimir. Una vez introducido, el sistema validara la información introducida y si es correcta nos remitirá a una página en blanco con la relación de bultos que hemos solicitado. Para poder imprimir este bulto, seleccionaremos un formato de página adecuado al tamaño de la etiqueta de impresión ya guardado en el sistema.



Fig. 49 – Formulario de impresión de etiquetas



Fig. 50 – Vista de impresión de números de bulto

Si no introducimos el numero de bulto que deseamos generar, el nos mostrara un mensaje de advertencia indicándonos de que falta por introducir este campo.

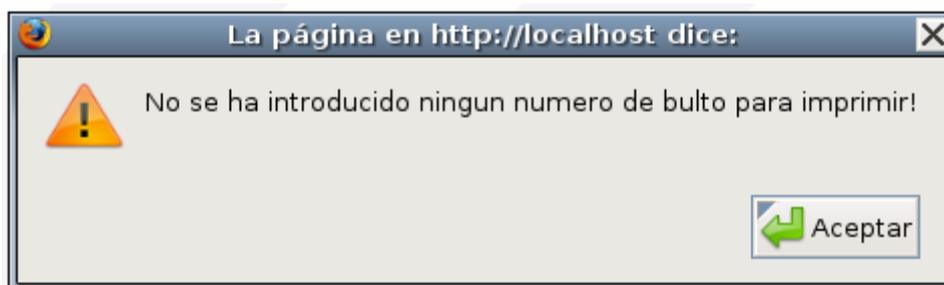


Fig. 51 – Mensaje de advertencia de formulario impresión etiquetas

Si el número de bulto introducido no existe en la base de datos, el sistema informara de ello mediante un mensaje con un fondo en color rojo, indicándonos la incidencia. También nos mostrara un link hacia la página anterior.



Fig. 52 – Mensaje de error en formulario impresión etiquetas

### 5.5.2 Interfaz para dispositivo móvil

El diseño para los dispositivos es igual que el diseño para PC, con la diferencia que se ha tenido que adaptar para que se pueda visualizar correctamente en las pantallas, con una resolución más pequeña, de los dispositivos móviles.

Otra diferencia, es que esta interfaz no tendrá acceso a todos los módulos que ofrece el programa, tan solo tendrá acceso a recuento de entradas, preparar pedido, recuento de bulto, mover bulto ubicación y regulación de stock.

También se ha simplificado bastante el menú principal para que resulte lo más fácil de utilizar y lo más intuitivo posible para los operarios. Se ha extraído elementos como la hora, fecha actual y los accesos a la agenda y al gestor de correo. Se ha utilizado un estilo muy simple con un fondo con rayas en diagonal y no se han utilizado demasiadas imágenes para no sobrecargar la red de radiofrecuencia.

Cabe destacar, que tal y como se ha diseñado la interfaz, es capaz de adecuarse a muchos tipos de resolución, ya que tan solo basta adecuar el tamaño de la ventana del navegador, y los contenidos de las páginas se centran a este tamaño.

## Capítulo 6: Implementación

### 6.1 Conexión a la base de datos

Para realizar la conexión con la base de datos y no tener que ir introduciendo el código php que realiza esta conexión en cada una de las funciones, se ha implementado un archivo php externo donde pondremos el código que realiza la conexión. Para ir llamando a esta función que realiza la conexión cada vez que la necesitemos, pondremos al inicio del script php un include del archivo externo y luego llamaremos a la función `conect_bd()` para realizar la conexión con la base de datos

A continuación se muestra el archivo externo de conexión, el include que se realiza de este archivo y la llamada a la función de conexión:

```
<?//Declaración de función del archivo de conexión
function conect_bd(){
//Declaración las variables que se utilizaran
$direccion = 'localhost';
$usuario = 'root';
$password = 'vmb';
//Nos conectamos a la BBDD
$db = mysql_connect($direccion, $usuario, $password);
// Comprobamos la conexión con el servidor
if (!$db)
{
    echo 'Error al intentar conectarse con el servidor MySQL';
    exit();
}
// Comprobamos la conexión con la BBDD en el servidor
if (!mysql_select_db("BBDD",$db))
{
    echo 'No se pudo conectar correctamente con la Base de datos';
    exit();
}
return $db;
}
?>
```

```
<?
include ("conexion.php"); // Incluimos el archive php externo de conexión
//Declaramos la función Listar ubicaciones
function list_ubic()
{
$db=conect_bd(); //Llamamos a la función para conectar con la BBDD
$qry = mysql_query("SELECT codificacion FROM `ubicaciones`");
    echo("<table align=\"center\"><tr><td><div align=\"right\"
class=\"Estilo2\">Ubicacion: </div></td>
<td><select name=\"ubic\" id=\"ubic\"><option value = \"1\"></option>");
    while($row = mysql_fetch_array($qry))
    {
        echo("<option value = \"$row[0]\">$row[0]</option>");
    }
    echo("</select></tr>");
}
?>
```

Este tipo de programación, nos facilita que si alguna vez se tiene que cambiar algún parámetro de conexión, tan solo cambiándolo en el archivo de conexión será suficiente, y no tendremos que ir función por función cambiando los parámetros modificados.

