



# Sublime Management

Gestionar tu empresa nunca fue tan fácil



Nombre: Marcelino Garcia Gomez

Curso: 2 DAM

Dirección: Ctra. de Jaén 17B, Albacete

Nacimiento: 22-03-2001

Edad: 21 Años

<b>DESCRIPCIÓN DEL PROYECTO</b>	<b>4</b>
OBJETIVO DEL PROYECTO	4
<b>DESCRIPCIÓN DEL ENTORNO DE TRABAJO</b>	<b>4</b>
ACTIVIDAD ECONÓMICA	4
ESTADO SOCIOECONÓMICO	4
<b>ESTUDIO DE NECESIDADES</b>	<b>5</b>
NECESIDADES PROPUESTAS POR LA EMPRESA	5
NECESIDADES EXTRAS	6
<b>RECURSOS NECESARIOS</b>	<b>6</b>
REQUISITOS HARDWARE	6
○ <i>Requisitos mínimos</i>	6
○ <i>Requisitos recomendados</i>	7
REQUISITOS SOFTWARE	7
RECURSOS HUMANOS	7
TEMPORALIZACIÓN	7
<i>Desglose de tiempo</i>	7
<i>Tiempo total</i>	8
<b>PROPUESTA TÉCNICA</b>	<b>9</b>
TECNOLOGÍAS	9
- <i>Entorno de desarrollo utilizado:</i>	9
- <i>Lenguaje de programación</i>	9
- <i>Apartado grafico</i>	9
- <i>Tecnologías de comunicaciones</i>	9
- <i>Tecnología de Almacenamiento:</i>	10
- <i>Tecnología de Reportes</i>	10
- <i>Tecnología de ayuda</i>	10
SEGURIDAD	11
- <i>Inicio de sesión</i>	11
- <i>contraseñas de usuarios</i>	11
DEMOSTRACIÓN DE TECNOLOGÍAS	12
MySQL	12
DB4O	12
Flatfla y JCalendar	12
Comunicaciones /Sendinblue	13
Reportes	14
Ayuda	14
<b>IMPLEMENTACIÓN</b>	<b>15</b>
<b>CONCLUSIONES</b>	<b>19</b>
<b>PROPUESTAS DE MEJORA</b>	<b>19</b>
<b>FUENTES UTILIZADAS</b>	<b>20</b>
<b>ANEXOS</b>	<b>20</b>
PRUEBAS Y CÓDIGO RELEVANTE	21
<i>Pruebas</i>	21
- Almacenamiento	21
- Reportes	23
- Ayuda	27



- E-Mail	28
<b>Código</b>	<b>31</b>
- Estructura de datos base	31
- Contenido de las clases	32
<b>MANUAL DE USUARIO</b>	<b>33</b>
- <b>Login</b>	<b>33</b>
- <b>Configuración</b>	<b>34</b>
o Base de Datos	35
o Apariencia	36
o Datos Empresa	37
o Usuarios	38
- <b>Almacén</b>	<b>41</b>
o Crear producto	41
o Stock	43
- <b>Cuentas</b>	<b>46</b>
o Albaranes	46
o Ventas	49
o Cliente	51
- <b>Calendario</b>	<b>52</b>
o Crear evento	52
o Evento	54
- <b>Archivo</b>	<b>54</b>
o Imprimir albaranes	54
o Imprimir clientes	55
o Imprimir proporción producto	55
o Imprimir información de un albarán	55



## Descripción del Proyecto

Nos encontramos ante un ERP (Enterprise Resource Planning – Planificación de Recursos Empresariales), que integra módulos tales como la gestión de almacén, clientes, proveedores, compra-venta, gestión de usuarios ... entre otros.

Este proyecto se adapta a las necesidades de una empresa ejemplo que mas adelante se explicaran y desarrollaran las necesidades de la misma.

### Objetivo del proyecto

Este software busca informatizar los movimientos que puede llevar a cabo cualquier empresa de compra y venta de productos, enfocado a pequeñas empresas que buscan un software fácil de utilizar e implementar.

Sin tener la necesidad de tener que dedicar horas en el funcionamiento del mismo.

Tratando de integrar todas necesidades de la empresa en un solo programa, evitando así el uso de otros softwares, que complican y ralentizan el trabajo.

## Descripción del entorno de trabajo

En este apartado vamos a describir la situación de la empresa para posteriormente ver como el proyecto se adapta a ella.

### Actividad económica

La empresa basa su actividad económica en la compra y venta de productos tanto alimentarios como utilitarios. Una de las características más destacables de esta empresa es su hora de cierre ya que sus horarios son de 9:00am hasta las 3:00 am

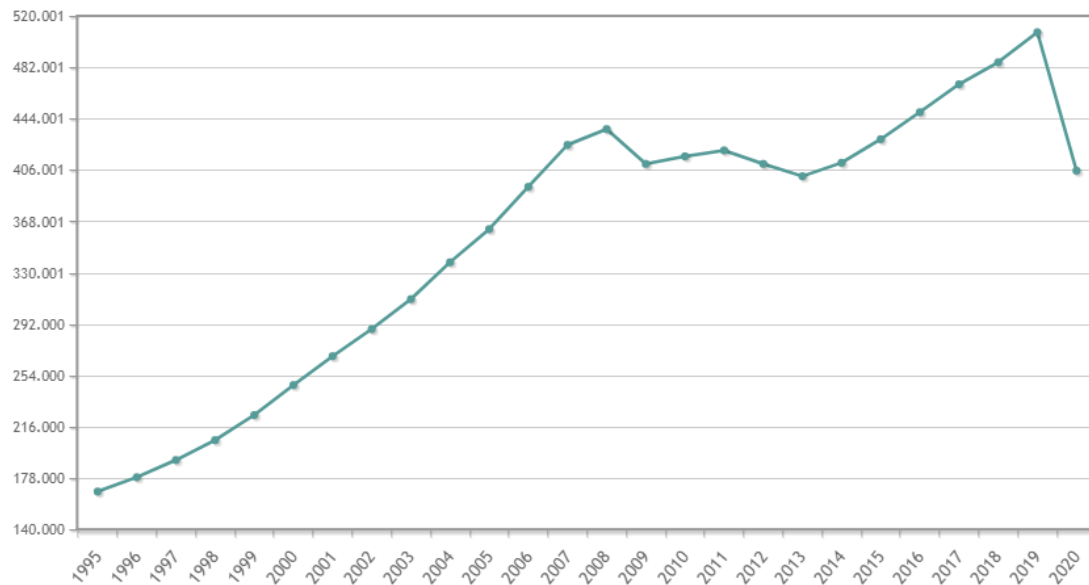
### Estado socioeconómico

Es una empresa que empieza de cero, localizada en Albacete capital, concretamente en un barrio muy transitado por lugareños, donde no hay comercios por los alrededores que ofrezcan el servicio que ofrece, y tiene un presupuesto inicial de 22000\$.

El empresario dispone de otro negocio que le aporta una media de 20000\$ al año.

A continuación, veremos una grafico que muestra el crecimiento de las empresas de comercio al por menor en España





El histórico muestra un incremento en el número de ingresos medios por cada año para la empresa de comercio al por mayor y al por menor.

Podemos ver que hay una gran bajada en 2020 debido a pandemia, no obstante, los datos de 2021 no están aún disponibles en [INE](#).

Pero según algunos periódicos, el sector se está recuperando volviendo a la subida que estaba siguiendo los años pasados.

## Estudio de necesidades

Una de las patas fundamentales de para que una empresa mejore es la disposición de un buen software que le permita administrar y digitalizar sus movimientos, de esta forma poder llegar un control sobre la empresa, según un estudio en 2019 mas del 60% de las empresas disponían de un software de gestión empresarial (ERP), por lo cual hoy en día es casi una imposición disponer de uno.

### Necesidades propuestas por la empresa

La empresa a la que va dirigido nuestro software debe de cubrir los siguientes puntos:

- Control de almacén
- Control de Entradas y salidas de productos
- Capacidad de efectuar ventas a clientes y poder imprimir ticket
- La posibilidad de realizar informes de cualquiera de los módulos anteriores
- Trabajar con un volumen de datos personales y evitar la pérdida de los mismos
- Posibilidad de trabajar con base de datos en la nube y en local
- Seguridad de acceso
- Control de usuarios
- Versatilidad a la hora de establecer los datos de Empresa (debido a futuros cambios que harán)
- Necesitan un controlador de eventos, para que puedan definir acontecimientos



### Necesidades extras

La empresa espera que nuestro programa sea familiar, es decir, que no sea necesario dedicarle mucho tiempo al aprendizaje de uso.

Por ello se usan metodologías familiares y terminología simple, en caso de tener dudas nuestro software tiene un apartado de “Ayuda” el cual posibilita al usuario de poder ver cualquier ejemplo de uso de cualquier apartado del sistema.

La empresa se encuentra en una situación de des Informatización, por lo que el objetivo que tienen es poder conectar todos los departamentos y tenerlo todo registrado en una base de datos informática y así poder estar a la altura de otras empresas de la competencia.

Según un estudio realizado anteriormente por la empresa, vio que sus fortalezas estaban en el ámbito de la cercanía con el cliente y el trato con el mismo, así como en la calidad de los productos, no obstante, una de sus mayores debilidades era la administración de grandes masas de datos ya que muchos de la información se perdían o extraviaba, por lo cual llegaron a la conclusión que necesitaban un software que pudiera gestionar todos esos datos de una forma eficiente.

Los puntos resaltados anteriormente son los solicitados por la empresa, no obstante, carece de otros apartados que ellos no sabían que necesitaban, pero a la hora de la implementación van a necesitar como son:

- Un sistema de recuperación de usuarios
- Sistema autocorregido (que ningún error impida el funcionamiento de la aplicación)
- Posibilidad de mandar E-mail a los clientes registrados con ofertas o con campañas

Todas las necesidades descritas anteriormente están cubiertas por nuestro producto, mas adelante veremos mas en detalle cada una las funcionalidades, justificando este apartado.

## Recursos necesarios

Para poder llevar a cabo el desarrollo del programa debemos de disponer del siguiente equipo:

### Requisitos hardware

- Un ordenador
  - o Requisitos mínimos:
    - 8GB de RAM 2100MHz
    - HDD 250GB
    - 4 núcleos 8Hilos a 3.2Mhz
    - Conexión a internet
    - 1 monitor Full HD (1080p) 21”



- Requisitos recomendados
  - 16GB de RAM 3200MHz
  - SSD 500GB
  - 8 núcleos 16Hilos a 3.5Mhz
  - Conexión a internet
  - 2 monitor Full HD (1080p) 24"
- Teclado / teclado mecánico
- Ratón
- Silla
- Escritorio
- USB 32GB

### Requisitos software

- Apache NetBeans 12.5 / 8.0
- Java versión 16.0 / 8.311
- Navegador web
- Windows 10 pro
- XAMPP / base de datos MySQL
- DB4O

### Recursos humanos

Para este proyecto el apartado de recursos humanos puede limitarse a un solo desarrollador, ya que en este caso no hay tiempo máximo de entrega.

También podría desarrollarse con un equipo de desarrollo, y el tiempo de entrega se vería reducido, por cada participante podría reducirse en tipo de entrega un 70%, hasta un máximo de 5 personas, ya que a partir de este número, la productividad se vería afectada

### Temporalización

#### Desglose de tiempo

El desarrollo de este proyecto estará compuesto de las siguientes fases:

- Recogida de información:
  - 10h, preguntar y las necesidades de la empresa y estudiar la actividad económica que desarrollan
- Idea Inicial:
  - 8h, Analizar las necesidades de la empresa y saber que es lo que realmente necesitan



- Planteamiento:
  - o 16h, analizar las tecnologías disponibles y cuales serias las más apropiadas para este proyecto
- Creación de Bocetos:
  - o 30h, desarrollar un conjunto de herramienta que buscan comprender mejor el futuro funcionamiento de la aplicación final.
- Desarrollo de Aplicación base:
  - o 85h, Desarrollar la base sobre la cual es desarrollara el resto del programa, de esta forma podemos tener un programa funcional que podemos mostrar a cliente y ver si es lo que necesita y quiere
- Desarrollo de la aplicación Final:
  - o 115h, Desarrollo del producto final, con todas las utilidades que tendrá finalmente y todo implementado
- Búsqueda de errores:
  - o 20h, búsqueda y reparación de “BUGS”
- Levantar la aplicación en un entorno controlado
  - o 8h, Correr nuestra aplicación sobre un sistema similar al real y ver el desempeño de la aplicación
- Implementar en software en el entorno de trabajo
  - o 10h, Implementar el producto en un entorno real
- Formación de los trabajadores
  - o 3h, explicar a los trabajadores como hacer uso del producto, así como, como deberían actuar en caso de cualquier problema no conocido

#### Tiempo total

El total de horas seria de: 305h.

Por lo cual el tiempo de desarrollo se externaría por 2 meses, para este caso en concreto ya que solo hay un desarrollador.





## Propuesta técnica

En este apartado vamos a indagar en profundidad cuales son las técnicas, tecnologías y recursos que he decidido utilizar para el desarrollo de este proyecto.

Para poder entenderlo con mayor facilidad vamos a definir el proyecto en partes:

### Tecnologías

- Entorno de desarrollo utilizado:
  - o Esta aplicación a sido completamente desarrollada en Oracle NetBeans 8.2 sobre Windows 10 pro, En este caso he optado por usar esta versión por que muchos de los plugin y addons usados en el proyecto solo funcionaban con java 8 y el IDE que más útil me resultaba era NetBeans, entre otras por la implementación de elementos “SWING”
- Lenguaje de programación
  - o Java 8.311 con el plugin de Desktop Aplicación y framework, también estaba la alternativa de usar una versión de java superior tal y como lo es la 16 o la 17, no obstante, las librerías usadas en este proyecto no eran compatibles con versiones superiores a la 8, ya que el sistema árbol de estas versiones ha cambiado, también se podría haber optado por otros leguajes de programación tal y como son C++ o Python, Se ha elegido Java ya que al utilizar una maquina virtual, en vez del sistema, hace que se reduzca el numero de errores por culpa del equipo (Drivers, Actualizaciones...)
- Apartado grafico
  - o Para los componentes gráficos he hecho uso de Java Swing y JavaBeans desarrollado por mí, así como un componente adicional sacado de GitHub ([JCalendar](#)), también estaba la alternativa de usar ScenneBuilder no obstante para esta interface gráfica, las herramientas que ofrece NetBeans por defecto son sufrientes.
  - o Para las apariencias personalizadas he usado una librería llamada “[flatlaf](#)”, siendo esta la más potente y versátil según mi experiencia, había otras alternativas como usar “Ninbus” o” JTattoo”, “flatlaf” a diferencia de estas, tiene una amplia cantidad de temas que le pueden aplicar con una sola línea de código. También incluye una herramienta llamada “[DEMO](#)” la cual nos permite tener una previa de cada uno de los temas que incluye y cómo afectaría a cada uno de los componentes
- Tecnologías de comunicaciones
  - o Como servidor de correo he usado “[Sendinblue](#)” haciendo uso de su servidor SMTP para mandar correos transicionales a través de él y usando “Java Mail”. Hay muchas alternativas de servidores SMTP como por ejemplo el propio servidor de GMAIL, no obstante, Sendinblue nos permite analizar quienes, y cuando han abierto los Mail, así como también dispone de una API para Java, Python, NodeJS... una



característica a tener en cuenta es que es que tiene la posibilidad de crear listas de clientes, así como flujos de mensajes según algún evento que ocurra en tu sistema, lo cual lo convierte en una gran opción para este proyecto.

