



# MARQUETBEEZUP

## Documentación del proyecto

### Descripción breve

Aplicación de escritorio que permite gestionar la información de pedidos de ventas online de manera rápida y eficiente.

Manuel Rivallo Bejarano  
<https://github.com/rivaman25>

## Contenido

1. TITULO:	2
2. INTRODUCCIÓN	2
a. Explicad por qué se ha decidido realizar este proyecto	2
b. Qué objetivos se persiguen.	2
c. Justificación	3
d. Qué problema intentamos solucionar.	4
3. RECURSOS MATERIALES Y PERSONALES PARA REALIZARLO	6
4. PRESUPUESTO ECONÓMICO.	7
5. FASES Y SECUENCIACIÓN DE LAS ACTIVIDADES DEL PROYECTO.	7
a. Análisis: Diagrama de casos de uso.	7
b. Diseño: Diagrama de clases (UML).	10
c. Diseño: Diagrama E/R de la base de datos.	11
d. Diseño: Interfaces gráficas.	11
e. Implementación: Desarrollo de código e integración con la base de datos.	14
f. Pruebas de Software.	17
6. DEFINIR PROCEDIMIENTOS PARA EVALUAR EL PROYECTO CON LOS CLIENTES. ELABORAR DOCUMENTACIÓN PARA EL CLIENTE.	20
7. CONCLUSIONES	24
a. Dificultades encontradas y soluciones adoptadas.	24
b. Propuestas de mejora del proyecto.	25
c. Ofertas de mantenimiento.	26
8. REFERENCIAS BIBLIOGRÁFICAS.	26
9. ANEXOS.	27

## **1. TITULO:**

MarketBeezUp

## **2. INTRODUCCIÓN**

### **a. Explicad por qué se ha decidido realizar este proyecto**

Marketbeezup surge con la idea de mejorar la productividad de los empleados de una empresa que se dedica a la venta online de electrodomésticos, informática, imagen y sonido. El sistema que utilizan actualmente es la base de datos Microsoft Access, que usan para gestionar una tabla alojada en una base de datos de un Prestashop. En Access tienen implementados varios formularios que les permite realizar tareas como la impresión de albaranes de envío, generar tickets, generar listados en Excel, etc... El problema de este sistema es que el acceso a la información es lento, no existe una validación de los datos que se registran o modifican y es fácil cometer errores que generan inconsistencia en los datos almacenados.

### **b. Qué objetivos se persiguen.**

El objetivo es desarrollar una aplicación de escritorio, que permita a los empleados gestionar la información de los diferentes pedidos de ventas online que reciben a diario de manera rápida y eficiente. Marketbeezup permitirá gestionar los datos de los diferentes pedidos, validando la información que manejen los usuarios y obteniendo información del ERP de la empresa para agilizar diferentes tareas. Registrará la información relativa a cada pedido de venta online: Marketplace, identificador de pedido, fecha de venta, datos del cliente, artículos, cantidades, precios... También almacenará los datos necesarios para que los usuarios realicen diferentes gestiones con los pedidos: códigos postales y las provincias que les corresponden (hay pedidos que no tienen el dato de la provincia, solo el código postal), los rappels (descuentos) por fabricante y gama de producto, el precio de última compra de cada artículo que esté en almacén, el coste de los portes según la agencia de transporte, las comisiones de los Marketplace y los incentivos por venta que tiene cada producto en determinados periodos de tiempo. Los usuarios de la aplicación realizarán diferentes gestiones con los pedidos para que el producto vendido le llegue al cliente.

Las funcionalidades que debe presentar la aplicación son:

- Obtener los datos de los pedidos nuevos que entran a diario.

- Registrar, modificar o eliminar pedidos.
- Gestionar el envío del producto: el usuario introducirá para cada pedido el almacén de salida, la fecha de salida y la agencia de transporte.
- Imprimir/reimprimir los albaranes de envío.
- Obtener un listado de envíos para cada agencia de transporte para gestionar la recogida de la mercancía.
- Registrar información de la compra del producto que no esté disponible en almacén.
- Obtener el ticket de venta de uno o varios pedidos para almacenarlos en PDF. Estos tickets son necesarios para subirlos a determinados Marketplace. No es necesario que estos tickets se registren en el Sistema ERP de la empresa.
- Registrar observaciones o incidencias para un pedido.
- Gestionar consultas, altas, modificaciones y bajas de los Rappels, coste de los portes, incentivos, comisiones de los Marketplace y provincias con su código postal.
- El precio de última compra del producto vendido se obtendrá de la base de datos del Sistema ERP de la empresa.
- Generar un informe de rentabilidad por venta en función de los valores almacenados: el usuario podrá obtener un informe de rentabilidad en función del precio de última compra, rappel, coste de los portes, incentivos del proveedor y comisión del Marketplace.

### **c. Justificación**

La empresa para la que se va a desarrollar la aplicación cuenta con un sistema ERP que no le permite gestionar los datos de los diferentes pedidos de venta online que realizan a diario.

El sistema que usan es lento, propenso a errores y con una funcionalidad muy limitada. Cualquier fallo puede suponer un envío erróneo a un cliente con la consiguiente pérdida económica y valoraciones negativas que afectan a la reputación de la empresa como vendedor en los diferentes Marketplace donde trabajan.

La empresa necesita una herramienta como Marketbeezup que les permita ser eficaces y precisos a la hora de gestionar los pedidos.