## 6.2 Autenticación y sesiones de usuarios en el sistema

La gestión de acceso de usuarios al sistema se ha realizado mediante la validación del usuario que desea entrar comprobando que esta dado de alta en la base de datos y la implementación de sesiones php para almacenar el nombre del usuario validado. Para tener un concepto más claro del proceso, se especifica a continuación cada uno de los pasos llevados a cabo:

### Autenticación

Cuando el usuario introduce el nombre de usuario y password, se realiza una llamada a una función php llamada login.php que se conecta a la base de datos y comprueba que el usuario junto con la contraseña existe. Si la información introducida existe en la base de datos, se crea un sesión php asignando el nombre del usuario validado y se le reenvía al menú principal, tal y como podemos ver en el siguiente código.

```
<?
//Incluimos el archivo php con la función de conexión a la base de datos
include ("conexion.php");
//Declaramos las variables que utilizaremos
$username = $_POST['user'];
$password = $_POST['pass'];
$pda = $_POST['pda'];
//Nos conectamos a la base de datos
$db=connect_bd();
//Realizamos la consulta para comprobar si el usuario existe en la BBDD
$sql = mysql_query("SELECT pass, nombre FROM usuarios WHERE user='$username'");
$sql2 = mysql_fetch_row($sql);
if (!$sql2)
{
    //Mostramos mensaje de error indicando que el usuario introducido es incorrecto
}
else
{
    if ($password !== $sql2[0])
    {
        //Mostramos mensaje de error indicando que el password introducido es incorrecto
    }
    // Si el usuario y password son correctos, creamos la sesión y le asignamos un nombre
    else {
        session_start();
        $_SESSION["user"] = $sql2[1];
        if($pda=='pda')
            header ("Location: ../pda/principal_pda.php"); // Redirigimos al menú del disp. móvil
        else
            header ("Location: ../web/principal.php"); // Redirigimos al menú del PC
        }
    }
mysql_close($db); //Cerramos la conexión con la BBDD
?>
```

### Sesiones php

Tal y como hemos visto anteriormente, cuando el usuario se valida, se crea la sesión para almacenar el nombre de usuario que ha iniciado la aplicación. Para utilizar esta variable que almacena el nombre del usuario, deberemos llamar al inicio de cada página que la utilizara, a la función `session_start()` que nos iniciara la sesión y mediante el código `<? echo $_SESSION["user"];?>` podremos obtener el nombre de usuario que inicio la sesión.

Para cerrar la sesión de usuario, mediante un enlace, se llamara al archivo externo `logout.php` que eliminara la sesión actual y se redirigirá a la página de login. En contenido de este archivo se muestra a continuación:

```
<?
// Declaramos las variables que llegaran por POST
$pda = $_POST['pda'];
//Realizamos la eliminación de la sesión
session_start();
session_unset();
session_destroy();
if($pda=='pda')
    header("location: ../index_pda.php"); // Redirigimos al login de los disp. móviles
else
    header("location: ../index.php"); // Redirigimos al login de la interfaz de PC
?>
```

Si una sesión ha caducado porque ya ha expirado el tiempo de duración, redirigiremos la pagina actual al login para que el usuario se vuelva a validar. Esto nos sirve como medida de seguridad por si el usuario se le olvida cerrar la aplicación, que nadie puede trabajar con su nombre de usuario. Todo esto lo realizamos mediante la implementación del siguiente código:

```
If ($_SESSION["user"]=="") // Comprobamos que no hay nombre de usuario asignado
{
    header ("Location: ../index.php"); // Redirigimos a la página de login
}
```

Este código tiene que ir a continuación de la llamada a la función `session_start()`, para que cuando inicie la sesión comprobemos si ya ha expirado y no continúe con la carga de la pagina y nos redirija al login.

### 6.3 Estructura base de las páginas

Todas las páginas de la aplicación siguen una estructura base, exceptuando el visor de informes y la visualización de los códigos de barra. La estructura también depende de la interfaz sobre la que trabajemos, ya que existe una estructura base para los dispositivos móviles y otra para la interfaz de PC. A continuación se muestra la estructura base de las páginas para cada una de las interfaces:

**INTERFAZ DE LOS DISPOSITIVOS MOVILES**

```
<? session_start(); echo '<?xml version="1.0" encoding="iso-8859-1"?>';?><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
    <script type="text/javascript" src="../../jsp/functions.js"></script>
    <? include('../../php/functions.php'); ?>
    <link rel="stylesheet" type="text/css" media="screen" href="../../css/style_pda.css"/>
    <title> Titulo de la pagina </title>
</head>
<body>
    <p align="center" class="Estilo3"> </p>
    <div align="center">Aquí indicamos en que opción nos encontramos<hr size=3 noshade>
Contenido del body. Información que cambia en cada página
    <hr size=3 noshade>
    <table class="Estilo1"><tr><td><br>Usuario actual: <strong><? echo
$_SESSION["user"];?></strong><td width="70"></td>
    <div align="center" ><td><td><br><form name="valid" id="valid" method="POST"
action="../../php/logout.php"><input type="submit" class="Estilo1" id="envia" title= "Logout" value="Cerrar
sesion"/><input name="pda" type="hidden" id="pda" value="pda"></form></td></tr></div>
    </table>
</body></html>
```

**INTERFAZ DEL PC**

```
<? session_start(); if($_SESSION["user"]=="") { header ("Location: ../../index.php"); } echo '<?xml
version="1.0" encoding="iso-8859-1"?>';?><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html
xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
    <script type="text/javascript" src="../../jsp/functions.js"></script>
    <script type="text/javascript" src="../../jsp/menu.js"></script>
    <? include('../../php/functions.php'); ?>
    <link rel="stylesheet" type="text/css" href="../../css/style.css" title="Style"/>
    <title> Titulo de la pagina </title>
</head>
<body onLoad="clk()">
    <div id="bor_der"></div>
    <div id="central"><div id="fecha">
    <table width="225" border="0">
    <tr><div align="right" class="Estilo2"><script>
    <type="text/javascript">document.write(fecha())</script></div>
    </tr></table></div><div id="reloj"><script language="javascript">document.write(clk())</script>
    </div></div>
    <div id="logo"></div>
    <div id="localizacion" align="right"> Aquí indicamos en que opción nos encontramos </div>
    <div id="barra1"></div>
    <div id="barra2"></div>
    Contenido del body. Información que cambia en cada página
    <div id="usuari" align="right" class="Estilo2">Usuario actual: <strong><? echo
$_SESSION["user"];?></strong></div>
    <div id="bor_izq"></div>
    <div id="cerrar_ses" align="center" class="Estilo2"><a href="../../php/logout.php">Cerrar
sesion</a></div>
    <div id="home"><a href="../../principal.php"></a></div>
    <div id="mail"><a href="../../webmail/index.php" target="_blank"></a></div>
    <div id="agenda"><a href="../../calendar.php" target="_blank">
    </a></div></div>
</body>
</html>
```