- **Tecnología de Almacenamiento:**
  - Como almacenamiento local: DB4O, una de las alternativas mas cercanas era el uso de “SQLite” o “Derbi”, las cuales nos permiten un almacenamiento local en un archivo, pero la diferencia que me hizo optar por “DB4O” era la posibilidad de almacenar los objetos enteros sin necesidad de hacer ningún cambio de la estructura de clases, simplemente creando una instancia de la clase se podía almacenar, lo cual era más rápido que el resto de opciones
  - Como Almacenamiento En la nube: MySQL /XAMP, una de las alternativas más cercanas era Oracle siendo muy parecida a nivel de concepto, no obstante, al final opte por MySQL (aun siendo menos potente), ya que era software sobre el que mas experiencia tenia y a su vez el mas sencillo de implementar, ahorrando tiempo y recursos.
- **Tecnología de Reportes**
  - Para poder imprimir informes por pantalla he usado “[JasperReports](#)” para java, otra buena alternativa hubiera sido usar “[DynamicJasper](#)” el cual es muy parecido a JasperReports, otra alternativa seria PDFBox, pero no te aporta tantos elementos como si lo hace JasperReports, Otras de las ventajas de JasperReports es la de mostrar un PDF con el reporte creado y mostrarlo en un visor que trae incorporado, dando la opción de verlo antes de imprimirlo por ello está ha sido a la opción elegida para el desarrollo de la app.
- **Tecnología de ayuda**
  - Para el apartado de ayuda he usado “[JavaHelp](#)”, una de las alternativas mas cercanas a esta era el uso de los propios elementos de swing, pero para ahorra recursos y tiempo se ha decidido hacer uso de esta clase, por otro lado usar esta clase, nos aporta una serie de ventajas que su alternativa no, como puede ser el uso de un buscador por similitudes y de un menú en forma de árbol.  
Este tipo de menú es mas familiar para los usuarios ya que se usan en todo tipo de aplicaciones.



## Seguridad

Respecto al apartado de seguridad podemos dividirlo en dos:

- Inicio de sesión
  - Para poder ingresar en el programa debemos tener una cuenta de usuario en el sistema, y esta cuenta debe estar asociada con un correo real para poder recuperar la contraseña en caso de olvidarla.
  - Cuando iniciamos el sistema por primera vez tenemos un usuario con credenciales por defecto: Correo: [admin@admin.com](mailto:admin@admin.com) y una contraseña: “admin”, nos pedirá cambiar la contraseña una vez ingresamos por primera vez
  - En caso de iniciar sesión con un usuario y no recordar la contraseña podemos mandarla al correo con el que tenemos asociada la cuenta, este correo no mostrará una contraseña, que, al introducirla en el sistema, nos pedirá cambiarla por otra personal y privada.
- contraseñas de usuarios
  - Cuando estamos dentro del sistema podemos ver las cuentas de los usuarios que hay en el sistema, no obstante, las contraseñas no podremos verlas a simple vista ya que, en caso de querer verla nos mandará un correo al respectivo correo con la contraseña.
  - También podemos crear usuarios desde dentro del programa simplemente con una cuenta de correo y una contraseña.

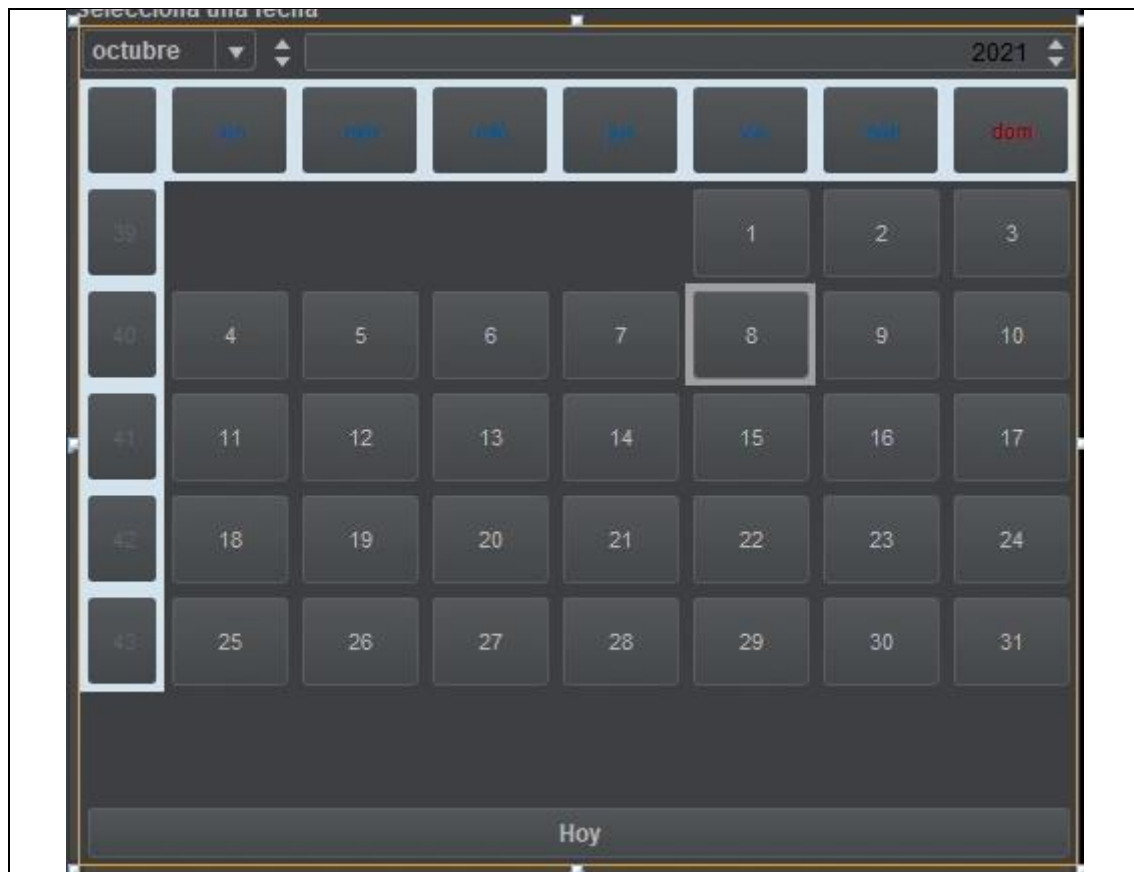


## Demostración de tecnologías

A modo de demostración veremos unos ejemplos reales de nuestra aplicación, viendo como esta implementada la tecnología anteriormente descrita.

MySQL	
<pre> public static void crearBDD() throws SQLException {     String borrarBDD = "drop DATABASE IF EXISTS ~" + nombreBDD + "~";     String crearBDD = "CREATE DATABASE IF NOT EXISTS ~" + nombreBDD + "~";     Connection conexion;     conexion = DriverManager.getConnection(rutaSQL, usuarioBDD, passBDD);     Statement sentencia = conexion.createStatement();     if (servicio) {         sentencia.executeUpdate(borrarBDD);     }     sentencia.executeUpdate(crearBDD);     conexion.close(); } </pre>	<pre> //sentencia para Elimina la base de datos //sentencia para Crea la base de datos //crea una instancia de la base de datos //crea un objeto para hacer consultas //borra la base de datos //crea la base de datos </pre>
DB4O	
<pre> public static void guardarEnLocal() {     String LocalBDD = "Sublime_Management.db";     if (db == null) {         System.out.println("no se ha abierto la base de datos local");     } else {         try {             db.close();             Path path = Paths.get("./Sublime_Management.db");             Files.delete(path);         } catch (Exception e) {             System.out.println("No se ha eliminado le archivo local: " + e.getMessage());         }         //guardarClientes();         //guardarProductos();         //guardarAlabranes();          db = Db4oEmbedded.openFile(Db4oEmbedded.newConfiguration(), LocalBDD);         db.store(new Datos());     }     System.out.println("==Guardado en Local=="); } </pre>	<pre> //establece el nombre de la BDD local //cierra cualquier conexion en local //Elimina el archivo local //crea el archivo de base de datos //Guarda una instancia de las clases </pre>
Flatfla y JCalendar	
<pre> public static void AplicarTema(int op){     switch (op){         case 0:             com.formdev.flatlaf.intellijthemes.FlatArcDarkIJTheme.setup();             UIManager.put( "TextComponent.arc", 999 );             UIManager.put( "Button.arc", 999 );             UIManager.put( "Component.arc", 999 );             UIManager.put( "ScrollBar.thumbArc", 999 );             break;         case 1:             com.formdev.flatlaf.intellijthemes.FlatArcDarkOrangeIJTheme.setup();             break;         case 2:             com.formdev.flatlaf.intellijthemes.FlatDarkFlatIJTheme.setup();             break;         case 3:             com.formdev.flatlaf.FlatLightLaf.setup();             break;         case 4:             com.formdev.flatlaf.intellijthemes.FlatOneDarkIJTheme.setup();             break;         case 5:             com.formdev.flatlaf.intellijthemes.materialthemeuilite.FlatMoonlightContrastIJTheme.setup();             break;         case 6:             com.formdev.flatlaf.intellijthemes.materialthemeuilite.FlatMonokaiProIJTheme.setup();             break;         case 7:             com.formdev.flatlaf.intellijthemes.materialthemeuilite.FlatMaterialDeepOceanContrastIJTheme.setup();             break;     } } </pre>	<pre> //aplica el tema //redondea textField //redondea botones //aplica angulos //reondea Scrollbar </pre>





## Comunicaciones /Sendinblue

```

public static void SendEmailRecoverPass(final String correo, final String TempPass) {
    final String HOST_NAME = "smtp-relay.sendinblue.com"; //direccion del servidor SMTP
    final String USER_NAME = "marce3secundaria@gmail.com"; //Cuenta de correo que manda los mail
    final String PASSWORD = "nP4Kt1gbDx0VXj5p"; //contraseña proporcionada por el servidor
    final String TO = correo; //receptor
    final Properties props = new Properties(); //creamos una instancia de las propiedades
    props.put("mail.smtp.host", HOST_NAME); //configurar el servidor
    props.put("mail.smtp.auth", "true"); //autenticar?
    props.put("mail.smtp.port", "587"); //puerto del servidor SMTP
    final Session session;
    session = Session.getDefaultInstance(props,
        (Authenticator)new Authenticator() { //crear una sesion
            protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication("marce3secundaria@gmail.com", //iniciar sesion
                    "nP4Kt1gbDx0VXj5p");
            }
        });
    try {
        final MimeMessage message = new MimeMessage(session); //creamos un mensaje con la configuracion de antes
        message.setFrom((Address)new InternetAddress("sublimeMagement@sublimeMagement.com")); //nombre del mail emisor
        message.addRecipient(Message.RecipientType.TO, (Address)new InternetAddress(TO));
        message.setSubject("Contrase\u00f1a cambiada en su usuario"); //asunto
        //cuerpo
        message.setText("<h1 style='text-align: center;'>Cambio de Contrase\u00f1a Solicitado</h1>\n<p style='text-align: center;'>Se le manda un correo informativo, \"Informacion\", 1);
        Transport.send((Message)message); //enviar mensaje
        JOptionPane.showMessageDialog(null, "Se le manda un correo informativo", "Informacion", 1);
    }
    catch (MessagingException e) {
        e.printStackTrace();
    }
}

```





## Reportes

```

public static void reporte1(){
    int seleccion;
    Connection conn = null;

    // Opciones del menú principal
    String[] opciones = {"<html>1.- Generar informe de<br>/>Todos los albaranes", "<html>2.- Salir"};
    // El menú principal se repite, mientras no pulse SALIR
    seleccion = JOptionPane.showOptionDialog(null, "Seleccione un opción", "Menú de opciones", JOptionPane.CLOSED_OPTION, JOptionPane.PLAIN_MESSAGE, null, opciones, null);
    switch (seleccion) {
        case 0:
            try {
                Class.forName("com.mysql.jdbc.Driver");
                conn = DriverManager.getConnection(SqlManager.rutaBD0, SqlManager.usuarioBD0, SqlManager.passBD0); //esto hay que cambiarlo si queremos que funcionen

                //especificamos sobre que jasper vamos a trabajar
                JasperReport reporte = (JasperReport) JRLoader.loadObjectFromFile("src/reportes/report1.jasper");
                //creamos el print
                JasperPrint JasperPrint = JasperFillManager.fillReport(reporte, null, conn);
                //objeto exportador a pdf
                Exporter exporter = new JRPdfExporter();
                //agregamos cual sera la entrada en este caso el print
                exporter.setExporterInput(new SimpleExporterInput(JasperPrint));
                //agregamos la salida
                exporter.setExporterOutput(new SimpleOutputStreamExporterOutput("./miReporte1.pdf"));
                //extraemos le pdf
                exporter.exportReport();
                //permite ver desde la propia app
                JasperViewer.viewReport(JasperPrint, false);
            } catch (ClassNotFoundException ex) {
                System.out.println("algo salio mal: "+ex.getLocalizedMessage());
            } catch (JRException ex) {
                Logger.getLogger(GestionReportes.class.getName()).log(Level.SEVERE, null, ex);
            } catch (SQLException ex) {
                JOptionPane.showMessageDialog(null, "Para imprimir los informes se necesita una base de datos online configurada", "Error", JOptionPane.WARNING_MESSAGE);
                Logger.getLogger(GestionReportes.class.getName()).log(Level.SEVERE, null, ex);
            }
            break;
    }
}

```

## Ayuda

```

private void ponLaAyuda() {
    try {
        // Carga el fichero de ayuda
        File fichero = new File("./help/help_set.hs");
        URL hsURL = fichero.toURI().toURL();

        // Crea el HelpSet y el HelpBroker
        HelpSet helpset = new HelpSet(getClass().getClassLoader(), hsURL);
        helpset.setTitle("Ayuda SublimeManagement");
        HelpBroker hb = helpset.createHelpBroker();

        // Pone ayuda al pulsar el botón y a F1 en la ventana.
        hb.enableHelpOnButton(miAyuda, "aplicacion", helpset);
        hb.enableHelpKey(this.getContentPane(), "aplicacion", helpset);
        hb.setSize(new Dimension(1650, 900));

    } catch (IllegalArgumentException e) {
        e.printStackTrace();
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (HelpSetException e) {
        e.printStackTrace();
    }
}

```




## Implementación

Para el correcto funcionamiento de este software vamos a numerar los requisitos hardware que necesitamos, así como software:

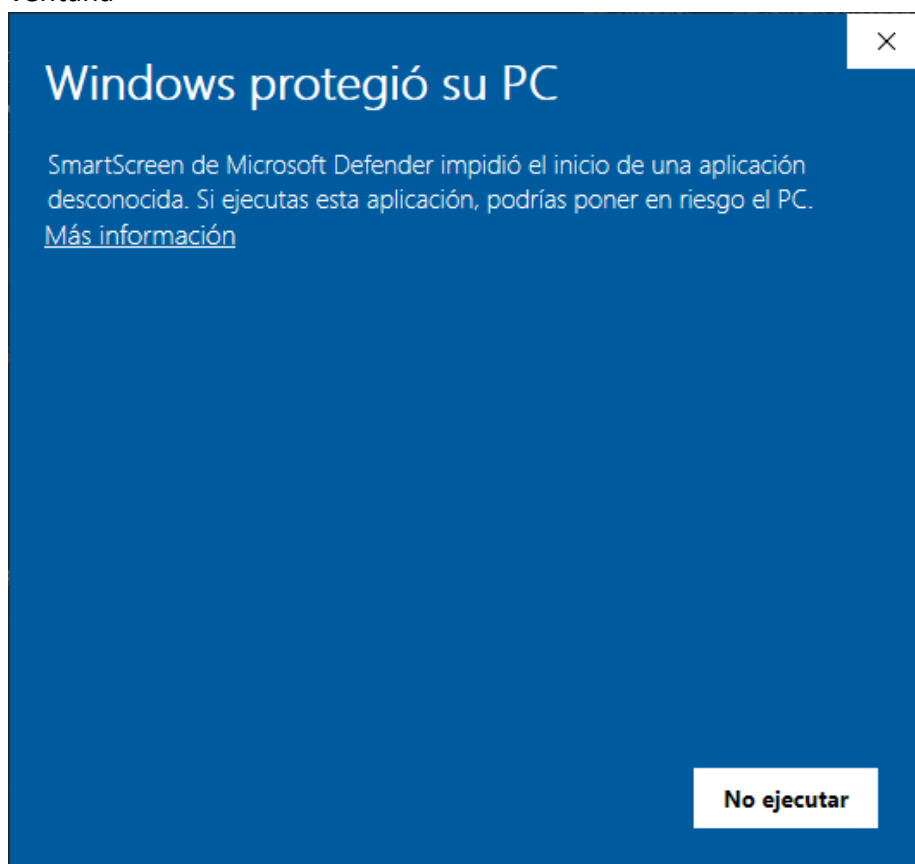
- recursos mínimos
  - Un equipo con Windows 7 o superior, con java 8 instalado
  - Conexión a internet
- Recursos recomendados (para sacarle el máximo rendimiento):
  - Un equipo o varios con Windows 10 y java 8 instalado
  - Conexión a internet
  - Un servidor MySQL al que poder conectarnos