#### **d. Qué problema intentamos solucionar.**

Los gestores de venta online utilizan el sistema para obtener los pedidos diarios de las ventas realizadas en los diferentes Marketplace o Webs en las que trabajan. La empresa puede tener una o varias tiendas vendiendo en cada Marketplace o Web. El sistema obtiene los nuevos pedidos accediendo a una base de datos alojada en un servidor web, donde un script registra periódicamente los datos de los pedidos. Normalmente habrá tres o cuatro gestores usando el sistema de manera simultánea. Para cada pedido se necesita obtener el nombre del Marketplace o Web, el nombre de la tienda, el identificador de pedido, la fecha de venta, el importe total, el coste de los portes, la comisión del Marketplace y los datos del cliente: DNI, nombre y apellidos, dirección, código postal, población, provincia y correo electrónico (la provincia en ocasiones no es posible conseguirla y el correo electrónico es opcional). Además, se necesitan los datos de los artículos incluidos en el pedido: código de artículo, descripción, precio de venta, precio de última compra (lo debe obtener del Sistema ERP) cantidad y tipo artículo (es un campo que registra si el artículo existe en el ERP de la Empresa, el campo se actualiza automáticamente por un script y solo es necesario registrarlo). El código de artículo es el mismo para los diferentes Marketplace, pero la descripción y el precio de venta pueden ser diferentes.

Un gestor puede dar de alta pedidos a través de un formulario en caso de fallo del script que registra los pedidos en la base de datos.

El gestor puede modificar los datos del cliente o de los artículos de un pedido. También puede eliminar pedidos si es necesario.

Para gestionar el envío de un pedido el gestor debe introducir la fecha de salida, el almacén de salida y la agencia de transporte para cada artículo del pedido. Para facilitar esta tarea el sistema debe mostrar para cada artículo del pedido el stock por almacén (la empresa cuenta con diferentes almacenes). El sistema debe validar los datos que el gestor introduce, por ejemplo, no se puede marcar una fecha anterior a la actual o no se puede introducir un almacén o una agencia que no existe. El sistema debe permitir al gestor la edición o el borrado de los datos de envío. Además, el gestor podrá listar, añadir, modificar o borrar los almacenes y las agencias con las que trabajan.

En el caso de que algún pedido no se pueda enviar por falta de stock, el gestor realizará la compra del producto al proveedor y registrará en el sistema el número de compra, el

proveedor, la fecha de compra y la fecha de entrada en almacén. El sistema debe permitir modificar o borrar una compra registrada.

Una vez introducida la fecha de salida, almacén y agencia, para que los pedidos puedan ser preparados en almacén para su envío, es necesario que el gestor imprima los albaranes. Para realizar esta tarea, el gestor obtendrá un listado de los pedidos que tienen registrada esta información. Este listado se puede filtrar por la fecha de salida, agencia y/o identificador de pedido. Inicialmente se mostrará el listado de pedidos filtrado con fecha de salida actual para facilitar la impresión de los pedidos del día. El usuario podrá realizar la impresión masiva de este listado o de los pedidos seleccionados. Para cada artículo incluido en el pedido se imprimirá un albarán (el personal de almacén lo introducirá en un sobre que irá pegado al artículo). Una vez impresos los albaranes, el sistema registrará la fecha y hora de impresión para controlar si un albarán está impreso o no. En el Anexo I se incluye un Albarán de ejemplo y una especificación de los datos que debe incluir.

El gestor podrá reimprimir albaranes de pedidos ya impresos. Obtendrá un listado de los pedidos que tienen marcada la fecha de impresión y podrá reimprimir los pedidos seleccionados. El listado se podrá filtrar por fecha de salida, fecha de impresión, agencia o identificador de pedido.

El gestor facturará cada pedido en el Sistema ERP y luego registrará el número de factura y la fecha en el sistema MarketBeezUp. También podrá modificar o borrar esta información.

Un cliente puede cancelar un determinado pedido y el gestor deberá poder marcar el pedido como cancelado. Si el pedido ya tiene registrado fecha de salida, almacén y agencia, y además tiene albarán impreso, el sistema debe avisar al gestor para que pueda comunicar al almacén la cancelación del envío. Además de marcar el pedido como cancelado, el sistema debe eliminar la fecha de salida, almacén y agencia.

El sistema registrará diferentes estados para cada pedido. Inicialmente el pedido aparecerá como "Nuevo". Si se cancela el pedido, el estado pasará a "Anulado". Si se marca fecha de salida, agencia y almacén, el estado cambiará a "Enviar". Cuando se imprime el albarán cambia a "Enviado". En el caso de que se registre la compra, el estado pasará a "Pedido". El gestor podrá introducir un estado personalizado para un pedido que no tiene registrado envío, compra o no está cancelado.

El listado de pedidos se podrá filtrar por una o varias tiendas, uno o varios Marketplace, identificador de pedido, intervalo de fecha de pedido, código de artículo, nombre del cliente, teléfono, una o varias agencias, uno o varios almacenes, intervalo de fecha de salida, estado (si está pedido, cancelado, etc.), número de compra, intervalo de fecha de compra, número de ticket, intervalo de fecha de ticket, si tiene albarán impreso y si tiene o no observaciones.

Para cada agencia de transporte el usuario obtendrá un listado de los pedidos que se envían a través de la misma. Cada listado se guardará en un archivo Excel con un formato específico según la agencia de transporte. Estos listados son necesarios para que el usuario pueda subirlos a la web de cada agencia de transporte y generar las órdenes de recogida en el almacén correspondiente. Se generarán listados diferentes para cada tienda. Para generar un listado el gestor debe seleccionar la tienda, la agencia de transporte y la fecha de salida. Los pedidos del listado deben tener el albarán impreso.