Tengo que destacar que estas estructuras puede sufrir alguna pequeña modificación en determinadas página, debido cambio en los accesos directos, compatibilidad entre lenguajes (problemas para visualizar mensajes ayuda de Ajax con la función de mostrar la hora de Javascript) o inclusiones de archivos externos de css, javascript o php.

## 6.4 Informes

Existen tres archivos php, los cuales generan diferentes informes. Una vez se introducen los parámetros necesarios para generar cada uno de ellos, se visualiza mediante el visor de informes. Este visor está diseñado de tal forma que sea completamente dinámico, de tal forma que con un único archivo php podemos mostrar diferentes listados de información.

Hay que tener en cuenta aspectos de esta página para que, por ejemplo, no se muestren dos informes a la vez o el botón de Atrás nos redirija a la pagina que solicito el informe. Para ello, se han ido jugando con las variables para que cuando pidamos un determinado informe, sea el que realmente se muestre.

Para entender un poco mejor el funcionamiento de este informe, se muestra a continuación el tanto el código html comentado del visor como el de las funciones php que generan los listados:

```
<? session_start(); if($_SESSION["user"]=="") { header ("Location: ../index.php"); } echo
'<?xml version="1.0" encoding="iso-8859-1"?>';?><!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html
xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
  <script type="text/javascript" src="../jsp/functions.js"></script>
  <? include ('../php/functions.php'); ?>
  <link rel="stylesheet" type="text/css" href="../css/visor_int.css"/>
  <link rel="stylesheet" type="text/css" href="../css/style.css"/>
  // Incluimos el css que omite elementos que se visualizan en la pantalla para imprimir
  <link rel="stylesheet" type="text/css" href="../css/style_imp.css" media="print"/>
  <title>SGAR: Sistema de gestion de almacenes por radiofrecuencia</title>
</head>
<body>
<div id="bor_der"></div>
<div id="central"><div id="contenedor">
  <div id="cabecera">VISOR DE INFORMES - SGAR</div>
  <div id="titulo"><strong>COPIA DE INFORME<br><br></strong></div>
  <div id="cuerpo">
  <div id="lateral">
    <br><a href="javascript:print()"></a>Imprimir<br><br>
    <? mostrar_atras(); ?>
  </div><div id="derecha">
  <div id="principal">
    <?
    list_material(); //Función que genera el listado de Artículos
    hist_ubicaciones(); //Función que genera el listado Histórico
    imp_ubicaciones(); //Función que genera el listado de Ubicaciones
    ?>
  </div></div></div></div></div>
<div id="bor_izq"></div>
</body>
</html>
```

Para evitar que nos muestre el resultado de más de un informe, ya que en el mismo archivo html llamamos a tres funciones php, pondremos una restricción mas en el *if* que realiza la comprobación. Esta restricción es, que los id's clave de los formularios de los otros informes estén vacíos, cosa que se cumplirá siempre ya que no podemos llamar a dos informes a la vez desde una misma pantalla. Con esto conseguimos que únicamente muestre el resultado de un único informe.

Para no mostrar tanto código fuente que dificulte la comprensión, se muestra un ejemplo de una única función que genera un listado, ya que las demás son similares a esta:

```
function imp_ubicaciones() // Función que genera listado de Ubicaciones
{
// Declaramos las variables que se utilizaran
// Nos conectamos a la base de datos
// Realizamos las consultas necesarias para obtener la información
// Comprobamos que los parámetros clave están en blanco y que las consultas hayan
generado resultados
if($comp==0 && $comp2==0 && ($fecha_ini==" && $nom_art=="))
{
//Mostramos mensaje de Error debido a que no hay resultados en las consultas realizadas
}
if(($comp>=1 || $comp2>=1) && ($fecha_ini==" && $nom_art=="))
{
if($checkbox !=") // Comprobamos que se ha marcado la opción de tener en cuenta
material de salida
//Mostramos información del listado en relación a material de entrada
else
//Mostramos información del listado en relación a material de pedidos
}
// Cerramos la conexión con la base de datos
}
```

Otro aspecto que debemos tener en cuenta, tal y como hemos comentado antes, es el de redireccionar a la pagina donde se genera el informe a través de un botón. Para ello, se ha implementado el siguiente código:

```
function mostrar_atras() // Declaramos la función php
{ // Declaración de variables que se utilizaran
$fecha_ini = $_POST['fecha_ini'];
$ubic_ini = $_POST['ubic_ini'];
$nom_art = $_POST['nom_art'];
// Comprobamos que se la generado el listado Histórico
if($ubic_ini=="&& $nom_art==" && $fecha_ini!=")
echo("<form method='POST' action='Listado_hist.php'><input type='submit'
value='Atras'>"); // Mostramos un botón que redirige al formulario del informe Histórico
// Comprobamos que se la generado el listado Ubicaciones
if($ubic_ini!=" && $nom_art==" && $fecha_ini==" )
echo("<form method='POST' action='Listado_ubic.php'><input type='submit'
value='Atras'>"); // Mostramos un botón que redirige al formulario del informe Ubicaciones
// Comprobamos que se la generado el listado Material
if($ubic_ini==" && $nom_art!=" && $fecha_ini==" )
echo("<form method='POST' action='Listado_articulos.php'><input type='submit'
value='Atras'>"); // Mostramos un botón que redirige al formulario del informe Material
}
```