A continuación, viene la explicación de instalación y pues en funcionamiento del programa:

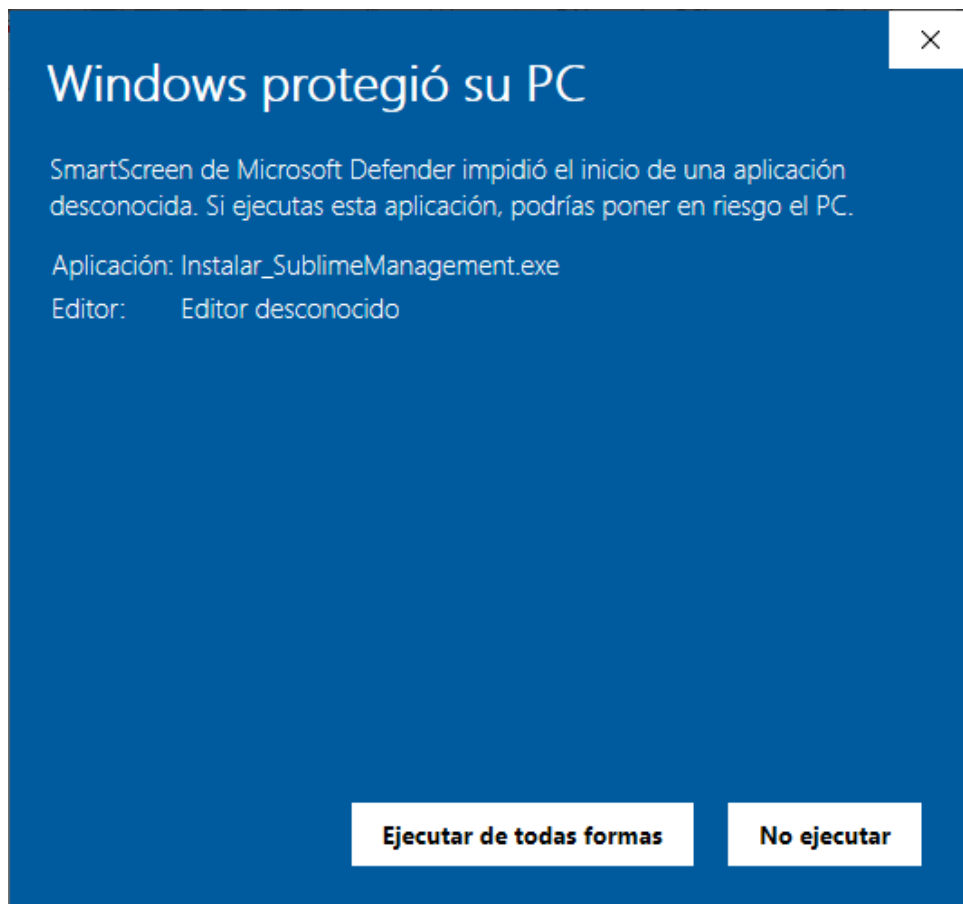
1. Copiar el instalador en el equipo donde vamos a instalar el programa

 Instalar_SublimeManagement.exe	09/06/2022 10:47	Aplicación	85.766 KB
--	------------------	------------	-----------

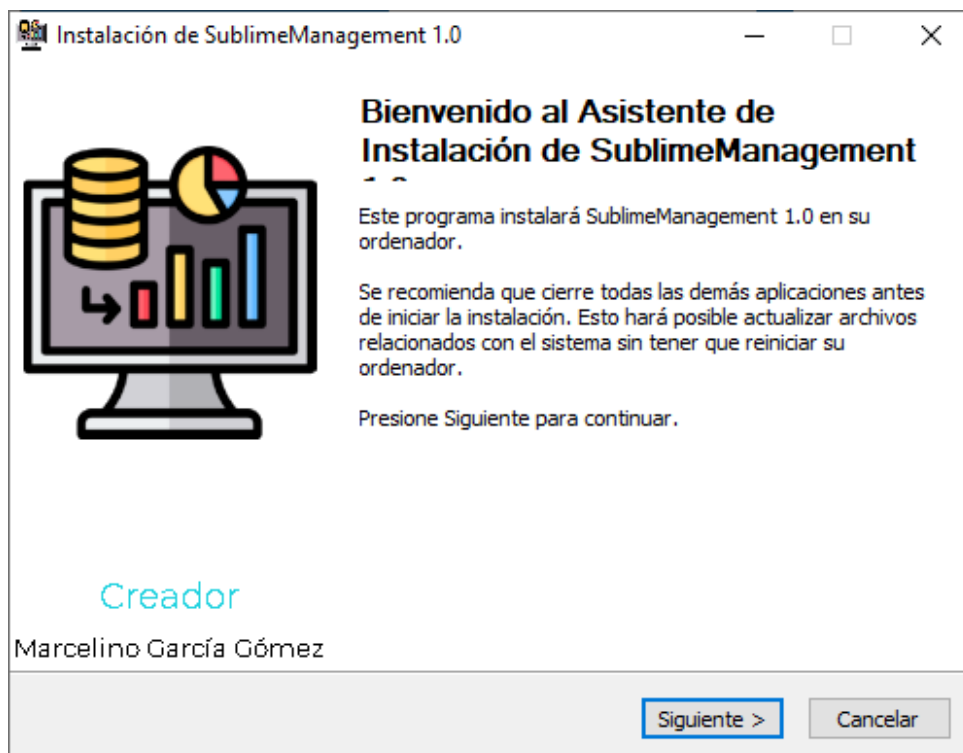
2. Ahora debemos hacerle doble clic y es posible que nos salga la siguiente ventana



3. En caso de ser así, le damos a “Mas información” y a “Ejecutar de todas formas”

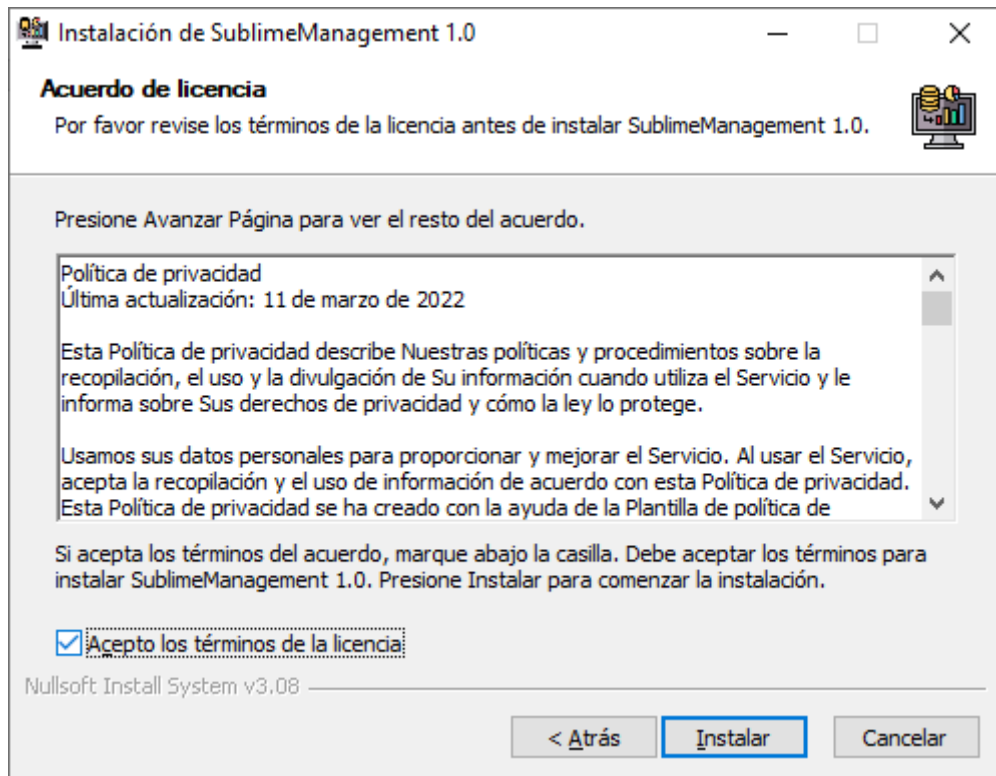


4. Nos pedirá permisos de administrador, le damos que si ya tendremos el instalador abierto

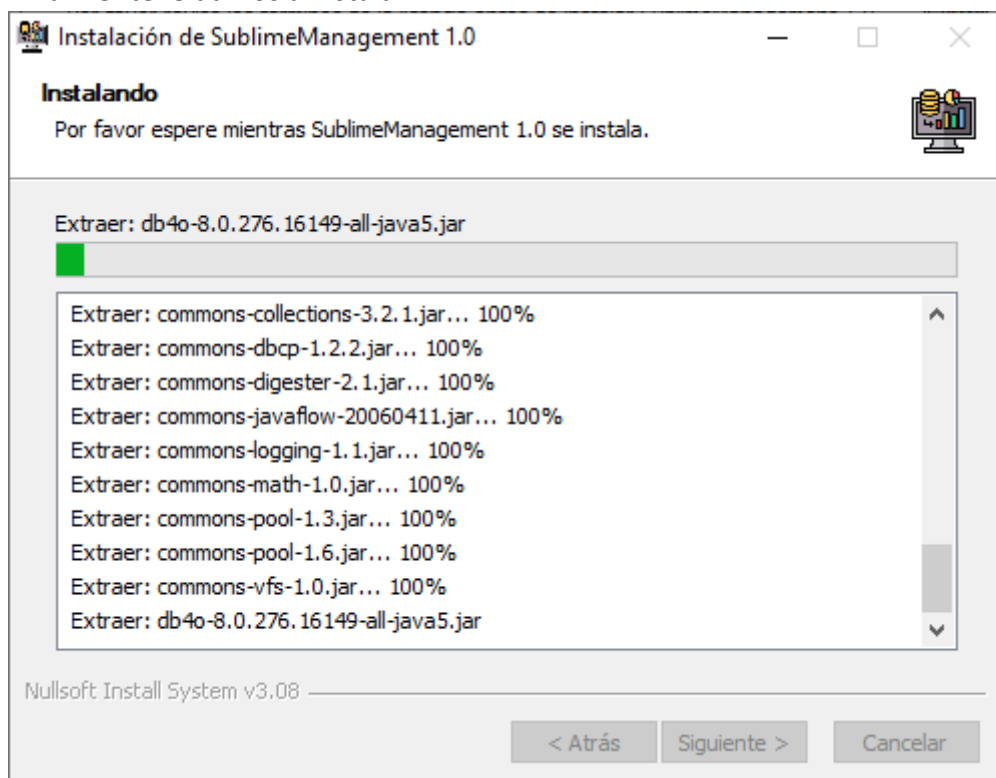




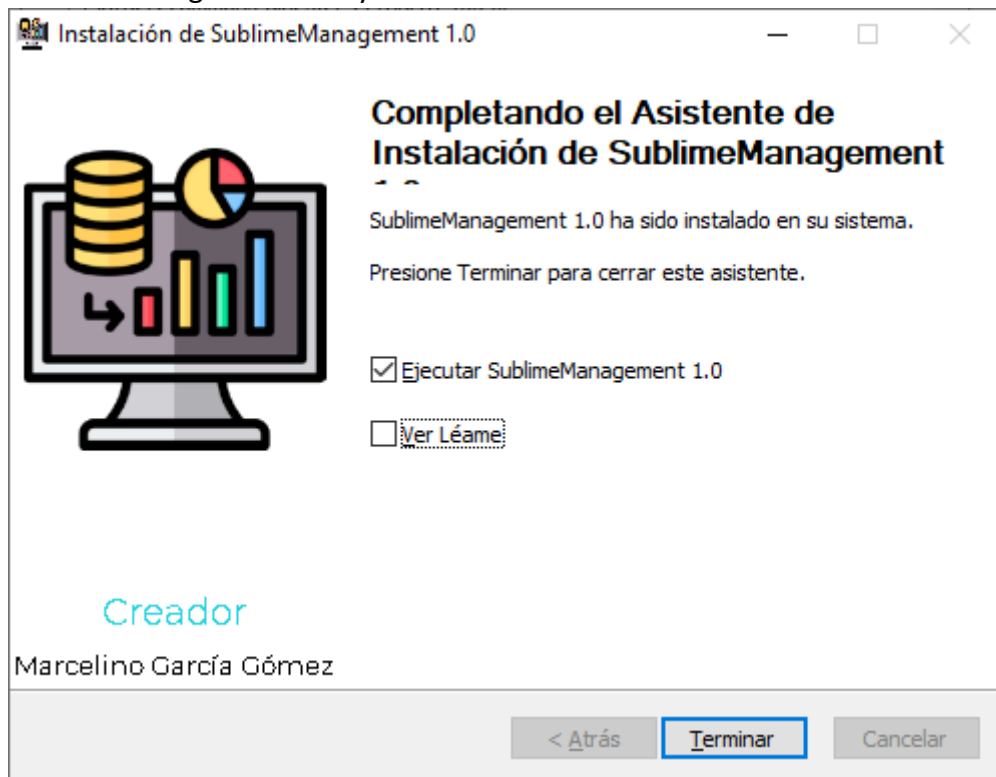
5. Le damos a siguiente y nos llevara a esta otra ventana donde debemos aceptar los términos



6. Finalmente le damos a instalar



7. Una vez finaliza podemos ejecutar el programa y ver el archivo “ReadMe.txt”, en este caso vamos a dejar marcada la opción de “Ejecutar SublimeManagement 1.0” y le damos a terminar



8. En caso de que nos salga esta ventana, buscamos java y le damos a aceptar

¿Cómo quieres abrir este archivo?

Seguir usando esta aplicación



Java(TM) Platform SE binary

Otras opciones



WinRAR archiver



Buscar una aplicación en Microsoft Store

Más aplicaciones ↓

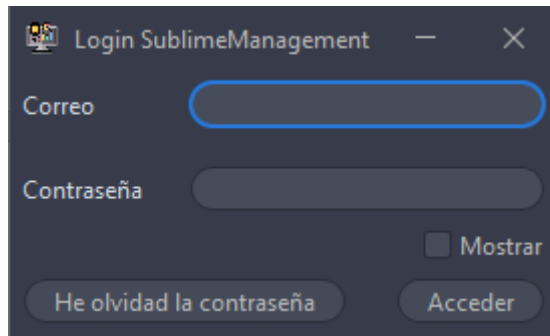


Usar siempre esta aplicación para abrir los archivos .jar

Aceptar

9. Ya tenemos el programa abierto listo para introducir el usuario





## Conclusiones

La realización de este producto me ha ayudado a tomar buenas costumbres a la hora de desarrollar cualquier aplicación futura.