Hay Marketplace que obligan al gestor a subir el ticket de venta a su web una vez que el pedido se ha enviado. El gestor podrá obtener para cada pedido un ticket que podrá guardar en disco, para luego subirlo a la web del Marketplace. Para obtener los tickets, el gestor seleccionará el Marketplace, la tienda y un intervalo de fechas de salida. El ticket deberá tener un diseño similar al albarán.

En ocasiones se producen incidencias con los pedidos que son necesarias registrar para tener un seguimiento por parte del gestor. Para cada pedido el gestor podrá registrar una o varias observaciones y el sistema debe guardar la fecha y hora en la que se registra. El usuario podrá obtener un listado de pedidos con observaciones. Además, podrá editar o borrar observaciones si es necesario.

El gestor podrá registrar, modificar o borrar los datos relativos a las provincias con su código de provincia. Para cada provincia es necesario registrar los dos primeros dígitos del código postal que identifican a la provincia y el nombre de la provincia. En caso de que un pedido no tenga la provincia, pero si el código postal, se usará esta información para obtener la provincia y guardarla.

### **3. RECURSOS MATERIALES Y PERSONALES PARA REALIZARLO**

Los recursos materiales necesarios son:

PC portátil donde instalar todas las herramientas de desarrollo:

- Netbeans IDE

- MySQL server y MySQL Workbench
- Jaspersoft Studio
- Visual Paradigm

El portátil debe tener suficiente potencia para ejecutar estas herramientas con fluidez, la configuración ideal es:

- Procesador Intel Core i5, Amd Ryzen 5 o superior
- Memoria RAM: 16Gb
- Almacenamiento: disco duro SSD 500Gb
- Pantalla: 15,6"

Para mejorar la productividad y disminuir la fatiga visual añadir un monitor de 24" para conectar al portátil.

Los recursos personales necesarios son un técnico superior de desarrollo de aplicaciones, aunque si se quiere disminuir el plazo de entrega del proyecto puede ser necesario contar con algún técnico más.

#### **4. PRESUPUESTO ECONÓMICO.**

Todo el software utilizado es libre, con lo que la empresa tendrá un considerable ahorro en concepto de licencias.

Un portátil con la configuración recomendada tiene un coste de entre 519€ y 799€.

Un monitor de 24" tiene un precio que parte desde los 150€ para las gamas más básicas, hasta los 300€ las gamas más profesionales.

No es necesario incluir gasto asociado a local o transporte ya que el proyecto se puede realizar desde casa.

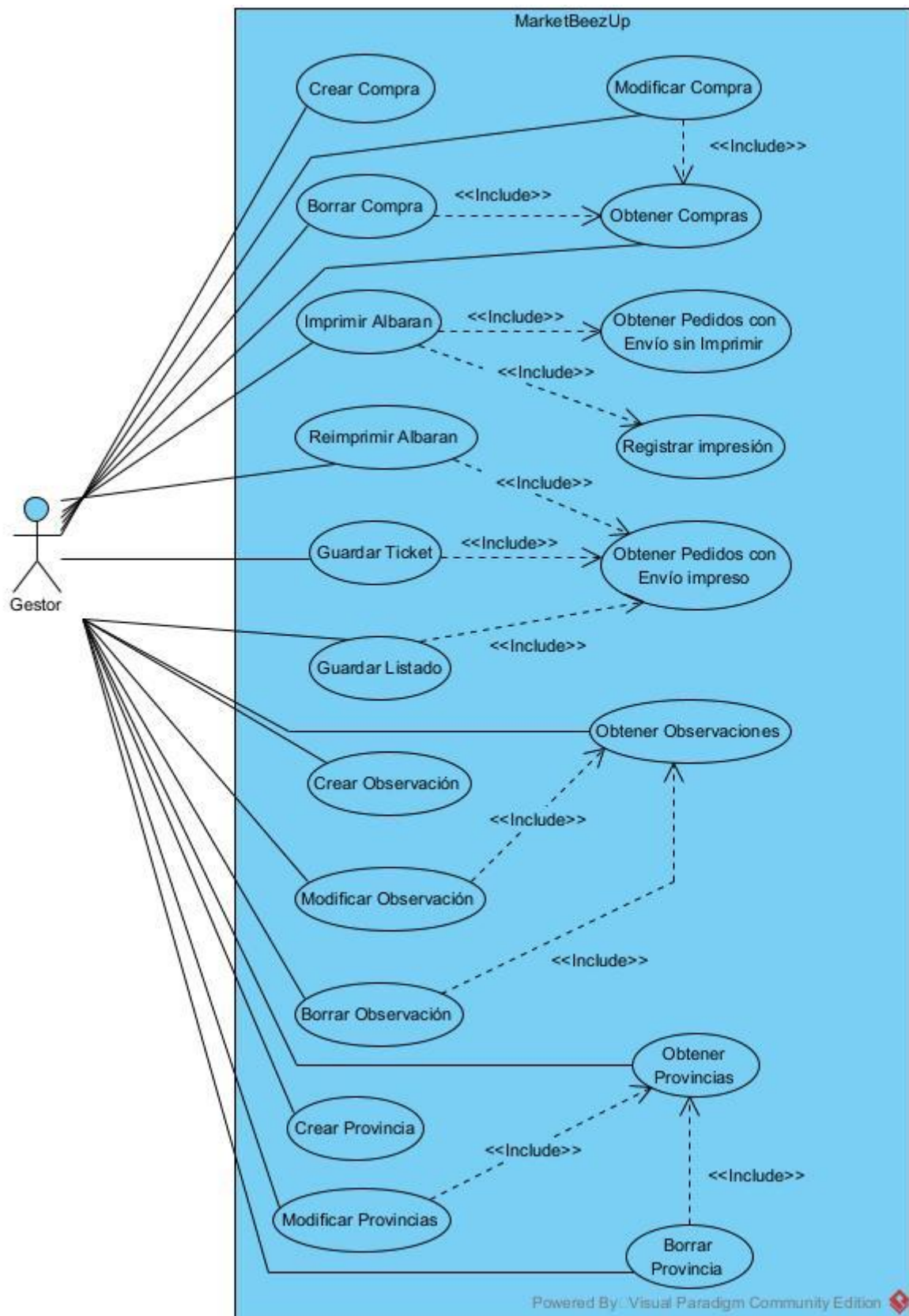
Se estiman 40 horas para el desarrollo del proyecto, aunque pueden aumentar en función de los cambios en los requerimientos de la aplicación. Según el convenio colectivo de ámbito provincial para el sector de la industria, la tecnología y los servicios del sector del metal de Ciudad Real el salario de un desarrollador de aplicaciones para las 40 horas puede rondar los 400€, a los que hay que sumar las cotizaciones a la Seguridad Social.