### 6.5 Validación de formularios mediante Javascript

La validación de formularios se ya realizado mediante JavaScript. Lo que principalmente se ha comprobado es que se introduzca los parámetros necesarios en los respectivos campos de los formularios para poder llevar cabo la acción deseada. También se ha comprobado, que en el formulario de listado Histórico, se introduzca la fecha con un formato DD/MM/AAAA, para ello hemos utilizado la siguiente expresión regular que esta implementada en el siguiente código:

```
function hist_ubicacion() // Declaramos la función de JavaScript
{
// Declaramos las variables necesarias
val=0;
msg1="";
msg2="";
//Declaramos la expresión regular que utilizaremos para comprobar la fecha introducida
fecha=new RegExp(/^(0|[1-9])|(12|[0-9])3[01])(V|-)(0[1-9]|1[012])\2(\d{4})$/);
//Comprobamos la fecha introducida con la expresion regular
if (fecha.test(document.forms["histubic"].fecha_ini.value)==false)
    if(document.forms["histubic"].fecha_ini.value=="")
    {
        msg1="- Falta introducir la fecha inicial\n";
        val=1;
    }
    else
    {
        msg1="- La fecha inicial introducida es incorrecta\n";
        val=1;
    }
    // Se realice lo mismo para el otro campo de fecha del formulario
if(val==0)
    document.histubic.submit(); // Devolvemos un submit para redirigir la pagina
else
    //Mostramos una ventana con los campo que faltan por introducir
    alert("Han ocurrido los siguientes errores:\n" + msg1 + msg2);
}
```

También se pensó en mostrar un único mensaje donde se englobara todos los campos que faltaban por introducir en un formulario, tal y como podemos ver en el siguiente pseudocódigo:

```
function nombre_funcion() // Declaramos la función de JavaScript
{
val=0;
msg1="";
msg2="";
msg3="";
// Comprobamos si el campo esta vacio, si es así lo indicamos en la variable mensaje y
cambiamos el valor a la variable de control. Esto lo realizaremos para cada uno de los campos
    msg1="- Nombre de campo\n";
    val=1;
// Comprobamos el valor de la variable de control para realizar el submit o mostrar el mensaje
if(val==0)
    document.movbult.submit();
else
    alert("Faltan los siguientes parámetros por introducir:\n" + msg1 + msg2 + msg3);
}
```

## 6.6 Mensajes de ayuda mediante Ajax

Como ayuda a los usuarios de sistema, se ha implementado unos iconos de ayuda en los formularios de recuento de entradas y preparación de pedidos, para facilitar la comprensión de los campos de dichos formularios.

A continuación se muestra el código Javascript y el archivo css que se ha implementado para llevar a cabo esta funcionalidad:

```

onload=function() // Al iniciar la pagina
{
    // Accedemos a los objetos y obtenemos la información de cada uno de ellos
    cAyuda=document.getElementById("mensajesAyuda");
    cNombre=document.getElementById("ayudaTitulo");
    cTex=document.getElementById("ayudaTexto");
    divTransparente=document.getElementById("transparencia");
    divMensaje=document.getElementById("transparenciaMensaje");
    // Creamos una array donde almacenamos los distintos mensajes que existen
    ayuda=new Array();
    ayuda["Descripcion"]="Descripción del material a contar";
    ayuda["Cantidad Teorica"]="Cantidad teórica que tendría que haber físicamente";
    ayuda["Cantidad Real"]="Cantidad real recontada";
    ayuda["Ubicacion Propuesta"]="Ubicación propuesta por el sistema";
    ayuda["Ubicacion Real"]="Ubicación donde se deposita realmente el material";
    ayuda["Ubicacion Propuesta"]="Ubicación propuesta por el sistema, donde hay material de la
misma familia";
    ayuda["Bulto"]="Numero de bulto que identifica dicho material";
    ayuda["Ubicacion Completa"]="Indica si en la ubicación caben más bultos";
    ayuda["Ubicacion Depositada"]="Indicar en que ubicación se deposita el bulto preparado";
    ayuda["Cantidad Preparada"]="Introducir la cantidad de material que se ha preparado";
    ayuda["Confirmacion Bulto"]="Comprobar que el bulto donde se ha extraído material es el
correcto";
    ayuda["Bulto Pedido"]="Introducir el numero de bulto de preparación de pedido";
}

// Declaramos la función que genera un nuevo objeto Ajax
function nuevoAjax()
{
    var xmlhttp=false; // Creamos el objeto para el navegador
    xmlhttp=new XMLHttpRequest("Msxml2.XMLHTTP");
    if (!xmlhttp && typeof XMLHttpRequest!="undefined")
    {
        xmlhttp=new XMLHttpRequest();
    }
    return xmlhttp;
}

// Función que coloca el cuadro de texto con la ayuda en la pagina
function colocaAyuda(event)
{
    //Declaramos las variables coordenadas que utilizaremos
    var corX=event.clientX+window.scrollX;
    var corY=event.clientY+window.scrollY;
    // Asigamos las coordenadas X e Y donde se colocara el cuadro de texto con la
    ayudacAyuda.style.top=corY+20+"px";
    cAyuda.style.left=corX+15+"px";}

    document.removeEventListener("mouseout", ocultaAyuda, true);
}