Me ha hecho ver el potencial que hay detrás de la programación y de las horas de trabajo, este proyecto puede resultar útil para cualquier usuario que necesite gestionar una pequeña cantidad de productos, y cualquiera pueda acceder a ella ya que es de código abierto.

La decisión de que fuera de código libre no estaba desde el inicio de proyecto, pero si no llega a ser por la ayuda y los recursos gratuitos que hay en la red, no podría haber logrado este proyecto, por ello en la última hora se ha decidido hacerlo de código libre, para que cualquier persona que necesite de él pueda acceder sin ningún problema.

Gracias a este proyecto ahora tengo una metodología más clara y una forma de trabajar más eficiente, pensando el siguiente paso antes de lanzarme a desarrollar.

## Propuestas de mejora

Durante el desarrollo de este proyecto he tenido muchas propuestas de mejora las cuales no he podido implementar por falta de tiempo, a continuación, expongo un listado de las mejoras que le haría al programa:

- Apartado de facturación  
Al igual que tenemos un apartado de ventas, un apartado de facturación sería útil a la hora de que las empresas pidan facturas a su nombre para así poder declarar con mayor facilidad, ya que el apartado de ventas hace un reporte de la venta, pero no expone el nombre ni los datos de la empresa
- Apartado con todos los correos de los clientes  
Esto sería útil para poder mandar correos de forma masiva a todos los clientes, esto le aportaría un beneficio por ser clientes, ya que se le podrían mandar ofertas o promociones, o simplemente tener informados nuestros clientes con mensajes periódicos
- Mandar mensajes de bienvenida a todos los nuevos clientes



A la hora de que unos clientes se de alta en nuestra base de datos le mantenemos informado mandado le un mail de confirmación

- Cambiar el lenguaje de la aplicación  
En caso de que nuestra aplicación crezca debemos darle posibilidad al usuario de cambiar el idioma del programa, y así mejorar la calidad del producto.
- Multiusuario  
Un gran punto que se puede desarrollar es el de hacer nuestra aplicación multiusuario, para que múltiples usuarios puedan trabajar con el programa a la vez
- Hacer un apartado de correos  
Esto sería muy útil para mandar correos a los clientes o proveedores y evitar salinos de la aplicación.
- Eventos inteligentes  
El apartado de los eventos debería poder avisarnos aun que programa no este abierto, por lo que una mejora seria crear una aplicación móvil que pudiera avisarnos de los eventos que definimos en programa.

## Fuentes utilizadas

<http://dynamicjasper.com/>

<https://stackoverflow.com/>

<https://docs.oracle.com/cd/E19253-01/819-0913/release/libraries.html>

[https://academic.uprm.edu/jhuerta/HTMLobj-223/Estudio\\_de\\_Necesidades.pdf](https://academic.uprm.edu/jhuerta/HTMLobj-223/Estudio_de_Necesidades.pdf)

<https://developers.sendinblue.com/recipes/create-or-update-a-contact-in-python>

<https://developers.sendinblue.com/reference/sendtransacemail>

<https://ranking-empresas.eleconomista.es/>

<https://docs.oracle.com/javase/8/docs/api/>

<http://www.java2s.com/>

[https://www.tutorialspoint.com/java/java\\_sending\\_email.htm](https://www.tutorialspoint.com/java/java_sending_email.htm)

<https://ine.es/jaxiT3/Datos.htm?t=32449#!tabs-grafico>

## Anexos

Este este conjunto de apartados vamos a ver con mayor profundidad el funcionamiento del software para poder comprender mejor como están implementadas cada una de las partes descritas anteriormente



## Pruebas y código relevante

Este será un pequeño resumen de todas las pruebas que se ha realizado a lo largo del proyecto y como se han llevado a cabo, así como explicar el código más importante de nuestra aplicación

### Pruebas

Las pruebas realizadas pueden ser divididas según el tipo de tecnología testada por ello vamos a dividir este apartado:

#### - Almacenamiento

- Esta tecnología debería de permitir activar y desactivar cualquiera de las dos opciones es decir podemos tener:

- Base de datos local activada en solitario

- Estado actual: local activada, online activada; creamos un registro (un producto)

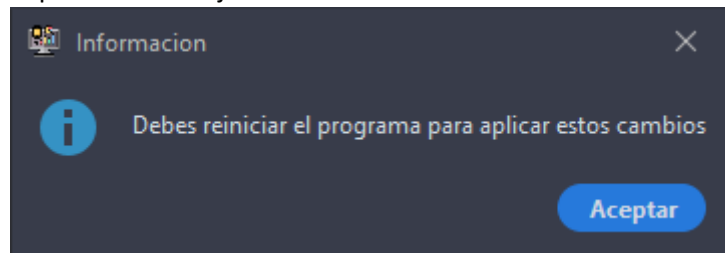
	Nombre	Referencia	Pvp
Ejemplo	ref	2.12	

- Procedemos a desactivar la base de datos online

**Habilitar base de datos local:** ☒

**Habilitar base de datos Online:** ☐

- Ahora aplicamos y reiniciamos la aplicación tal y como lo expresa el mensaje informativo



- Iniciamos el programa y comprobamos que el programa a cargado el registro anteriormente creado

	Nombre	Referencia	Pvp
Ejemplo	ref	2.12	

- Estado: local activa, online desactivada; Ahora debemos crear un nuevo registro

	Nombre	Referencia	Pvp
Ejemplo	ref	2.12	
Ejemplo2	ref2	3.00	

- Ahora comprobamos que no se ha sincronizado en la base de datos MySQL (Online)

**+ Opciones**

← T → ▼ ref nombre pvp cantidad

☐ Editar Copiar Borrar ref Ejemplo 2.12 23

↑ ☐ Seleccionar todo Para los elementos que están marcados:

- Conclusión: El resultado es el esperado solo está el producto previo a reiniciar el programa, este módulo funciona correctamente



- Base de datos online (MySQL) activada en solitario
  - Estado: local activada, online desactivada, viniendo de la prueba anterior, reiniciamos, ahora vamos a dirigirnos a configuración y a activar la base de datos online y a desactivar la base de datos local

Habilitar base de datos local: ☐

Habilitar base de datos Online: ☒

Dirección: LocalHost

Nombre Base Datos: sublime\_management

Puerto: 3306

Usuario: root

Contraseña:

Probar Conexión Mostrar ☐

- Reiniciamos la aplicación y comprobamos que la base de datos local a cargado correctamente, solo un registro, ya que la base de datos local no cargo su segundo registro en MySQL

ref	Nombre	Ejemplo	Referencia	Pvp
ref		Ejemplo		2.12

- Ahora vamos a crear un registro nuevo

ref	Nombre	Ejemplo	Referencia	Pvp
ref		Ejemplo		2.12
Ejemplo3		ref3		4.22

- Nos fijamos que en MySQL ahora hay un registro nuevo, lo cual indica que guarda bien

+ Opciones

	ref	nombre	pvp	cantidad
<input type="checkbox"/> Editar  Copiar  Borrar	Ejemplo	ref	2.12	23
<input type="checkbox"/> Editar  Copiar  Borrar	ref3	Ejemplo3	4.22	1

- Conclusión: la base de datos online guarda y carga correctamente (algo a tener en cuenta es que la base de datos local guarda cuando cerramos la app por lo cual si estamos usando la online y activamos la base de datos local, cuando cerremos la app e iniciemos nuestros datos de la base de datos online se cargaran en la loca y podemos trabajar con los datos actuales de MySQL)

- Ambas bases de datos conectadas a la vez

- Estado: online activa, local desactivado; ahora vamos a ir a configuración y a activar la opción de local

Habilitar base de datos local: ☒

Habilitar base de datos Online: ☒

Dirección: LocalHost

Nombre Base Datos: sublime\_management

Puerto: 3306

Usuario: root

Contraseña:

Probar Conexión Mostrar ☐

- Reiniciamos la app y podemos ver que los datos que nos carga son los de la base de datos online, ya que cuando cerramos la



aplicación y está la opción de “base de datos local” activada, el programa guarda su estado en local, y el estado estaba cargado con los datos de la online

ref	Nombre	Referencia	Pvp
Ejemplo3		Ejemplo ref3	2.12 4.22

- Ahora si creamos un producto se guardará en online y en local

ref	Nombre	Referencia	Pvp
Ejemplo3		Ejemplo ref3	2.12 4.22
Ejemplo4		ref4	3

- Revisamos la base de datos MySQL

	ref	nombre	pvp	cantidad
<input type="checkbox"/>  Editar  Copiar  Borrar	Ejemplo	ref	2.12	23
<input type="checkbox"/>  Editar  Copiar  Borrar	ref3	Ejemplo3	4.22	1
<input type="checkbox"/>  Editar  Copiar  Borrar	ref4	Ejemplo4	3	2

- Conclusión: el programa puede trabajar con las dos bases de datos conjuntamente y es lo recomendable

Estas pruebas en sido realizadas con todos los elementos que se guardan en la base de datos tanto en la online como en la local (Albaranes, productos, ventas, eventos, usuarios)

#### - Reportes

- Esta tecnología debe de imprimir un resumen, según el tipo de reporte elegido, de cada tipo de elemento guardarlo, cabe destacar que estos reportes cogen los datos de la base de datos MySQL(online) por ello debe de estar activada y actualizada, en este caso son 5 reportes
  - “Imprimir albaranes”
    - Previamente a imprimir este reporte debemos tener reportes creados previamente, en caso contrario no aparecerá ningún reporte, por ello para esta prueba vamos a cargar unos albaranes

Albaranes			
ID	Ref	n.Productos	Importe Total
1		1	5.0
2		3	61.0
3	234123443F	2	43.0

- Ahora imprimimos el informe



JasperViewer

### Listado de Albaranes

ID Albaran: 1

Referencia Albaran:	Ref. Producto	Unidades	Precio	Total
	Ejemplo	5	1.0€	5.0€
				5.0 €

ID Albaran: 2

Referencia Albaran:	Ref. Producto	Unidades	Precio	Total
	Ejemplo	5	1.0€	5.0€
	Ejemplo	6	3.0€	18.0€
	ref4	19	2.0€	38.0€
				61.0 €

ID Albaran: 3

Referencia Albaran: 234123443F	Ref. Producto	Unidades	Precio	Total
	Ejemplo	5	1.0€	5.0€
	ref4	19	2.0€	38.0€
				43.0 €

- Conclusión: este informe funciona correctamente
- “Imprimir Clientes”
    - Creo unos clientes

ID	Nombre	Apellido	DNI	Telefono	E-Mail	Localidad	Direccion	Proveedor
1	Marce	Garcia Go...	234123443F	3423482903	marce-g-...	Albacete		si
2	Patricia	Garcia Go...	2345243F	23424234	patru@g...	Albacete	Cter de jae...	no





- Ahora imprimimos

### Resumen Clientes

Cliente: 1

ID: 1

Nombre: Marce                      Apellidos: Garcia Gomez

DNI: 234123443F

Telefono: 3423482903

Email: marce-g-g@hotmail.com

Direccion:

Proveedor:No

Cliente: 2

ID: 2

Nombre: Patricia                      Apellidos: Garcia Gomez

DNI: 2345243F

Telefono: 23424234

Email: patru@gmail.com

Direccion: Cter de jaen 17 b

Proveedor:Si

- Conclusión: este informe funciona correctamente
- “Imprimir proporción productos”
- Creo unos productos

	Nombre	Referencia	Pvp
ref		Ejemplo	2.12
Ejemplo3		ref3	4.22
Ejemplo4		ref4	3

- Ahora imprimimos



- Conclusión: este informe funciona correctamente
- “Imprimir información de un albarán”
- Debemos tener albaranes creamos anteriormente

ID	Ref	n.Productos	Importe Total
1		1	5.0
2		3	61.0
3	234123443F	2	43.0



- Ahora imprimiremos, pero según la referencia de algún albarán (para imprimir por proveedor), en este caso lo dejare en blanco para que me imprima los que no tiene proveedor

## Albarán Detallado

REF.:

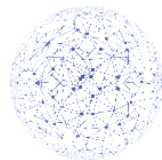
Fecha Informe: 09/06/2022

Ref.Producto	Nombre	Precio Compra	PvP	Unidades	T
Ejemplo	ref	1.0 €	2.12 €	5	
Ejemplo	ref	1.0 €	2.12 €	5	
Ejemplo	ref	3.0 €	2.12 €	6	
ref4	Ejemplo4	2.0 €	3 €	19	
				35	

- Conclusión: este informe funciona correctamente ya que tiene que mostrarte todos los productos que le has comprado a ese proveedor (un Histórico)
- “Imprimir venta”
  - Debemos de tener una venta de la cual queremos imprimir un ticket

ID	Fecha	Cliente	Importe Total
1	2022-06-09	2345243F	30.72

- Ahora imprimimos



Nombre Empresa: Marcelino S.L.  
 Direccion: Avnd los Toreros 22  
 Telefono: 35434534  
 Email: marce-g-g@hotmail.com

Ticket Venta: 1

Fecha Venta: 09 junio 2022

Fecha Impresion: 09/06/2022 20.52.23

Producto	Unidades	Coste	Total
Ejemplo	6	2.12 €	12.719999 €
ref4	4	3.0 €	12.0 €
			24.72 €



- Conclusión: este informe funciona correctamente ya que los datos de empresa estaban configurados con anterioridad

✓ Habilitar datos de Empresa

Nombre Empresa: Marcelino S.L.