#### **5. FASES Y SECUENCIACIÓN DE LAS ACTIVIDADES DEL PROYECTO.**

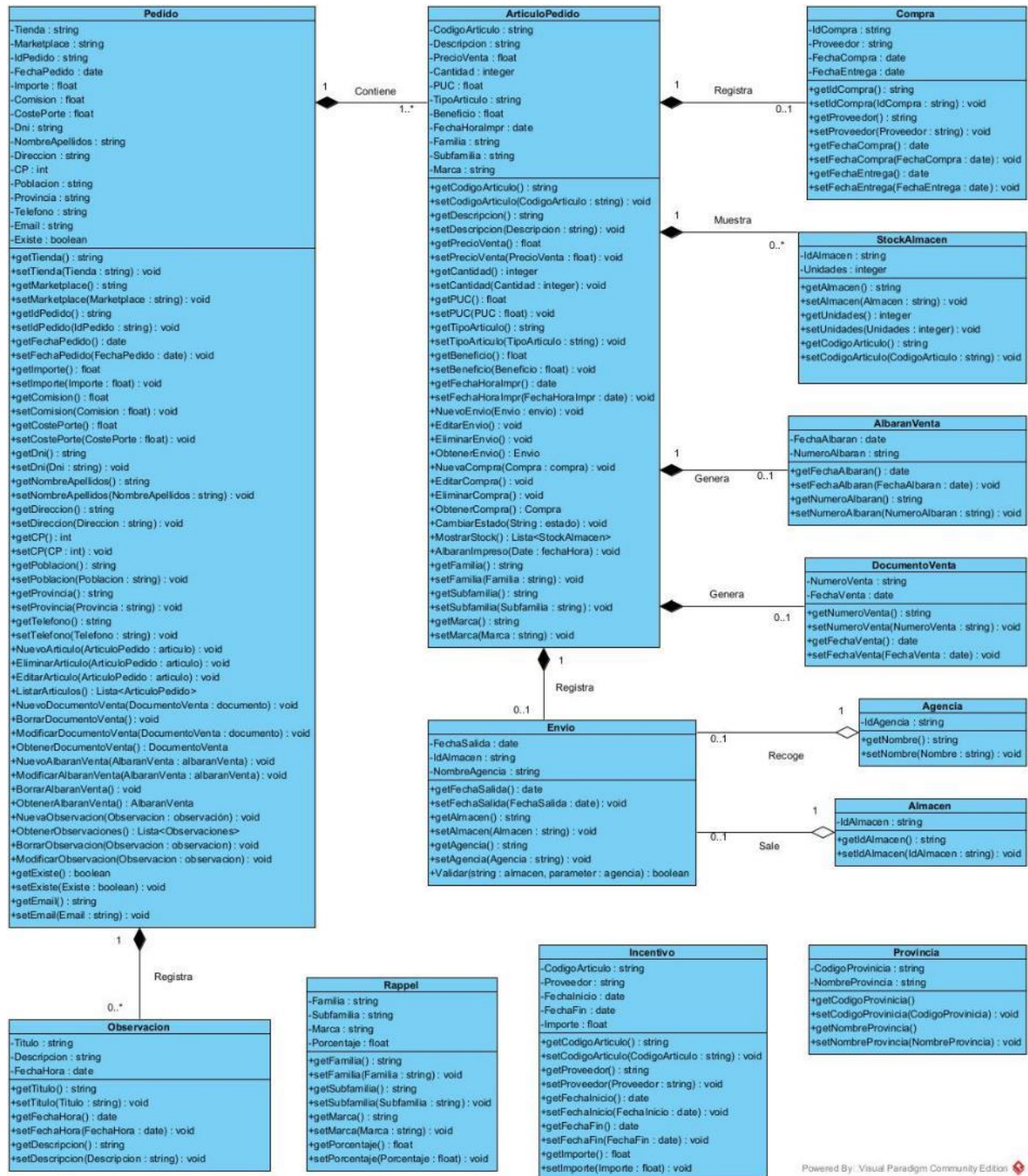
a. Análisis: Diagrama de casos de uso.