```

```
// Función que oculta el mensaje de ayuda
function ocultaAyuda()
{
    // Ocultamos el cuadro de texto con la ayuda
    cAyuda.style.display="none";
    // Eliminamos los eventos mousemove y mouseout
    document.removeEventListener("mousemove", colocaAyuda, true);
    document.removeEventListener("mouseout", ocultaAyuda, true);
}

// Función que muestra el mensaje de ayuda
function muestraAyuda(event, campo)
{
    // Obtenemos las coordenadas de la funcion colocaAyuda
    colocaAyuda(event);
    // Añadimos los eventos mousemove y mouseout
    document.addEventListener("mousemove", colocaAyuda, true);
    document.addEventListener("mouseout", ocultaAyuda, true);
    // Mostramos el cuadro de texto cNombre.innerHTML=campo;
    cTex.innerHTML=ayuda[campo];
    cAyuda.style.display="block";
}
```

Cabe destacar que este código tan solo funciona bajo todos los navegadores excepto Internet Explorer, que utiliza otras funciones. Como la aplicación solo trabajara sobre Firefox, se ha desestimado la opción de programar para IExplorer. No se descarta, que si en un futuro hiciese falta, se podría añadir las funciones necesarias para poder visualizar este código Ajax en Explorer.

A continuación el código css utilizado para visualizar los mensajes de ayuda:

<pre>.ayuda {     width:50px;     text-align:center; }  #transparencia {     background-color:#FFFFFF;     position:absolute;     width:400px;     height:260px;     display:none;     opacity:0.95;     filter:alpha(opacity="95"); }  #mensajesAyuda {     width:150px;     font-family: Arial;     font-size:10px;     border:1px solid #000000;</pre>	<pre>position:absolute; top:0px; left:0px; display:none; text-align:center; }  #ayudaTitulo {     background-color:#000000;     color:#FFFFFF;     padding:1px; }  #ayudaTexto {     vertical-align:middle;     padding:2px; }</pre>
---	--

## 6.7 Generación de códigos de barras

La generación de códigos de barra, ha supuesto varias dificultades en su diseño, ya que el simple hecho de generarlo y visualizarlo en un navegador, hay que añadirle que se está trabajando sobre un entorno Linux, el cual no ofrece muchas alternativas sobre este tema.

De las pocas alternativas que se ofrecen, se ha escogido la implementación del modulo *php-image-barcode* de Ubuntu 8.04. Este modulo ofrece la posibilidad de trabajar con 4 tipos de códigos de barras, entre los que esta Code128, que es el formato que utilizaremos.

No explicaremos el funcionamiento del archivo php que genera la imagen, ya que es bastante complejo y nos llevaría demasiado tiempo. Lo que si se explicara es como hemos implementado este modulo en la aplicación y como generamos y obtenemos la imagen del código de barras. Para ello iremos mostrando el código que se utiliza para cada paso:

### 1. Generar de los códigos en la base de datos

```
function gener_bultos() // Función que genera los bultos a partir del formulario
{
    //Conectamos con la BBDD
    $db=connect_bd();
    //Declaramos las variables que utilizaremos
    $ult_bult = $_POST['ult_bult'];
    $generar = $_POST['bult_gen'];
    $fin=$ult_bult+$generar;
    if($generar != "")
    { //Generamos e insertamos en la tabla bultos los nuevos códigos de bulto
        for($i=($ult_bult+1);$i<($fin+1);$i++)
        {
            $ins = mysql_query("INSERT INTO `bultos` (num_bulto) VALUES ('$i');");
            echo ("<table align=\"center\"><tr><td>Los bultos se han creado
correctamente!<br><br></td></tr></table>")
        }
    }
    //Cerramos la conexión con la BBDD
    mysql_close($db);
}
```

### 2. Generar la imagen del código de barras

```
<?
    // Declaramos las variables que utilizaremos
    $num = $_GET['num'];
    //Llamamos a la función que genera los códigos de barras
    require_once("../usr/share/php/Image/Barcode.php");
    //Creamos una imagen nueva
    $bc = new Image_Barcode;
    //Dibujamos el código a partir de los valores introducidos en el formulario
    $bc->draw($num,'png');
    //Pasamos la imagen generada para poder ser visualizada
    Header("Content-type: image/png");
?>
```

### 3. Visualizar el código de barras en el navegador para la impresión

```
<? session_start(); if($_SESSION["user"]=="") { header ("Location: ../../index.php"); } echo
'<?xml version="1.0" encoding="iso-8859-1"?>';?><!DOCTYPE html PUBLIC "-
//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"><html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">

<head>
    <? include('../../php/functions.php'); ?>
    <title>IMPRESION BULTOS</title>
</head>
<body>
    <? comp_bultos(); imp_bultos(); ?>
</script>
</body>
</html>
```

#### 6.8 Hojas de estilo para la impresión de informes y dispositivos móviles

A parte de las hojas de estilo utilizadas para la web, se han implementado css para la impresión de informes y para la visualización del aplicativo en los dispositivos móviles.

La hoja de estilo que se utiliza para la visualización de la aplicación en los dispositivos móviles, es muy similar a la hoja de estilo del diseño para PC. El único cambio relevante es que se han eliminado algunas capas y las demás se han adaptado para que se visualicen correctamente.

En cambio, la hoja de estilo utilizada para la impresión de informes, sí que es distinta a las demás, ya que se introducen conceptos de ocultación de capas para que, a la hora de imprimir, no se imprima todo lo que se ve en la pantalla.