Dirección: Avnd los Toreros 22

Teléfono: 35434534

E-mail: marce-g-g@hotmail.com

Logo: Elegir Icono  
C:\Users\Marce\Documents\bola.png

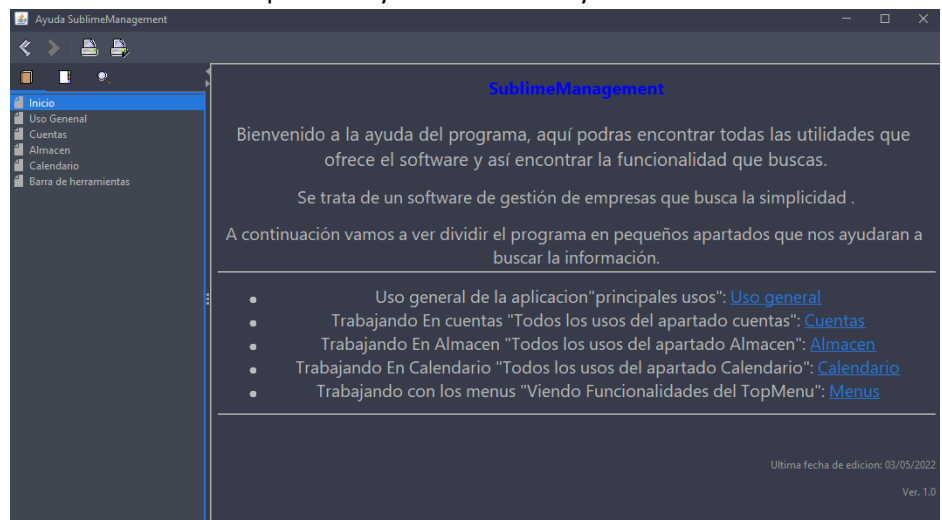


#### - Ayuda

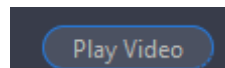
- Esta implementación debería de poder reproducir un video siempre que hagas en el botón correspondiente, así como hacer búsquedas por referencia, por ello vamos a hacer las pruebas pertinentes

- Reproducción de video:

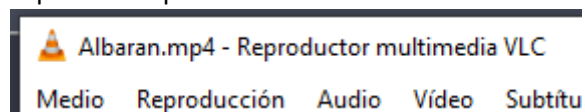
- Para ello abrimos la aplicación y nos vamos a “ayuda”



- Ahora localizamos un botón y vemos si reproduce el video



- Ahora le hacemos clic y vemos como se nos abre el reproductor por defecto



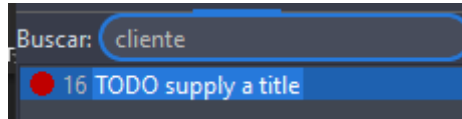
- Conclusión: La funcionalidad es correctamente implementada



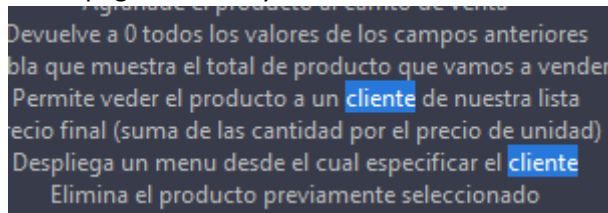
- Búsqueda por referencia:
  - Para ello localizamos el navegador que es la tercera opción



- Y escribimos una palabra



- Vemos que nos encuentra las referencias dentro de cada una de las páginas de la ayuda



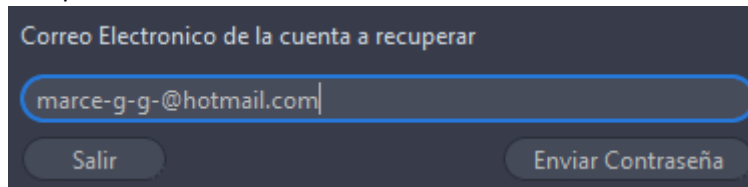
- Conclusión: Esta funcionalidad hacer lo correcto

#### - E-Mail

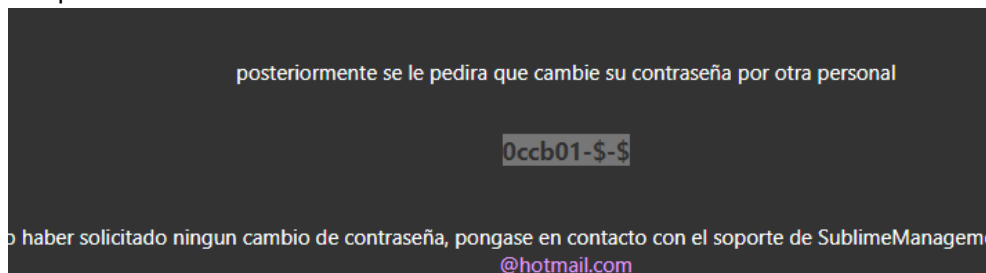
- El uso de JavaMail debe de estar reflejado a la hora de recuperar una contraseña de una cuenta de usuario y cuando queremos ver la contraseña de un usuario

##### ▪ Recuperación Cuenta:

- En la ventana de "Login" vamos a apartado de recuperar contraseña y escribimos el mail de la cuenta que queremos recuperar la contraseña

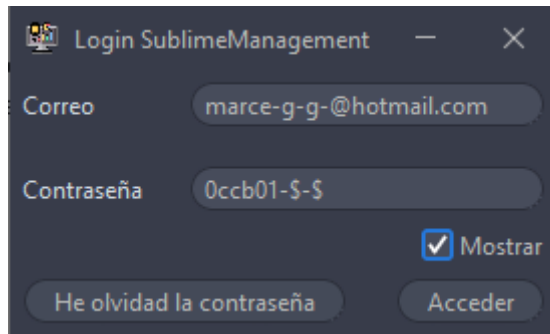
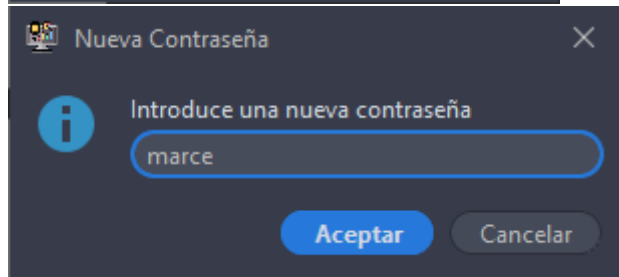


- Y al correo nos debe de haber llegado un mensaje con un clave temporal

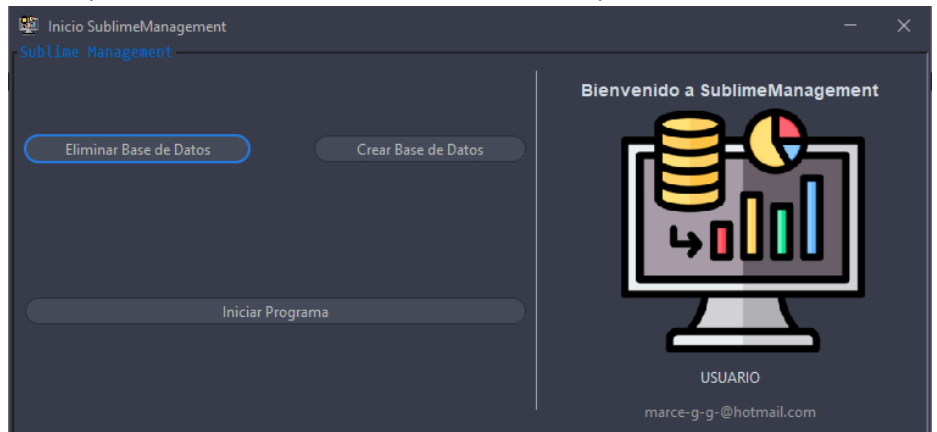


- Ahora la introducimos como como contraseña del correo correspondiente y nos aparece un cuadro de dialogo donde deberemos meter nuestra contraseña personal

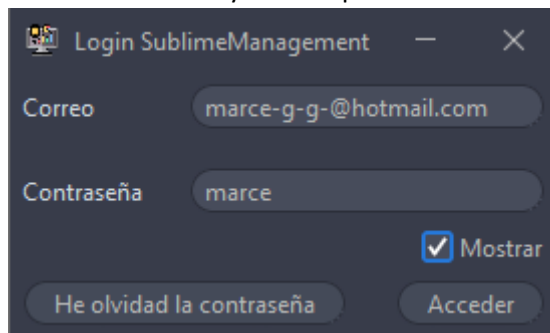


A screenshot of the 'Login SublimeManagement' window. It has a dark blue background. At the top, there's a title bar with a small icon, the text 'Login SublimeManagement', and window control buttons. Below the title bar, there are two input fields: 'Correo' with the value 'marce-g-g@hotmail.com' and 'Contraseña' with the value '0ccb01-S-S'. To the right of the password field is a checkbox labeled 'Mostrar' which is checked. At the bottom, there are two buttons: 'He olvidad la contraseña' and 'Acceder'.A screenshot of the 'Nueva Contraseña' window. It has a dark blue background. At the top, there's a title bar with a small icon, the text 'Nueva Contraseña', and window control buttons. Below the title bar, there's an information icon (i) and the text 'Introduce una nueva contraseña'. Below this is an input field with the value 'marce'. At the bottom, there are two buttons: 'Aceptar' and 'Cancelar'.

- Ahora podemos ver como estamos dentro de la aplicación

A screenshot of the 'Inicio SublimeManagement' window. It has a dark blue background. At the top, there's a title bar with a small icon, the text 'Inicio SublimeManagement', and window control buttons. Below the title bar, there's a subtitle 'Sublime Management'. In the center, there are three buttons: 'Eliminar Base de Datos', 'Crear Base de Datos', and 'Iniciar Programa'. On the right side, there's a sidebar with the text 'Bienvenido a SublimeManagement' and an illustration of a computer monitor displaying a database icon, a pie chart, and a bar chart. Below the illustration, it says 'USUARIO' and 'marce-g-g@hotmail.com'.

- Cerramos la aplicación e intentamos iniciar sesión con la nueva contraseña y vemos que funciona

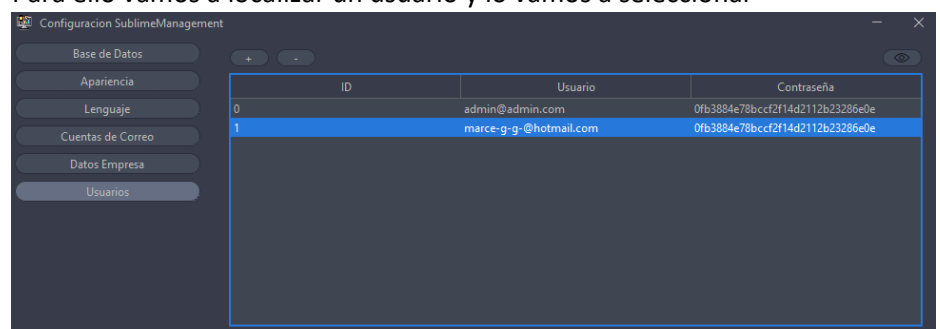
A screenshot of the 'Login SublimeManagement' window. It has a dark blue background. At the top, there's a title bar with a small icon, the text 'Login SublimeManagement', and window control buttons. Below the title bar, there are two input fields: 'Correo' with the value 'marce-g-g@hotmail.com' and 'Contraseña' with the value 'marce'. To the right of the password field is a checkbox labeled 'Mostrar' which is checked. At the bottom, there are two buttons: 'He olvidad la contraseña' and 'Acceder'.

- Ahora vemos que estamos dentro de la aplicación con nuestro usuario





- Conclusión: Esta funcionalidad funciona correctamente
- Ver Contraseña:
  - Para ello vamos a localizar un usuario y lo vamos a seleccionar



- Ahora le damos al ojo de arriba derecha y nos llegara un mail



- Este es mail con la contraseña



- Conclusión: podemos ver que esta funcionalidad hace lo correcto

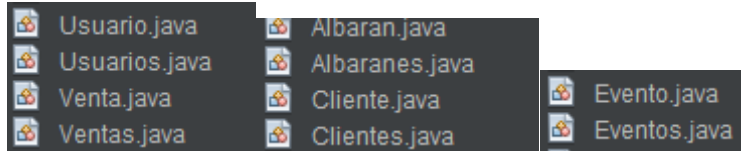


## Código

En este apartado explicare brevemente como esta administrado mi código

### - Estructura de datos base

Para poder gestionar de forma óptima todos los objetos he decido hacer clases y clases contenedoras, para los objetos como (clientes, productos, ventas, eventos, albaranes)



De esta forma logramos tener mejor estructurado el código.

Todas las clases siguen la misma filosofía, el objeto es la clase en singular y la contenedora en la clase en plural, veamos un ejemplo.

Aquí vemos una clase con sus atributos y getter / setter

```
public class Albaran {
    private static int contadorID;
    private int id;
    private List<Producto> producto;
    private List<Integer> cantidad;
    private List<Float> precio;
    private String refAlbaran;
    private float importeTotal;

    public Albaran(){
    }
    public Albaran(List<Producto> producto,List<Integer> cantidad,List<Float> precio,String refAlbaran){
        this.id=Albaran.contadorID;
        this.producto=producto;
        this.cantidad=cantidad;
        this.precio=precio;
        Albaran.contadorID++;
        this.refAlbaran = refAlbaran;
    }
}
```

Aquí podemos ver una clase formada por métodos y atributos estáticos, los métodos son todos para trabajar de forma cómoda con las listas de objetos







```
public class Albaranes {
    private static List<Albaran> albaranes = new LinkedList<>();
    public static void addAlbaranToAlbaranes(Albaran compra){
        Albaranes.albaranes.add(compra);
    }
    public static boolean removeAlbaranFromAlbaranesByID(int identificador){
        int contador=0;
        boolean resultado=false;
        for(Albaran aux : getAlbaranes()){
            if(aux.getId() == identificador){
                albaranes.remove(aux);
                resultado=true;
            }else{
                resultado=false;
            }
            contador++;
        }
        return resultado;
    }
}
```



- *Contenido de las clases*

Cada Clase tiene sus características, decir cada una administra una parte del código diferente, de esta manera poder tener localizados con mayor facilidad los métodos, a su vez estas clases tienen nombres descriptivos que nos ayudan a comprender el contenido de las clases sin necesidad de mirar el código.

Por ejemplo:

-  **Config\_Management.java**  
Permite administrar todo lo referente a la configuración desde el guardado de la configuración, hasta la recogida de esa información
-  **DataType\_Config.java**  
Contiene la estructura de datos de la configuración
-  **GestionReportes.java**  
Esta clase contiene los métodos que ejecutan los reportes de forma correcta
-  **Manager.java**  
Configuraciones generales sobre la aplicación como limpiar los modelos de las tablas, siendo métodos que son necesarios en toda la aplicación
-  **SqlManager.java**  
Gestiona el almacenamiento del programa, tanto la online como la local
-  **EmailManager.java**  
Está compuesto por métodos que permiten el envío de mensajes



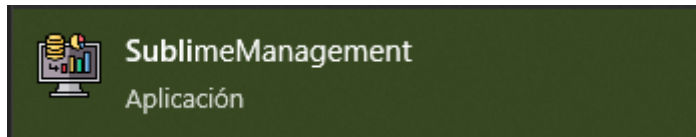


## Manual de usuario

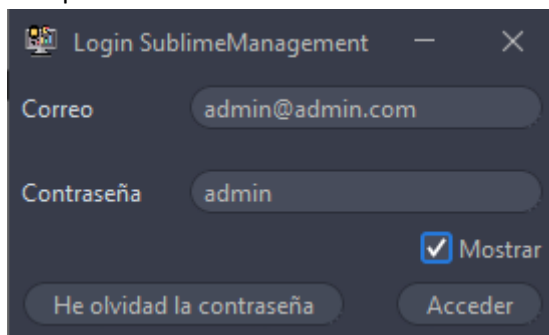
En esta pequeña guía vamos a conocer cuales son las funcionalidades de nuestra aplicación, para sacarle el máximo potencial.