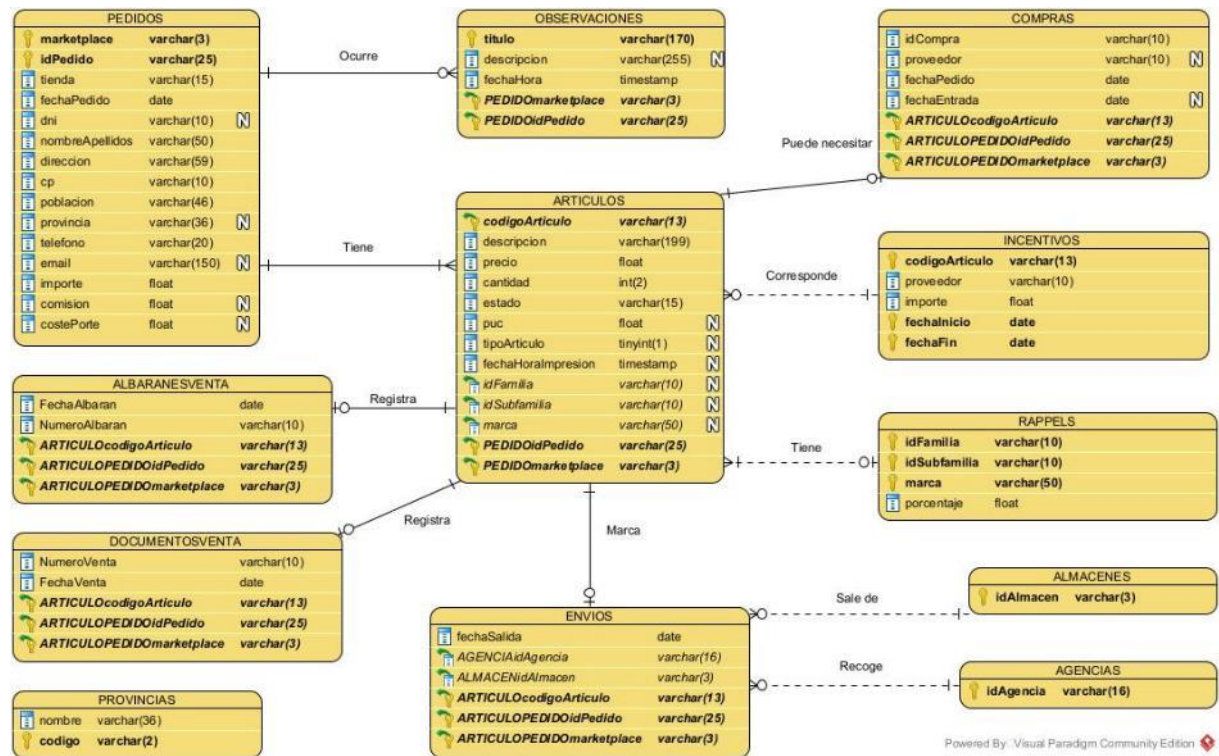


## b. Diseño: Diagrama de clases (UML).





c. Diseño: Diagrama E/R de la base de datos.



d. Diseño: Interfaces gráficas.

Ventana principal:

[Archivo](#)
[Editar](#)
[Pedidos](#)
[Envíos](#)
[Compras](#)
[Albaranes](#)
[Tickets](#)
[Listados](#)
[Observaciones](#)
MarketBeezUp

---

Filtro
 Enviar
 Compra
 Albaranes
 Tickets
 Listados
 Anular
 Observaciones

---

## Pedidos

Ventana de filtros:

**Seleccionar Filtro**

**Filtro**

Tienda	Marketplace	Agencia	Estado	Almacén
BombayFR	AMA	BOMBAY	ANULADO	000
Bombay	CAR	DAC	CANCELADO	001
ElectromuebleFR	CDI	DAC/GOI	CONSULTAR	002
BombayPT	FNA	DAC/HOME	EMBALAR EXPO	003
Electromueble	MAY	DROP TDN	ENVIAR	004
Bomba	PCC	DROP BSH	ESPERAR	005
Bombay	POL	DROP CAYBE	ESPERAR	006

Fecha Pedido: Desde 04-02-22 Hasta   
 Fecha Salida: Desde Hasta   
 Fecha Compra: Desde Hasta   
 Fecha Ticket: Desde Hasta   
 Fecha Albarán: Desde Hasta   
☐ Sin Envío   
☐ Sin Compra   
☐ Sin Ticket   
☐ Sin Albarán   
☐ Con observaciones ☐ Ocultar Existentes

Aplicar Cerrar Limpiar

Ventana de configuración:

**Parámetros de Configuración**

**BD Marketbeezup**

Dirección IP: localhost

Puerto: 3306

Base de datos: marketbeezup

Usuario: root

Contraseña: .....

**BD Online**

Dirección IP: localhost

Puerto: 3306

Base de datos: online

Usuario: root

Contraseña: .....

**BD ERP**

Dirección IP: localhost

Puerto: 3306

Base de datos: erp

Usuario: root

Contraseña: ....

**Listado de Pedidos**

Días a mostrar: 120

Guardar Cancelar

Ventana editar Provincias/Almacenes/Agencias:

**Provincia**

**Código:**  **Provincia:**

Registrar Eliminar

Código	Provincia
14	Cuenca
17	Toledo

Salir

Ventana de nuevo/editar pedido:

**Nuevo Pedido**

**Pedido**

Tienda  Marketplace

IdPedido  Fecha

NIF  Nombre

Dirección  Código Postal

Población  Provincia

Teléfono  Email

Comisión  0.0 Porte  0.0 Importe  0.0

**Artículo**

Id Artículo  Descripción

Precio  Unidades

**Lista de Artículos**

Id Artículo	Descripción	Precio	Unidades

Ventana de nuevo envío:

**Nuevo Envío**

**Envío**

Marketplace:  THE Fecha salida:  4 jun 2022

ID Pedido:  C220311000151-A Agencia:  BOMBAY

Código Artículo:  4242005237265 Almacén salida:  002

**Stock**

Ventana de nueva compra:

**Nueva Compra**

**Compra**

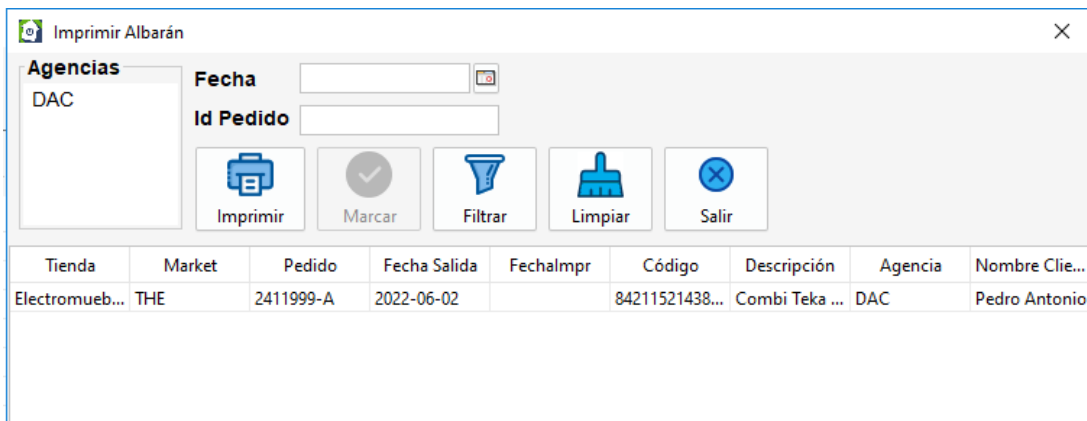
Marketplace:  THE Id Compra:

ID Pedido:  C220311000131-A Proveedor:

Código Artículo:  8434778013610 Fecha compra:  14 jun 2022

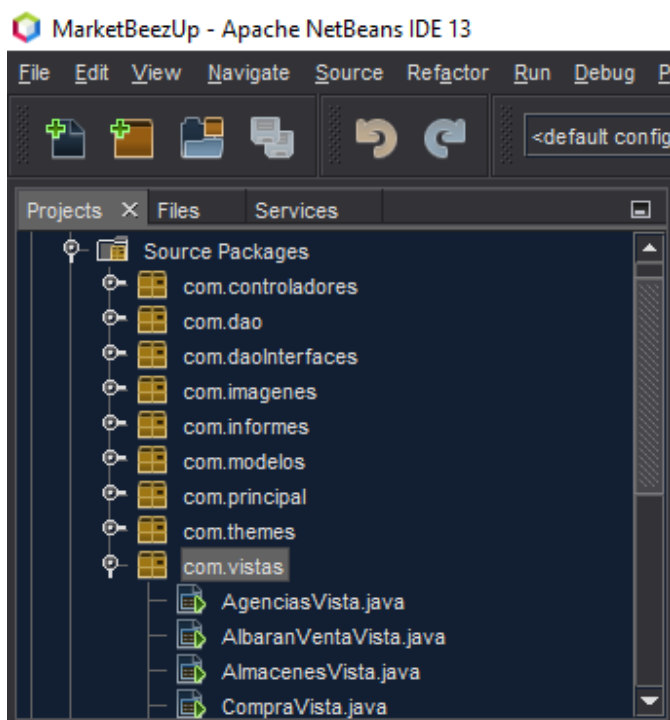
Fecha entrada:

Impresión de albaranes:



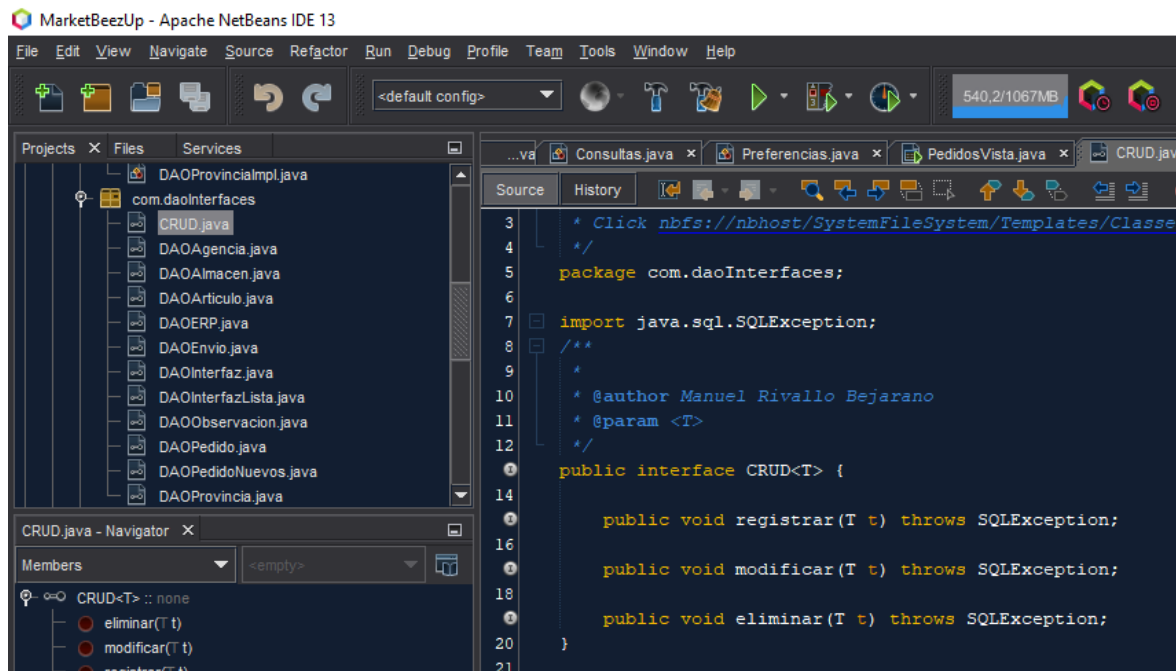
### e. Implementación: Desarrollo de código e integración con la base de datos

La aplicación se ha desarrollado siguiendo la arquitectura Modelo Vista Controlador (MVC). Las clases controladores, modelos y vistas se han almacenado en los paquetes `com.controladores`, `com.modelos` y `com.vistas` respectivamente.