Para ello hemos tenido que añadir un nuevo campo, llamado *media*, en la llamada a la hoja de estilo que realizamos en el código de Html. Si a este campo le asignamos el valor *Print*, el navegador interpreta que esa hoja de estilo se utilizara para la imprimir capas de la interfaz que se visualiza.

Una vez ya se ha definido en el código Html que tipo de hoja de estilo utilizaremos, crearemos un nuevo archivo css, donde codificaremos las capas de esta hoja de estilo con el mismo nombre de las capas que hay en el css que visualiza la aplicación en la pantalla. En este archivo iremos marcando, con la opción *display: none* las capas que queremos que no se impriman. Una vez seleccionadas las capas que queremos que se impriman y las que nos, si realizamos una vista previa de impresión, podemos ver que los campo que hay marcados con la opción *display: none*, no se visualizaran

Este no es la única opción que nos permite este tipo de hojas de estilo, ya que si lo deseamos, podemos cambiar el tamaño y tipo de letra, la distribución de la capas, etc., tal y como se ha implementado en la hoja de estilo con nombre *style\_imp.css*, que podemos encontrar en la carpeta **css** del directorio principal de la aplicación.

## Capítulo 7: Pruebas de funcionamiento y testeo

### 7.1 Pruebas efectuadas

Durante las distintas fases del desarrollo se han ido efectuando diferentes pruebas de funcionamiento sobre la aplicación, y al finalizar toda la programación, se ha realizado una prueba más completa y exhaustiva comprobando la integración entre los módulos y su correcto funcionamiento global.

Al finalizar la programación de cada modulo, se realizaron diferentes pruebas para comprobar que dicho modulo se integraba correctamente en el aplicativo final, así factores como su acceso y tratamiento de la información de la base de datos o el control de los datos introducidos por los usuarios. Realizando estos procedimientos, nos aseguramos de que cada modulo que se integraba, funcionaba correctamente y esto nos aseguraba que en la prueba final no hubiese demasiados fallos.

Para llevar a cabo estas pruebas, se han introducido diferentes entradas en las tablas de la base de datos en relación a pedidos y albaranes de entrada ficticios así como artículos, clientes y proveedores.

Para comprobar realmente si la aplicación era funcional y fácil de usar, se realizo un simulacro de ubicación de material de entrada, preparación de pedido y una regulación de stock con un operario que gestiona material a diario y que posee pocos conocimientos de informática. Después de realizar las diversas pruebas, el operario nos comento que la aplicación le había parecido fácil de usar y que no le había supuesto demasiados problemas entender cómo funcionaba. Además nos comento que la forma de trabajar mediante la etiquetación de bultos y ubicaciones, le supuso emplear menos tiempo en la gestión de las tareas asignadas.

En resumen, se han realizado las siguientes pruebas de funcionamiento y testeo sobre la aplicación:

- Se ha comprobado la correcta visualización de la aplicación en el navegador web así como en los dispositivos móviles.
- Se ha comprobado la validación de los datos introducidos por los usuarios así como los mensajes de advertencia, ayuda o error que muestra la aplicación.
- Todas las conexiones a la base de datos a través de los módulos, la escritura y lectura de datos en los diferentes registros de cada tabla, son aspectos que también han sido comprobados.
- También se ha comprobado que se cumple correctamente el árbol de navegación diseñado e implementado así como los diferentes accesos directos a las aplicaciones externas desde cada una de las páginas del aplicativo.

- Se realizaron pruebas de acceso con usuarios dados de alta en la base de datos así como usuarios ficticios que no constaba en esta para comprobar que no se puede acceder sin estar dado de alta en la base de datos.
- Se ha comprobado que las sesiones de usuario funcionan correctamente, ya que se utilizan para registrar los procesos que realizan los operarios.
- Se han realizado pruebas para comprobar que los módulos de entrada y salida cumplían con los procesos que se han de seguir para gestionar bien el material desde que se ubica en el almacén hasta que sale de este.
- Se ha realizado una simulación con un operario, donde se ha comprobado que la facilidad de uso y la funcionalidad del programa son las adecuadas para gestionar correctamente un almacén.

Por último cabe destacar, que el modelo de desarrollo empleado ha funcionado correctamente y esto nos ha ayudado en el éxito global del proyecto.

## 7.2 Posibles ampliaciones

Durante el desarrollo de la aplicación, nos hemos ido dando cuenta de las posibles limitaciones que puede tener. Esto nos ha llevado a pensar posibles ampliaciones que pueden incrementar considerablemente las prestaciones que ofrece la aplicación.

A continuación una lista de posibles ampliaciones que se pueden llevar a cabo:

- **Ampliación del número de informes**
  - Actualmente existen tres informes de los cuales podemos extraer una determinada información. A medida que la aplicación se vaya integrando, podremos determinar que tipo de información es mas requerida por los usuarios y si esta no se puede extraer mediante los informes existentes, se podría estudiar la posibilidad de añadir más informes que contemplen dicha información necesaria.
- **Validación de formularios mediante Ajax**
  - La validación actual se lleva a cabo mediante Javascript, pero este tipo de programación esta cada vez más en desuso ya que existen otras tecnologías más avanzadas como Ajax. Este lenguaje nos permite que la página donde trabajamos no se recargue para variar algunos datos, lo que reduce muchísimo los tiempos de espera del usuario y también la cantidad de código que es preciso codificar. Además, simplifica y agiliza nuestra la aplicación cuando mostramos resultados paginados. Simplifica también la validación de un formulario ya que, al no cambiar de página, ya no es necesario recuperar los datos que ha enviado el formulario para mostrarlos nuevamente en la página de introducción de datos.