### - Login

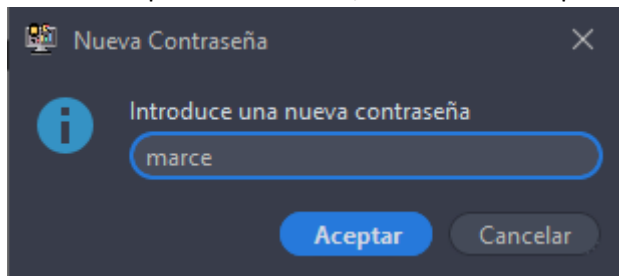
1. Iniciamos la aplicación de SublimeManagement



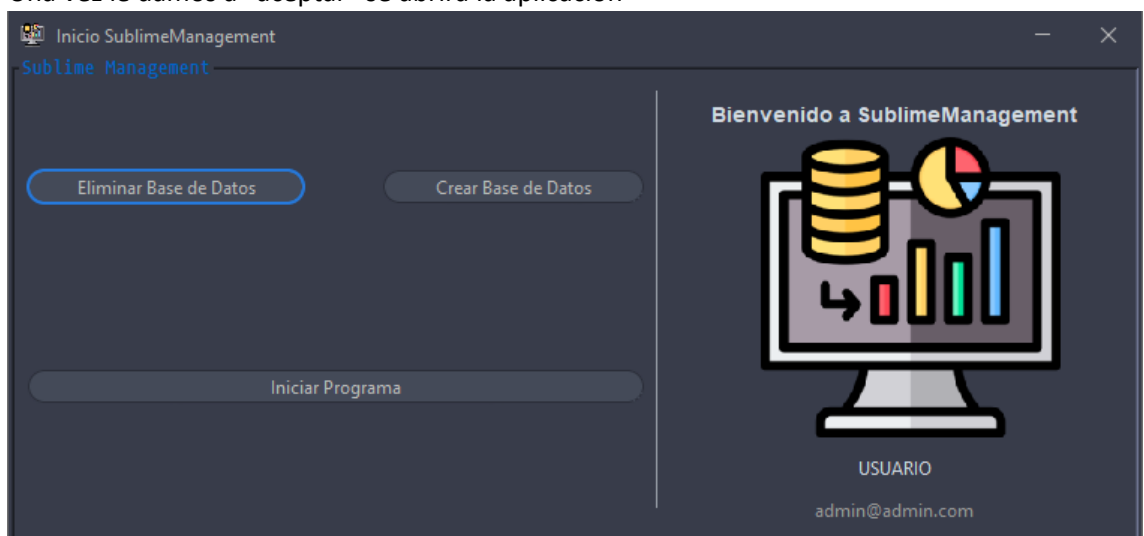
2. Al ser el primer inicio de sesión de nuestra aplicación, vamos a poner los siguientes parámetros, Usuario: [admin@admin.com](mailto:admin@admin.com), contraseña: "admin" tal y como aparece en la captura



3. Ahora le pulsamos al botón de acceder y nos pedirá que introduzcamos otra contraseña para este usuario, la cual debe ser privada

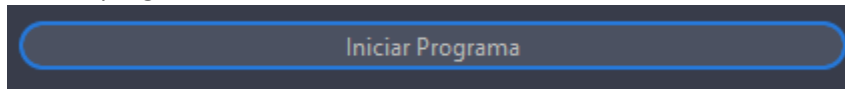


4. Una vez le damos a "aceptar" se abrirá la aplicación

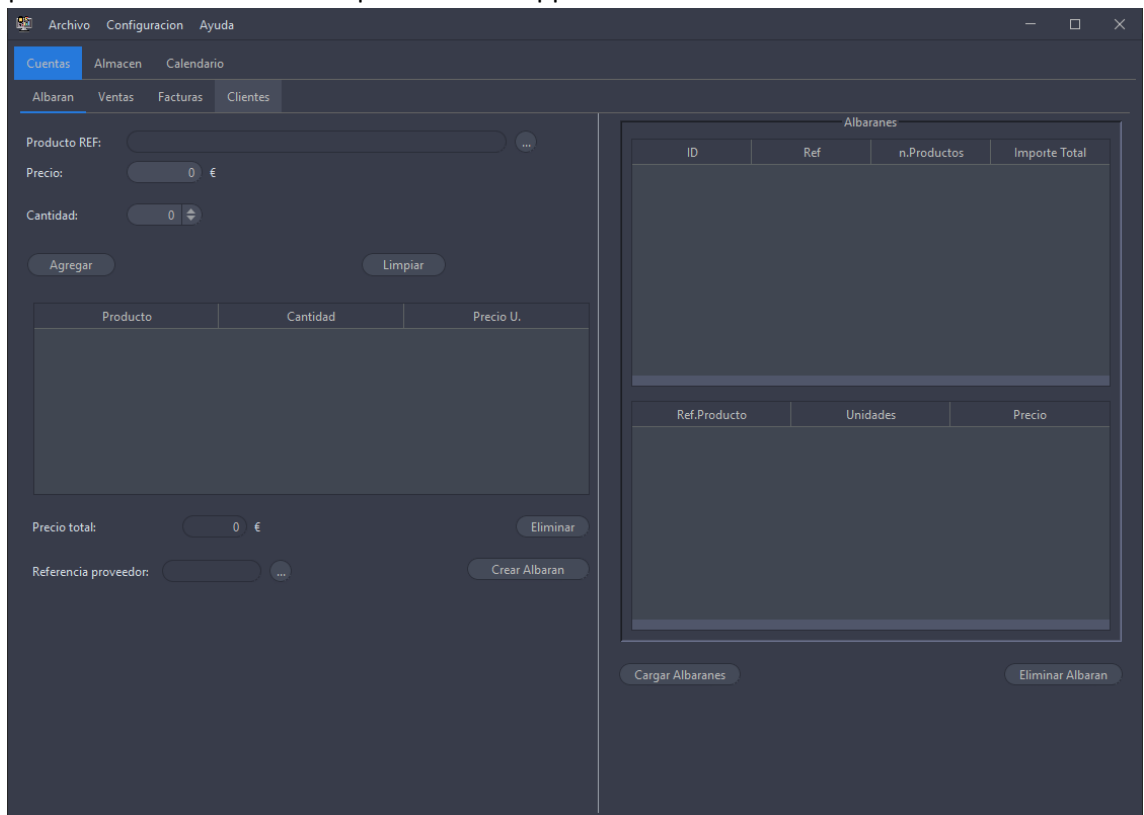


5. Aquí tenemos mucha información, vemos que tenemos 3 botones “Elimina Base de datos”, “Crear base de datos” e “Iniciar programa” también podemos ver que tenemos información del usuario que ha iniciado sesión

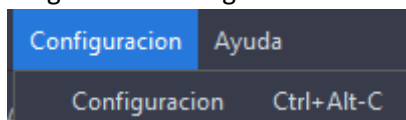
Cuando iniciamos por primera vez solo debemos darle a “Iniciar Programa” y la base de datos se creará por defecto, pero en caso de querer mayor control podemos pulsar el botón de “Eliminar base de datos” y se eliminarán todos los registros de la base de datos y también podemos crearla “Crear base de datos”, la cual crea todas las tablas necesarias antes de iniciar el programa, no obstante, para este caso solo le daremos a “Iniciar programa”



6. Ya estamos dentro de la aplicación, esta es la ventana principal desde donde podremos sacarle el máximo potencial a la app



- Configuración
7. “vayamos por partes”, para explicar como funciona cada apartado primero debemos dirigirnos a “Configuración”

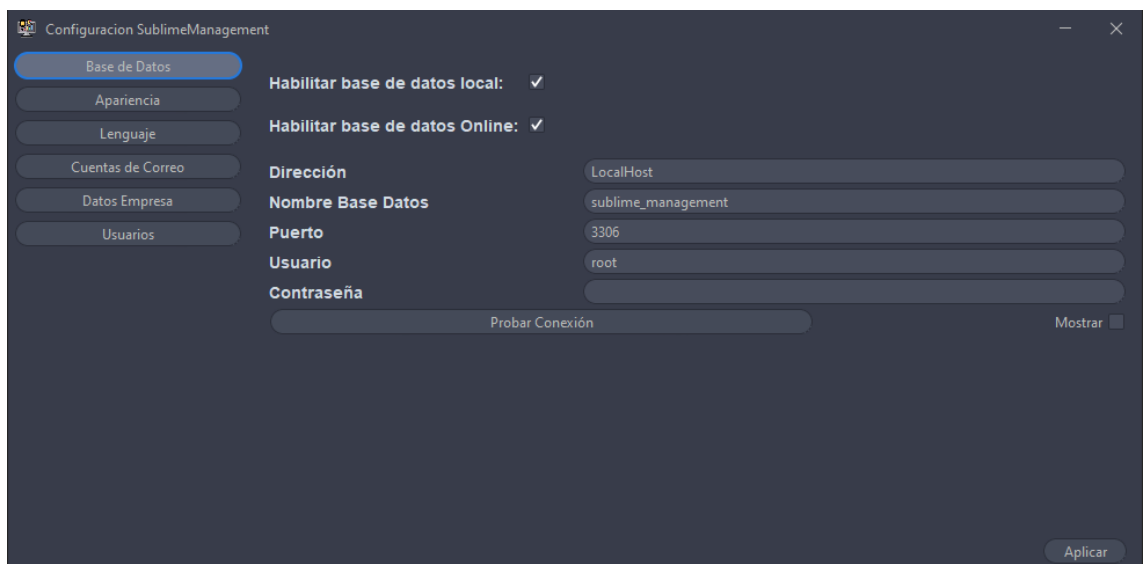


8. Una vez dentro podemos ver varios botones a la izquierda

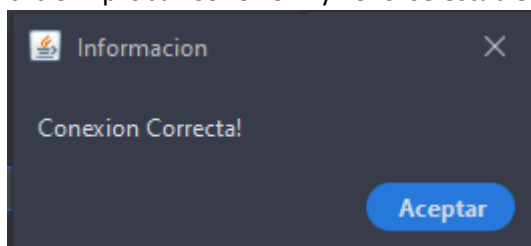


○ *Base de Datos*

9. Nos vamos a dirigir a “Base de Datos”, desde donde podremos gestionar todo lo referente al almacenamiento.

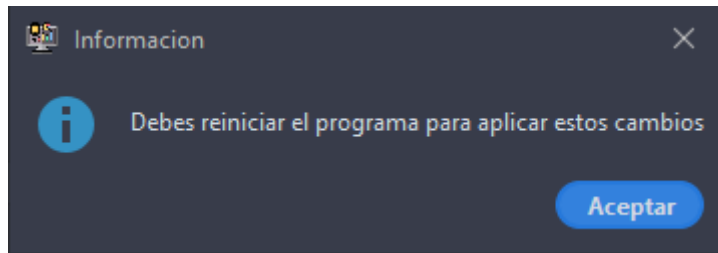


10. Si tenemos activadas las dos casillas nuestra app trabajara con una base de datos local y otra basada en la nube, y la cual deberemos de configurar con los parámetros que aparecen en la captura, en caso de no estar seguros de los parámetros podemos hacer clic en “probar Conexión” y ver si se establece conexión



11. La aplicación puede funcionar con cualquiera de las dos o incluso con las dos al mismo tiempo, el usuario no notara cambio, no obstante, muchas funcionalidades se verán reducidas en el caso de no tener base de” datos online”, una vez terminado de configurar la base de datos le damos a “aplicar”

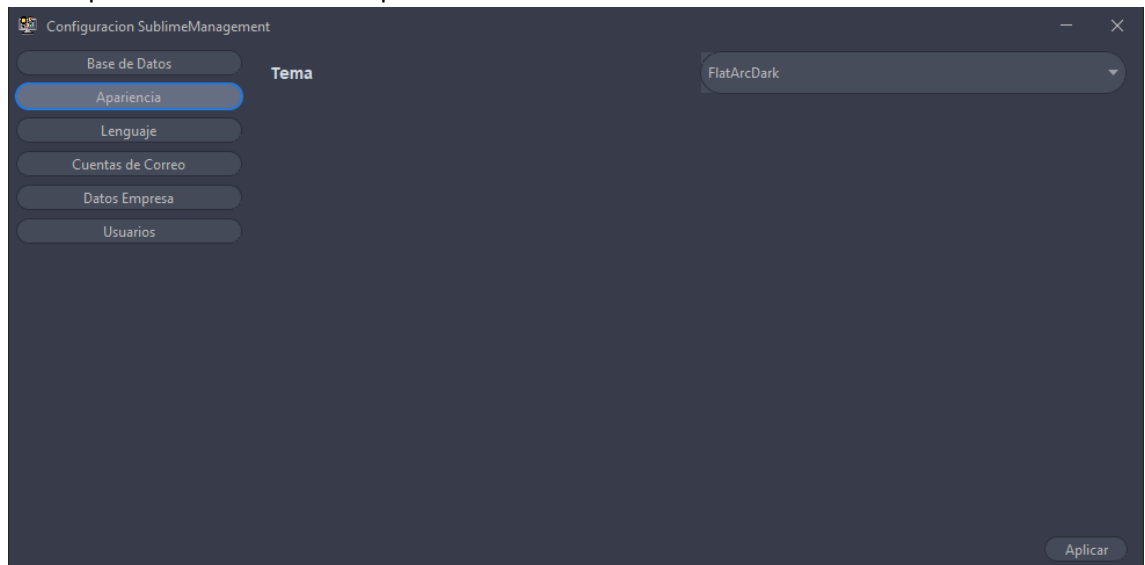




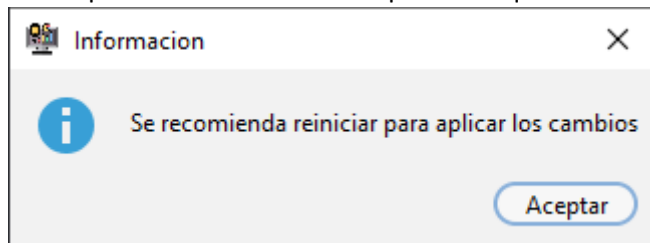
“Algo a destacar es que la base de datos online trabaja mediante un servicio que guarda cada 2 segundos”

- *Apariencia*

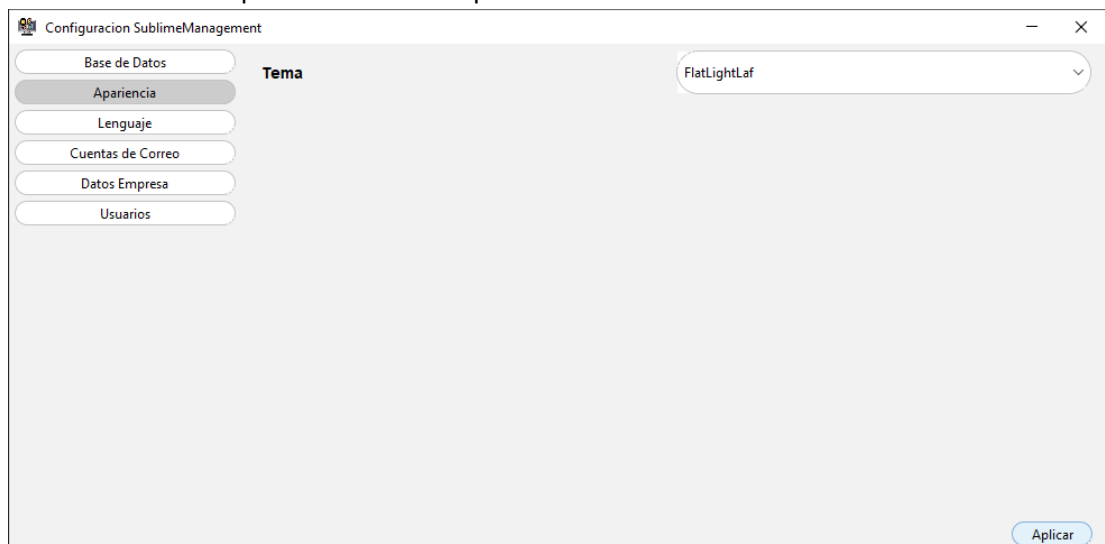
12. Ahora pulsamos el botón de “Apariencia”



13. Ahora podemos seleccionar la apariencia que más nos guste y le damos a aplicar

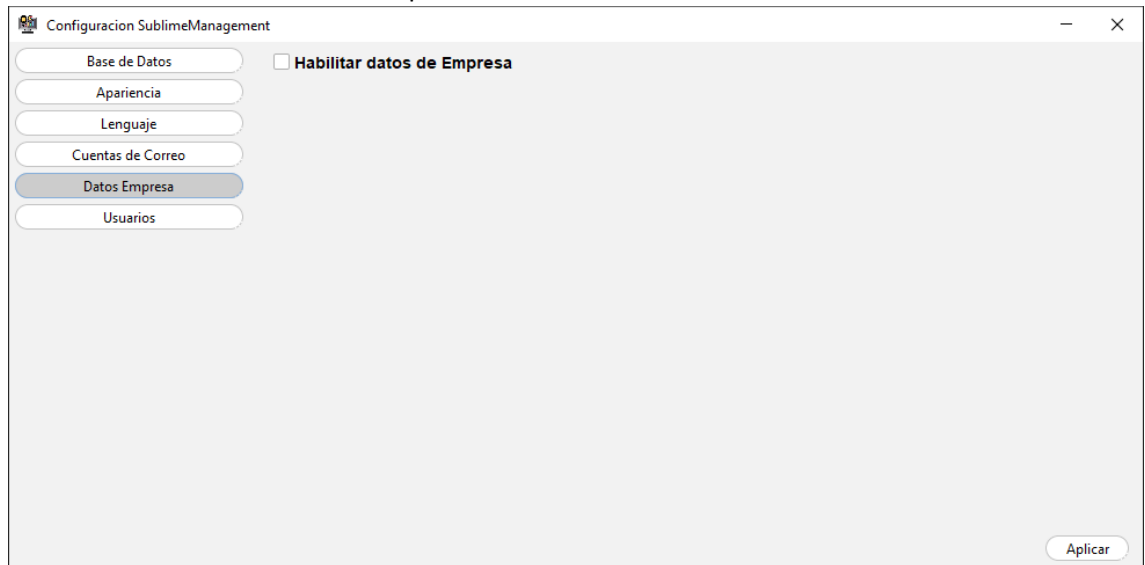


14. Ahora tenemos la aplicación con otra apariencia



- *Datos Empresa*

15. Ahora hacemos clic en “Datos empresa”



Configuracion SublimeManagement

Base de Datos

Apariencia

Lenguaje

Cuentas de Correo

**Datos Empresa**

Usuarios

☐ **Habilitar datos de Empresa**

Aplicar

16. Cuando activemos la opción, nos aparecerán unos campos a rellenar



Configuracion SublimeManagement

Base de Datos

Apariencia

Lenguaje

Cuentas de Correo

**Datos Empresa**

Usuarios

☒ **Habilitar datos de Empresa**

**Nombre Empresa**

**Direccion**

**Telefono**

**E-mail**

**Logo**

Elegir Icono

Aplicar

17. Esto nos permite, posteriormente, imprimir los reportes de las ventas con la información de la empresa



Configuracion SublimeManagement

Base de Datos

Apariencia

Lenguaje

Cuentas de Correo

**Datos Empresa**

Usuarios

☒ **Habilitar datos de Empresa**

**Nombre Empresa**

**Direccion**

**Telefono**

**E-mail**

**Logo**

Marcelino S.L.