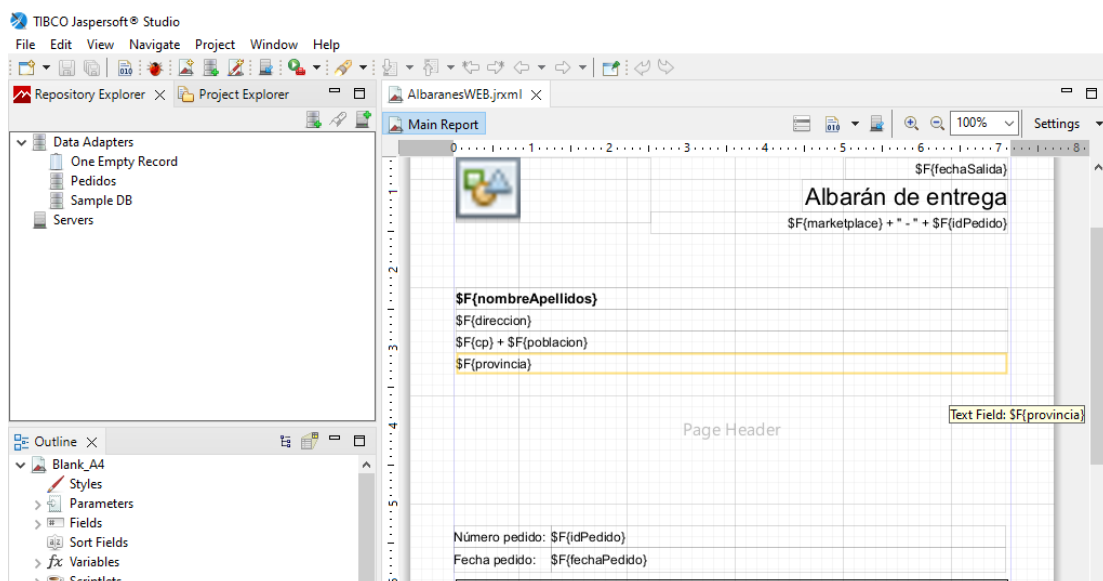
(MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. He usado esta arquitectura porque facilita el diseño de la aplicación y su posterior mantenimiento.

Para el acceso a los datos he utilizado el patrón de diseño Data Access Object (DAO), que separa la lógica de negocio de la lógica de acceso a los datos. DAO proporciona los métodos para insertar, actualizar y borrar la información.



He seleccionado MySQL como Sistema Gestor de Bases de Datos. Es un SGBD relacional de tipo cliente-servidor y código abierto bajo licencia GNU-GPL para el desarrollo de productos no privativos. Para administrar y trabajar MySQL he usado MySQL Workbench 8.0 CE en el entorno de desarrollo y pruebas.

Para diseñar los informes de los albaranes de envío he usado la herramienta TIBCO Jaspersoft Studio CE 6.19.0.



Las librerías JasperReport permiten generar los informes en la aplicación, imprimirlos y

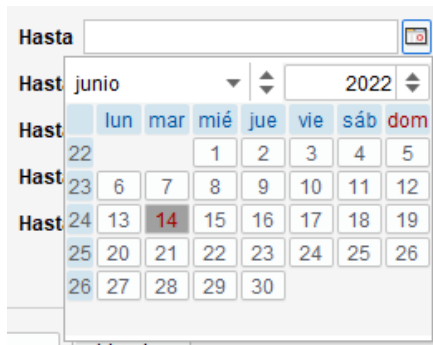


exportarlos a diferentes formatos, XLS, HTML, CSV, PDF, etc.

```
InputStream jasperStream =
Thread.currentThread().getContextClassLoader().getResourceAsStream("com/informes
/AlbaranesWEB.jasper");
    Map<String, Object> parametros = new HashMap<>();
    // Si quiero reimprimir Albaranes obtengo los que selecciona el usuario
    try {
        this.openConnection();
        parametros.put("idPedidos", Artículo.getIdpedidos(articulosImpr));
        parametros.put("marketplaces",
Artículo.getMarketplace(articulosImpr));
        parametros.put("codigoArticulos",
Artículo.getCodigoArticulos(articulosImpr));
        JDialog viewer = new JDialog(imprimirVista, "Impresión Albaranes",
true);

        viewer.setSize(800, 600);
        viewer.setLocationRelativeTo(null);
        JasperReport reporte = (JasperReport)
JRLoader.loadObject(jasperStream);
        JasperPrint jp = (JasperPrint) JasperFillManager.fillReport(reporte,
parametros, this.getConnection());
        JasperViewer jv = new JasperViewer(jp, true);
        viewer.getContentPane().add(jv.getContentPane());
        viewer.setVisible(true);
        if (!reimprimir) {
            imprimirVista.habilitarMarcar(true);
        }
    } catch (JRException ex) {
        JOptionPane.showMessageDialog(imprimirVista, "No se puede obtener
el Albarán", "Error Albarán", JOptionPane.ERROR_MESSAGE);
    } finally {
        this.closeConnection();
    }
}
```

Para el desarrollo de las interfaces gráficas de usuario he usado la librería Java Swing. Como Java Swing no tiene un componente que permita ingresar la fecha con un calendario, he descargado la librería JCalendar. Este componente permite ingresar una fecha de forma gráfica.