- **Optimizar interfaz de dispositivos móviles**
  - Actualmente, la interfaz de los dispositivos móviles es bastante sencilla. A medida de que se vaya implantando el sistema, se consultara con los usuarios si es necesario algún tipo de cambio, para que sea lo más eficiente posible.
  
- **Incluir mas funcionalidades en la interfaz de los dispositivos móviles**
  - Además de cambiar algunos aspectos visuales, también se contempla la opción de añadir funcionalidades como un visor de informes adecuado a estos dispositivos, la posibilidad de imprimir los albaranes desde los dispositivos, o funcionalidades que sugieran los propios usuarios
  
- **Integrar un módulo de gestión de cargas de material**
  - Sería interesante añadir la opción de gestión de cargas de material a través del programa. Añadiendo esta funcionalidad, abarcaríamos completamente las gestiones que lleva a cabo un almacén. Este modulo se encargaría de controlar las cargas que se realizan a los medios de transportes contratados, lo que nos aseguraría que, mediante el uso de bultos, las cargas se realizan correctamente.

## Capítulo 8: Conclusiones

### 8.1 Conclusiones finales

El objetivo principal de este proyecto era de abordar un problema y encontrar una solución informática que solventase dicho problema y todo ello mediante la elaboración de unos documentos, tales como el estudio de viabilidad, marco teórico, análisis, etc., y el aprendizaje de las tecnologías utilizadas que junto con los conocimientos adquiridos durante la carrera y las directrices marcadas por el director del proyecto, me han ayudado a finalizar satisfactoriamente este proyecto.

En todo momento se ha intentado que la aplicación fuese lo más efectiva y simple de utilizar posible, para que en el momento de implantarla, los usuarios se adecuen lo antes posible al nuevo sistema de trabajo. Mediante este nuevo sistema, garantizamos que la efectividad y fiabilidad en la gestión de material es mucho mayor a la que ofrece el sistema convencional mediante listado impresos.

Otro objetivo importante era aprender a utilizar y gestionar un sistema LAMP, ya que se trata de un sistema que cada vez es más utilizado y es interesante tener conocimientos sobre este. Por ello, podemos decir que la aplicación funciona correctamente en este tipo de sistemas y podemos dar por cumplido este objetivo.

Además del aprendizaje de estos sistemas, también hemos aprendido a utilizar lenguajes como Ajax, que cada vez se encuentra más extendido en el mercado, y que aunque se haya obtenido un nivel básico, esto nos ayuda a tener unos cimientos en los cuales podamos desarrollar más conocimientos sobre este lenguaje.

La planificación inicial del proyecto, no se ha cumplido en las fechas establecidas, debido principalmente a la apretada agenda laboral y académica que tengo y al aprendizaje de las tecnologías utilizadas en el desarrollo de la aplicación.

Tengo que destacar que la elaboración de este proyecto supuso un gran reto para mí, ya que nunca me había enfrentado al diseño, codificación y prueba de una aplicación. Además todos los conocimientos adquiridos en la realización de este proyecto, han supuesto para mí una motivación extra para aprender aun más cosas y no quedarme estancado con los que actualmente tengo.

Por último, me gustaría decir que estoy muy satisfecho por haber llevado a cabo este proyecto, ya que me ha servido para enfrentarme a retos que nunca me se me habían propuesto, además de adquirir nuevos conocimientos que me han ayudado a encontrar una solución al problema que me propuse al comienzo de este.

## Capítulo 9: Bibliografía

### 9.1 Relación de libros consultados

- Cesar Perez. **MySQL para Windows y Linux**. Editorial RA-MA, Año 2004. ISBN: 84-7897-601-9.
- Abraham Gutiérrez y Ginés Bravo. **PHP4 a través de ejemplos**. Editorial RA-MA, Año 2004. ISBN: 84-789-565-9.
- Justin Gehtland, Ben Galbraith y Dion Almaer. **Pragmatic Ajax**. Editorial The pragmatic programmers, Año 2006. ISBN: 0-9766940-8-5.

### 9.2 Relación de páginas Web consultadas

- **Los diferentes lenguajes de programación Web (Acceso 03/2009)**  
<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>
- **Conceptos básicos de servidores Web (Acceso 03/2009)**  
[http://www.cibernetia.com/manuales/instalacion\\_servidor\\_web/1\\_conceptos\\_basicos.php](http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php)
- **Una introducción a Apache (Acceso 03/2009)**  
[http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/)
- **¿Qué son las Bases de datos? (Acceso 03/2009)**  
<http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>
- **Carga unitaria (Acceso 03/2009)**  
<http://www.monografias.com/trabajos51/carga-unitaria/carga-unitaria3.shtml>
- **Servidor web Microsoft IIS vs. Apache (Acceso 06/2009)**  
<http://www.sahw.com/wp/archivos/2007/06/08/apache-vs-microsoft-iis-estadisticas-de-utilizacion-distribucion-y-alojamiento-de-malware/>
- **Ley Orgánica de Protección de Datos (Acceso 06/2009)**  
[http://es.wikipedia.org/wiki/Ley\\_Org%C3%A1nica\\_de\\_Protecci%C3%B3n\\_de\\_Datos\\_de\\_Car%C3%A1cter\\_Personal\\_de\\_Espa%C3%B1a](http://es.wikipedia.org/wiki/Ley_Org%C3%A1nica_de_Protecci%C3%B3n_de_Datos_de_Car%C3%A1cter_Personal_de_Espa%C3%B1a)
- **Obligaciones de la LOPD (Acceso 06/2009)**  
<http://www.leydeprotecciondedatos.com/2006/08/16/obligaciones-basicas-de-lopd/>
- **Niveles LOPD (Acceso 06/2009)**  
<http://www.portaley.com/protecciondatos/obligacioneslegales.shtml>
- **Página oficial de la LOPD (Acceso 06/2009)**  
<https://www.agpd.es/portalweb/index-ides-idphp.php>