Carretera de Jaen, 17B

23498579F

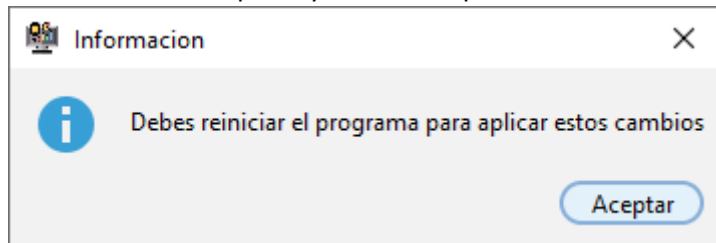
Example@Example.com

Elegir Icono

C:\Users\Marce\Documents\bola.png

Aplicar

18. Ahora le damos a aplicar y los furos reportes saldrán con estos datos



o *Usuarios*

19. Hacemos clic en “usuarios” y nos aparecerá un panel con información de los usuarios del sistema (en caso de que no aparezcan usuario le damos a “Cargar Usuarios”)

Configuracion SublimeManagement

Base de Datos

Apariencia

Lenguaje

Cuentas de Correo

Datos Empresa

**Usuarios**

+

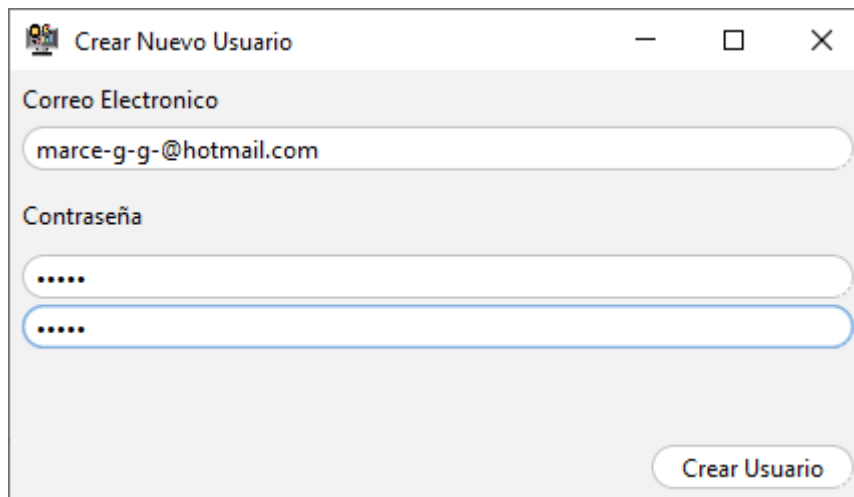
-

ID	Usuario	Contraseña
0	admin@admin.com	0fb3884e78bccf2f14d2112b23286e0e

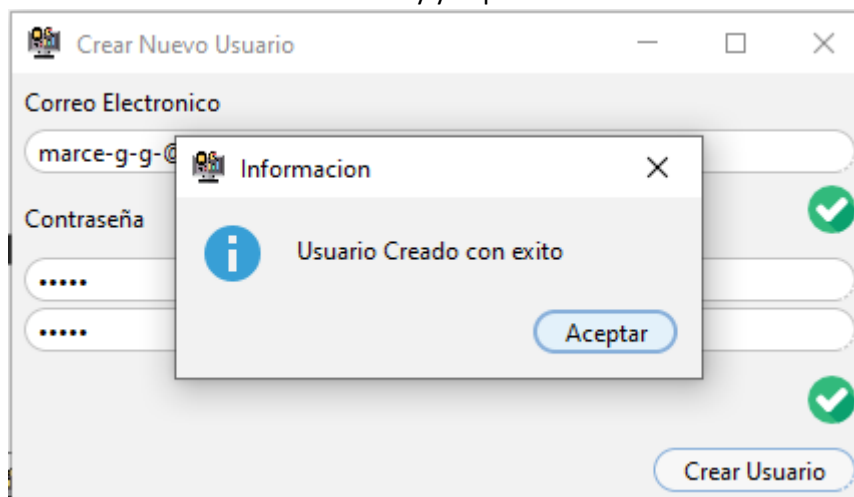
Cargar Usuarios

20. Desde este panel podemos crear, eliminar, ver los usuarios totales, y ver la contraseña de los mismos, en caso de querer crear un usuario le damos al signo “+” y nos saldrá un panel para rellenar con una cuenta de correo valida y operativa y una contraseña

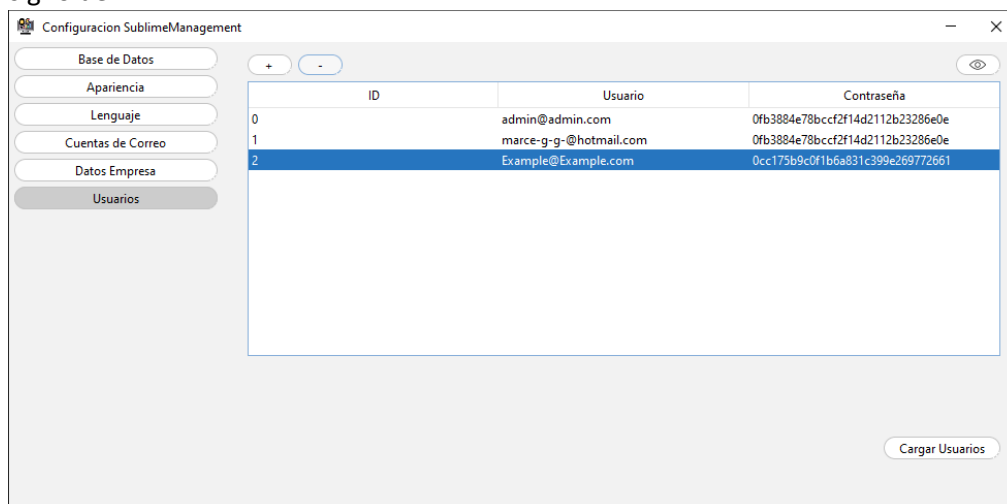




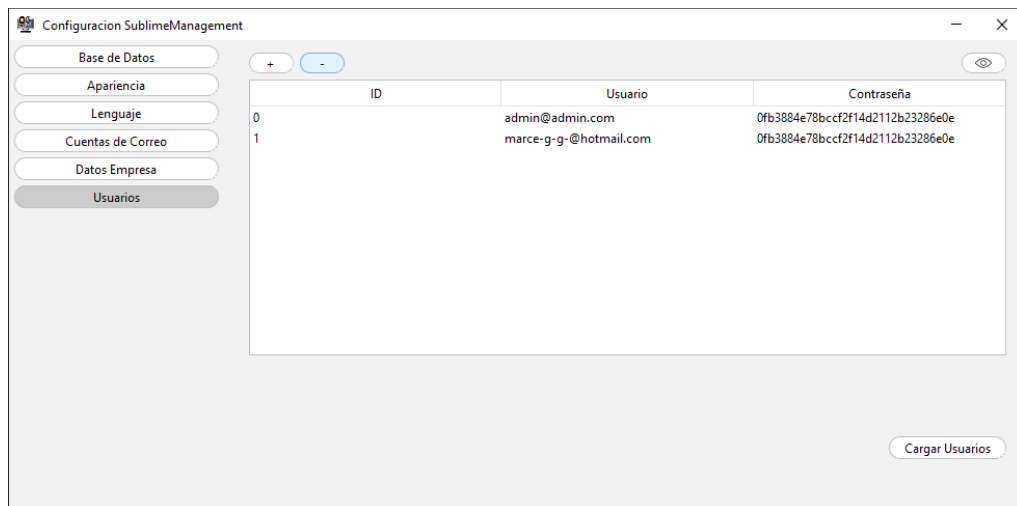
21. Ahora le damos a “Crear usuario” y ya aparecerá en la lista



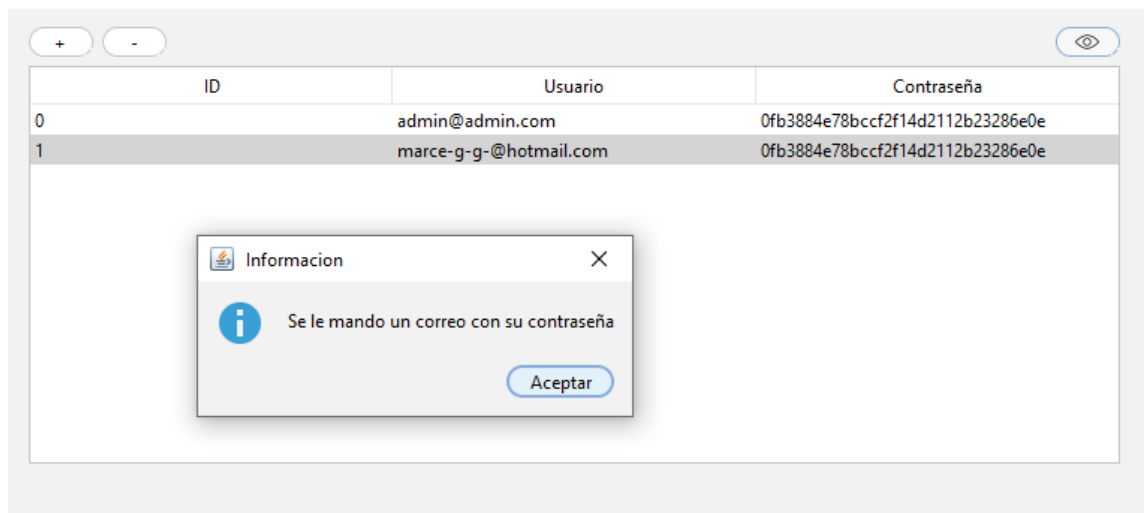
22. Ahora si queremos eliminar un usuario solo debemos selecciona un usuario y darle al signo de “-”



ID	Usuario	Contraseña
0	admin@admin.com	0fb3884e78bccf2f14d2112b23286e0e
1	marce-g-g@hotmail.com	0fb3884e78bccf2f14d2112b23286e0e
2	Example@Example.com	0cc175b9c0f1b6a831c399e269772661



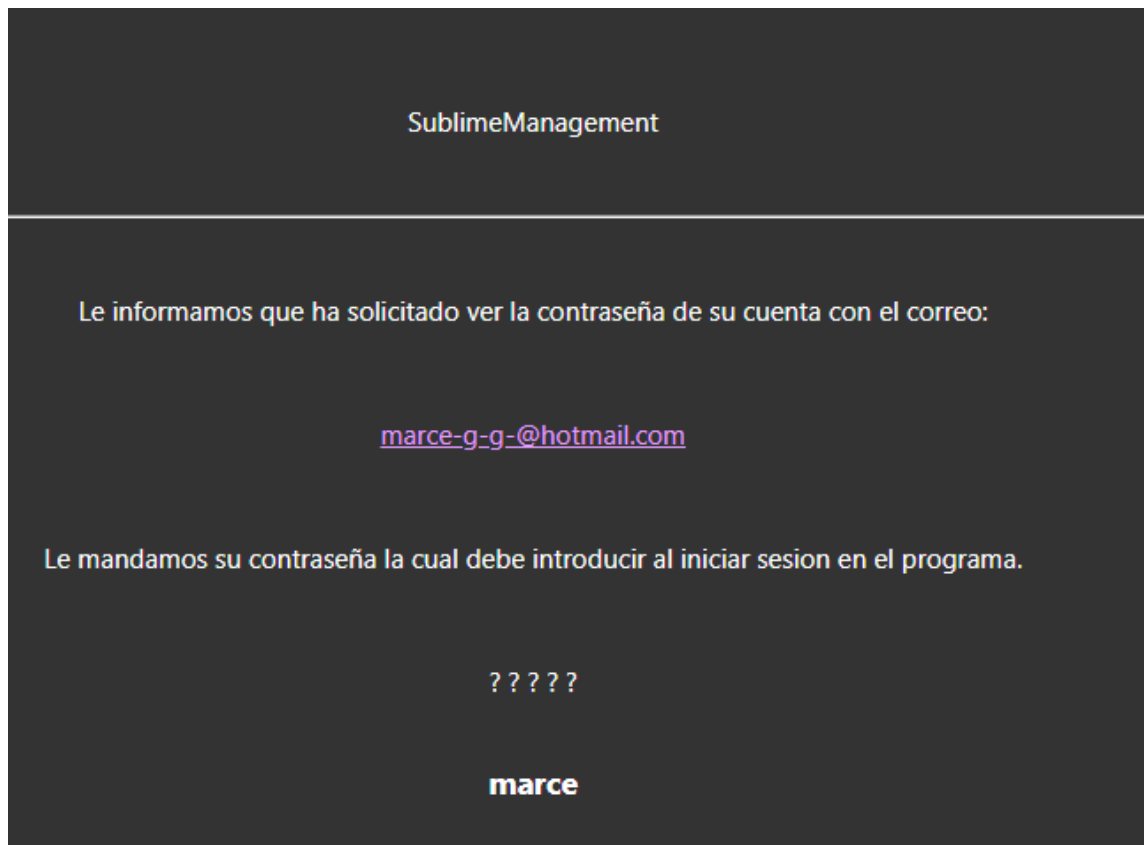
23. Si queremos ver la contraseña de un usuario debemos seleccionarlo y hacer clic en el botón con imagen de ojo de arriba derecha



24. Llegara un correo al usuario correspondiente.







25. El resto de los paneles son para futuras actualizaciones

- Almacén

Esta es la pestaña de crear un producto y ver el “Stock” que tienes

○ *Crear producto*

26. Para la creación de un producto nos dirigimos a “Crear producto” y rellenamos los campos con según el nombre de cada uno tal y como esta en el ejemplo



27. Finalmente le damos a crear y nos cargara el producto

Nombre	Referencia	Pvp
Pan	Ref.1	0.50

28. Este panel permite también “Cargar Todos los productos” en caso de que no aparezca y también permite eliminar un producto, simplemente seleccionándolo y dándole al botón

29. También tiene opción de búsqueda tanto por nombre como por referencia, esta búsqueda no debe ser exactamente de la palabra, ya que puedes poner parte de ella como en el ejemplo



Carne	Ref.Carne	23
-------	-----------	----

Productos Totales: 2

Eliminar producto seleccionado Cargar Todos los Productos

Buscar por nombre: Car Buscar!