- **Arquitectura LAMP (Acceso 06/2009)**  
<http://www.ibiblio.org/pub/linux/docs/LuCaS/Manuales-LuCAS/doc-curso-salamanca-LAMP/lamp-teoria-html/ch02s03.html>
- **RFID (Acceso 05/2009)**  
<http://es.wikipedia.org/wiki/RFID>
- **Ventajas de utilizar CSS (Acceso 06/2009)**  
<http://www.stardustxs.com/2008/03/05/ventajas-de-usar-css/>
- **Opciones para crear páginas web amigables de impresión (Acceso 06/2009)**  
<http://www.maestrosdelweb.com/editorial/opciones-para-crear-paginas-web-amigables-de-impresion/>
- **Generación de códigos de barras en Ubuntu (Acceso 06/2009)**  
<http://blog.controlzeta.net/?tag=php>
- **Validación o autenticación de usuarios en PHP (Acceso 06/2009)**  
<http://wlannot.wordpress.com/2007/03/05/validacion-o-autenticacion-de-usuarios-en-php-segunda-parte/>
- **CSS Tutorial (Acceso 05/2009)**  
<http://www.w3schools.com/css/default.asp>
- **HTML Tutorial (Acceso 05/2009)**  
<http://www.w3schools.com/html/default.asp>
- **JavaScript Tutorial (Acceso 06/2009)**  
<http://www.w3schools.com/js/default.asp>
- **Ajax Tutorial (Acceso 06/2009)**  
<http://www.w3schools.com/ajax/default.asp>

## Capítulo 10: Apéndice

### 10.1 Índice de figuras

	<b>Pág.</b>
Fig. 1 - Grado de utilización de servidores web en Internet .....	20
Fig. 2 - Diagrama de los procesos efectuados sobre los materiales .....	31
Fig. 3 - Estados posibles de un material .....	32
Fig. 4 - Diagrama arquitectura LAMP .....	39
Fig. 5 - Diagrama de la base de datos .....	40
Fig. 6 - Árbol de navegación .....	46
Fig. 7 - Menú principal de la aplicación .....	50
Fig. 8 - Listado de materiales a contar por operario.....	51
Fig. 9 - Formulario de ubicación de material .....	51
Fig. 10 - Mensaje de aviso de campos necesarios para ubicar material.....	52
Fig. 11 - Mensaje de ayuda de formulario de ubicar material .....	52
Fig. 12 - Mensaje de aviso de no asignación de tareas de recuento .....	52
Fig. 13 - Asignación de tareas de recuento de entradas a operarios.....	53
Fig. 14 - Mensaje de error por falta de parámetros de asignación.....	53
Fig. 15 - Reasignación de tareas de recuento de entrada operarios .....	54
Fig. 16 - Listado de pedidos a preparar asignados .....	54
Fig. 17 - Formulario preparar pedido .....	55
Fig. 18 - Mensaje de advertencia por falta de información .....	55
Fig. 19 - Asignación de tareas de preparación de pedidos .....	56
Fig. 20 - Reasignación de tareas de preparación de pedidos .....	56
Fig. 21 - Formulario de informe de ubicación.....	57
Fig. 22 - Mensaje de advertencia en informe de ubicación .....	57
Fig. 23 - Formulario de informe histórico .....	57
Fig. 24 - Mensaje de advertencia en informe histórico.....	58
Fig. 25 - Formulario de informe de artículos .....	58
Fig. 26 - Mensaje de advertencia en informe de artículos .....	58
Fig. 27 - Visor de informes .....	59
Fig. 28 - Error al generar la vista de informe.....	59
Fig. 29 - Formulario y listado de recuento de bulto .....	60
Fig. 30 - Mensaje de advertencia en recuento de bulto .....	60
Fig. 31 - Formulario de movimiento de bulto.....	60
Fig. 32 - Mensaje de error por falta de parámetros en mover bulto .....	61
Fig. 33 - Regularizaciones de stock .....	61
Fig. 34 - Mensaje de advertencia en regulación de stock .....	61
Fig. 35 - Mensaje de error en regulación de stock.....	62
Fig. 36 - Formulario de alta de nuevo usuario.....	62
Fig. 37 - Mensaje de advertencia de formulario alta usuario .....	62
Fig. 38 - Mensaje de error en el formulario alta usuario.....	63
Fig. 39 - Selección de usuario y modificación de parámetros .....	63
Fig. 40 - Mensaje de advertencia de formulario modificar usuario.....	63
Fig. 41 - Formulario de alta de nueva ubicación .....	64
Fig. 42 - Mensaje de advertencia de formulario alta ubicación .....	64
Fig. 43 - Mensaje de error en el formulario alta ubicación .....	64
Fig. 44 - Selección de ubicación y modificación de parámetros.....	65
Fig. 45 - Mensaje de advertencia de formulario modificar ubicación .....	65

---

Fig. 46 - Formulario de generación de bultos.....	65
Fig. 47 - Mensaje de advertencia de formulario generar códigos .....	66
Fig. 48 - Mensaje de generación códigos de formulario generar códigos....	66
Fig. 49 - Formulario de impresión de etiquetas.....	66
Fig. 50 - Vista de impresión de números de bulto.....	66
Fig. 51 - Mensaje de advertencia de formulario impresión etiquetas .....	67
Fig. 52 - Mensaje de error en formulario impresión etiquetas.....	67