Buscar por referencia: Buscar!

Nombre	Referencia	Pvp
Salchichon	Ref.Salchichon	23

Productos Totales: 2

Eliminar producto seleccionado Cargar Todos los Productos

Buscar por nombre: Buscar!

Buscar por referencia: Ref.Sal Buscar!

○ *Stock*

30. Ahora vamos a la pestaña de gestión del almacén “Stock”, desde aquí podemos ver todos los productos que tenemos en almacén, con los precios y las unidades disponibles, hay 3 botones en la parte inferior que nos permite ver con mayor claridad el stock de nuestro almacén, si hacemos clic en “Cargar productos sin Stock” veremos los productos sin stock actualmente



Nombre	Referencia	Cantidad	Precio
Jamon	Ref.Jamon	0	9.89

31. Si hacemos clic en “Cargar productos con Stock” veremos solo aquellos que tengan unidades

Nombre	Referencia	Cantidad	Precio
Carne	Ref.Carne	23	12.90
Salchichon	Ref.Salchichon	23	3.00

32. Si hacemos clic en “Cargar productos” veremos todos los productos en el almacén



Nombre	Referencia	Cantidad	Precio
Carne	Ref.Carne	23	12.90
Salchichon	Ref.Salchichon	23	3.00
Jamon	Ref.Jamon	0	9.89

33. También tenemos las opciones de búsqueda que funcionan igual que anteriormente

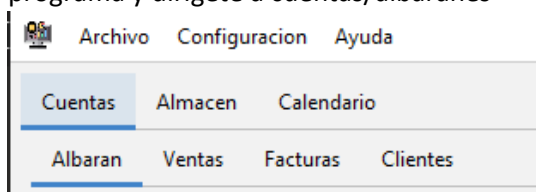
Nombre	Referencia	Cantidad	Precio
Carne	Ref.Carne	23	12.90

## - Cuentas

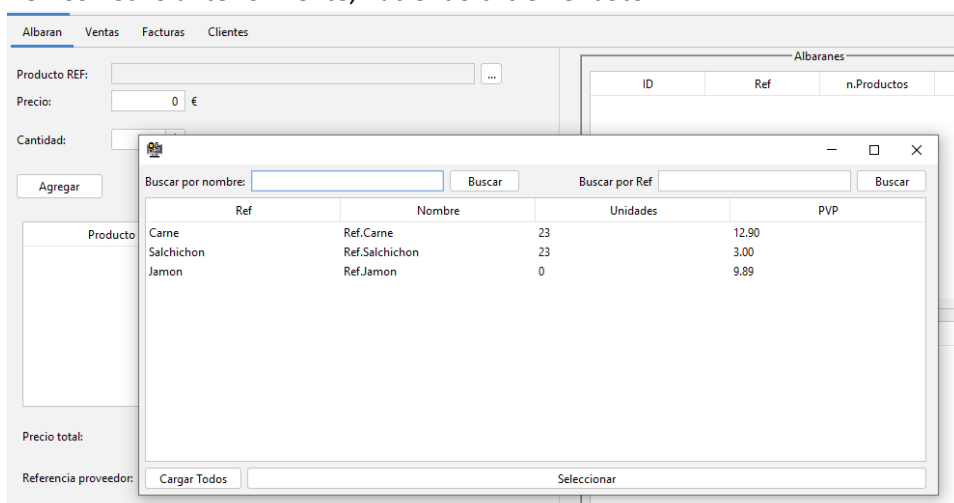
Desde esta pestaña podemos administrar todo lo referente a la economía de nuestra empresa, para poder administrar mejor cada apartado vamos a verlas por subpestañas (Albaranes, ventas, clientes)

### o Albaranes

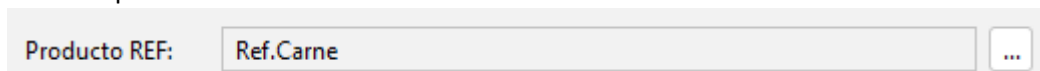
34. La creación de un albarán es útil para poder registrar la entrada un producto en nuestro sistema, por ello siempre que se realice una compra esta debe registrarse, para poder registrarlo debemos crear un alabarán de la siguiente forma, abre el programa y dirígete a cuentas/albaranes



35. Ahora veremos rellenar los campos con los datos correspondientes, primero el producto el cual debería estar registrado previamente en nuestro sistema, tal y como hemos hecho anteriormente, haciendo clic en el botón “...”



36. Ahora vamos a seleccionar el producto y le damos el botón de “seleccionar” y nos saldrá el producto en la en el recuadro



37. Establecemos el precio al que lo hemos comprado y luego la cantidad del producto

Producto	Cantidad	Precio U.
Ref.Carne	5	5.90

38. Ahora debemos darle al botón de agregar, así con todos los productos de los que este compuesto nuestro albarán

Producto	Cantidad	Precio U.
Ref.Carne	5	5.90
Ref.Salchichon	11	1.10

Precio total: 41.6 €

Referencia proveedor: ...

Eliminar

Crear Albaran

39. Ahora en caso de tener a nuestro proveedor registrado podemos seleccionarlo en el apartado "Referencia proveedor" y abrimos el desplegable dándole al botón "..." (más adelante veremos cómo ser crear un cliente y un proveedor)

DNI/NIF	Nombre	Apellido
234523234f	Mercadona	

DNI

Nombre

Apellido

Limpiar

Seleccionar

40. Este selector permite hacer búsquedas según el tipo de campo por el que queremos buscar, en este caso solo vamos a seleccionar el proveedor y le damos al botón de seleccionar

Referencia proveedor:

41. finalmente, ya podremos darle al botón de “Crear albarán” y el albarán se verá reflejado en la tabla de arriba a la derecha, y en caso de hacerle clic a este registro podremos ver un resumen en la tabla de la parte de abajo

**Albaranes**

ID	Ref	n.Productos	Importe Total
1	234523234f	2	41.6

Ref.Producto	Unidades	Precio
Ref.Carne	5	5.9
Ref.Salchichon	11	1.1

42. los productos se han sumado a nuestro almacén, podemos verlo desde la pestaña de “Stock”

Nombre	Referencia	Cantidad	Precio
Carne	Ref.Carne	28	12.90
Salchichon	Ref.Salchichon	34	3.00
Jamon	Ref.Jamon	0	9.89

43. La ventana de albarán también permite eliminar los albaranes en caso de que haber tenido algún error, simplemente debemos seleccionarlo, y darle al botón de eliminar.





ID	Ref	n.Productos	Importe Total
1	234523234f	2	41.6

Eliminar Albaran

44. Los productos se han vuelto a su estado anterior

Nombre	Referencia	Cantidad	Precio
Carne	Ref.Carne	23	12.90
Salchichon	Ref.Salchichon	23	3.00
Jamon	Ref.Jamon	0	9.89

o Ventas

45. Para este apartado nos vamos a dirigir a la ventana de ventas donde podemos efectuar las salidas, para ello

Cuentas	Almacen	Calendario
Albaran	Ventas	Facturas
	Cientes	

46. Ahora vamos a seleccionar los productos que queramos vender tal y como lo hemos con los albaranes

Producto:  ...

47. Ahora seleccionamos la cantidad y el precio se pondrá por defecto el que pusimos al crear el producto, pero también podemos modificarlo también podemos ponerle el descuento

Cantidad:

PVP.  Descuento

48. Finalmente le damos a agregar y hacemos le mismo proceso con el resto de lo producto

Ref.	Cantidad	Precio.U
Ref.Carne	3	12.9
Ref.Salchichon	7	3.0

49. Ahora podemos seleccionar el cliente en caso de querello debemos seleccionar el recuadro, aparecerá la opción de seleccionar un cliente

Cliente: ☒ Precio total:

DNI Cliente:  ...

50. Finalmente le damos a "Vender todo" y nos crea una venta en el panel de la derecha, y si lo seleccionamos podemos ver sus detalles



Registro Ventas

ID	Fecha	Cliente	Importe Total
1	2022-06-10	234523234f	47.699997

Producto	Cantidad	Precio.U
Ref.Carne	3	12.9
Ref.Salchichon	7	3.0

Cargar Ventas Imprimir

Eliminar

51. También podemos eliminar una venta seleccionándola y dando al botón de eliminar, en caso de querer imprimir un ticket con la información de la venta debemos seleccionar la venta que queremos y darle al botón de imprimir

Menú de opciones X

Seleccione un opción

1.- Generar informe Ticket Venta 2.- Salir

52. Ahora le damos al botón número 1 y ya tenemos nuestro ticket, con la información de la empresa que pusimos anteriormente

Nombre Empresa: Marcelino S.L.  
 Direccion: Carretera de Jaen, 17B  
 Telefono: 23498579F  
 Email: Example@Example.com

Ticket Venta: 1      Fecha Venta: 10 junio 2022      Fecha Impresion: 10/06/2022 13.30.25

Producto	Unidades	Coste	Total
Ref. Carne	3	12.9 €	38.699997 €
Ref. Salchichon	7	3.0 €	21.0 €
			59.699997 €

○ *Cliente*

53. Ahora vamos a crear un cliente y u proveedor, para ello nos vamos a la ventana de cliente

Cuentas   Almacen   Calendario

---

Albaran   Ventas   Facturas   **Cientes**

54. Ahora debemos rellenar los datos y en caso de ser un proveedor debemos seleccionar la última casilla.

Nombre

Apellidos:

DNI/NIF:

Telefono:

E-Mail

Localidad

Direccion

Proveedor: ☐



55. Al darle al botón de “Crear cliente” tendremos un cliente/proveedor creado, el cual reflejara en la tabla de la derecha

Datos Clientes								
ID	Nombre	Apellido	DNI	Telefono	E-Mail	Localidad	Direccion	Proveedor
1	Mercadona		234523234f	2323234234	mercadon...	albacete		si

56. En caso de querer eliminarlo debemos seleccionarlo y hacer clic en “Eliminar”, también podemos hacer búsquedas según el campo, es decir podemos poner información en cada uno de los campos de texto y darle al botón de buscar y nos buscara por las coincidencias de los campos rellenados.

Nombre:

Apellidos:

DNI/NIF:

Telefono:

E-Mail:

Localidad:

Direccion:

Proveedor: ☐

Datos Clientes

ID	Nombre	Apellido	DNI	Telefono	E-Mail	Localidad	Direccion	Proveedor
1	Mercadona		234523234f	2323234234	mercadon...	albacete		si

57. En caso de buscar algún cliente sin coincidencias, no nos saldrá nada

Nombre:

Apellidos:

DNI/NIF:

Telefono:

E-Mail:

Localidad:

Direccion:

Proveedor: ☐

Datos Clientes

ID	Nombre	Apellido	DNI	Telefono	E-Mail	Localidad	Direccion	Proveedor

## - Calendario

Ahora vamos a ver como funcionar la creación de eventos y como gestionarlos

### o Crear evento

58. Para ello nos dirigimos a Calendario/crear evento

Cuentas Almacen **Calendario**

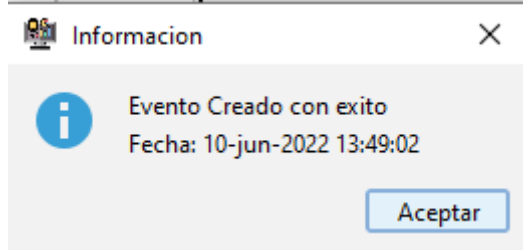
**Crear Evento** Eventos

59. Seleccionamos el día en el calendario de la izquierda y ponemos un nombre, hora, descripción y si es prioritario o no

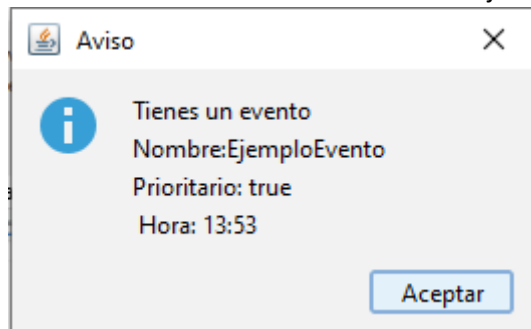


The screenshot shows a web form for creating an event. On the left, there is a calendar for June 2022. The date June 10th is selected and highlighted in grey. To the right of the calendar, there are input fields for event details: 'Nombre del Evento' (EjemploEvento), 'Hora' (12:50), 'Palabras' (1), 'Letras' (7), and 'Descripcion' (Ejemplo). Below these fields, there is a checkbox for 'Prioritario' which is checked. At the bottom of the form, there are two buttons: 'Crear Evento' and 'Limpiar'. A 'Hoy' button is located at the bottom of the calendar.

60. Cuando lo tengamos configurado vamos a darle al botón de “Crear evento”



61. Cuando sea la hora nos saldrá un mensaje informativo



○ *Evento*

62. Desde el apartado “Eventos” podemos ver un histórico de todos los eventos que tenemos creados

Crear Evento

Eventos

ID	Nombre	Descripcion	Fecha	Hora	Prioritario
2	EjemploEvento	Ejemplo	10-06-2022	13:53	SI

Cargar Todos

Eliminar

63. Si no nos aparecen los eventos podemos darle al botón de “Cargar Todos”, y en caso de quiere eliminar un evento debemos seleccionarlo y darle al botón de “eliminar”

Cargar Todos					
Eliminar					

- *Archivo*

aquí podemos ver todos los informes para mostrar información general de todos los datos almacenados, hay 4 tipos de informes

○ *Imprimir albaranes*

64. Nos imprimirá un resumen de todos los albaranes

Listado de Albaranes				
ID Albaran: 2				
Referencia Albaran: 234523234f				
Ref. Producto	Unidades	Precio	Total	
Ref.Carne	5	5.9€	29.5€	
Ref.Salchichon	11	1.1€	12.1€	
			41.6 €	



- *Imprimir clientes*

65. Nos imprimirá un resumen de todos los clientes registrados en el sistema

The screenshot shows a JasperViewer window displaying a report titled 'Resumen Clientes'. The report is for 'Cliente: 1'. The data is as follows:

ID:	1		
Nombre:	Mercadona	Apellidos:	
DNI:	234523234f		
Telefono:	2323234234		
Email:	mercadona@mercadona.com		
Direccion:			
Proveedor:	No		

- *Imprimir proporción producto*


66. Nos imprimirá un resumen en forma de proporción para ver cual es el porcentaje de cada producto sobre el total




- *Imprimir información de un albarán*

67. Para imprimir este tipo al labran nos pedirá la referencia del proveedor, del cual queremos sacar sus albaranes



 Entrada ×

 Referencia de Albaran:

68. Este es el resultado

## Albarán Detallado

REF.: 234523234f

Fecha Informe: 10/06/2022 14.27.40

Ref.Producto	Nombre	Precio Compra	PvP	Unidades	Total
Ref.Carne	Carne	5.9 €	12.90 €	5	29.5€
Ref.Salchichon	Salchichon	1.1 €	3.00 €	11	12.1€
				16	41.6€