Para mejorar la apariencia de la aplicación he importado a mi proyecto la librería Flatlaf, que es un moderno Look and Feel multiplataforma de código abierto para aplicaciones de escritorio Swing. Flatlaf tiene una gran variedad de temas que permiten mostrar una interfaz gráfica más moderna y limpia.

#### f. Pruebas de Software.

- Plan de Pruebas: Al principio se desarrollará una planificación general, que quedará reflejada en el "Plan de Pruebas". El plan de pruebas se inicia el proceso de Análisis del Sistema.
- Especificación del diseño de pruebas. De la ampliación y detalle del plan de pruebas, surge el documento "Especificación del diseño de pruebas".
- Especificación de un caso de prueba. Los casos de prueba se concretan a partir de la especificación del diseño de pruebas.
- Especificación de procedimiento de prueba. Una vez especificado un caso de prueba, será preciso detallar el modo en que van a ser ejecutados cada uno de los casos de prueba, siendo recogido en el documento "Especificación del procedimiento de prueba".
- Registro de pruebas. En el "Registro de pruebas" se registrarán los sucesos que tengan lugar durante las pruebas.
- Informe de incidente de pruebas. Para cada incidente, defecto detectado, solicitud de mejora, etc., se elaborará un "informe de incidente de pruebas".
- Informe sumario de pruebas. Finalmente, un "Informe sumario de pruebas" resumirá las actividades de prueba vinculadas a uno o más especificaciones de diseño de pruebas.

Caso de prueba	
Id de prueba	1
Descripción de la prueba	Actualizar pedidos
Tipo de prueba	Funcional
Prioridad	Alta
Condiciones previas	Base de datos MarketBeezUp con pedidos registrados.
Datos de prueba	Un nuevo pedido con 2 artículos
Pasos a ejecutar	Se introduce el nuevo pedido en la base de datos online simulando su registro.  Luego se inicia la aplicación para que actualice los pedidos y lo muestre.
Resultado esperado	Debe mostrar un mensaje de actualización de pedidos y lo mostrará en la tabla de pedidos.
Resultado real	La prueba tiene éxito y la aplicación muestra el mensaje “Se ha encontrado un nuevo pedido” y lo muestra en la tabla de pedidos

Caso de prueba	
Id de prueba	2
Descripción de la prueba	Nuevo Envío
Tipo de prueba	Funcional
Prioridad	Alta
Condiciones previas	Base de datos MarketBeezUp con pedidos registrados.
Datos de prueba	Pedido sin envío registrado.
Pasos a ejecutar	Se selecciona el pedido en la tabla de pedidos.  Se ejecuta la funcionalidad Nuevo Envío.  Se introducen los datos del envío.
Resultado esperado	Los datos se registran en el pedido.
Resultado real	La prueba tiene éxito y se registra el envío en el pedido.

Caso de prueba
----------------

Id de prueba	3
Descripción de la prueba	Nuevo Envío con fecha anterior a la actual
Tipo de prueba	Funcional
Prioridad	Alta
Condiciones previas	Base de datos MarketBeezUp con pedidos registrados.
Datos de prueba	Pedido sin envío registrado.
Pasos a ejecutar	Se selecciona el pedido en la tabla de pedidos. Se ejecuta la funcionalidad Nuevo Envío. Se introducen los datos del envío con una fecha de envío anterior a la fecha actual.
Resultado esperado	Debe mostrarse un mensaje de error y esperar a que el usuario modifique la fecha de envío.
Resultado real	La prueba falla y la aplicación registra el envío con la fecha incorrecta.

Caso de prueba	
Id de prueba	4
Descripción de la prueba	Impresión de albaranes
Tipo de prueba	Funcional
Prioridad	Alta
Condiciones previas	Base de datos MarketBeezUp con pedidos registrados.
Datos de prueba	Varios pedidos con envío registrado y sin fecha de salida.
Pasos a ejecutar	Se ejecuta la funcionalidad Impresión Albaranes Envío. Se selecciona Imprimir. Se cierra la ventana de informe con los albaranes generados. Se hace clic en Marcar.
Resultado esperado	Debe mostrar la lista de pedidos con fecha de salida, que el almacén no contenga la palabra DROP y sin fecha de impresión. Cuando se seleccione Marcar, debe aparecer la fecha y la hora de impresión en la tabla.

Resultado real	La prueba tiene éxito.
----------------	------------------------

## 6. DEFINIR PROCEDIMIENTOS PARA EVALUAR EL PROYECTO CON LOS CLIENTES. ELABORAR DOCUMENTACIÓN PARA EL CLIENTE.

El **procedimiento para evaluar el proyecto** con el cliente contará con los siguientes procesos:

- Creación de la base de datos en el servidor del cliente: se ejecuta el script diseñado para crear la base de datos con todas las tablas necesarias en el servidor del cliente.
- Instalación de la aplicación en los equipos del cliente: la aplicación se transfiere a los ordenadores del cliente y se configura en modo de prueba.

El cliente está presente en este momento y le comentamos cómo se va planteando la instalación.

- En este momento, se llevan a cabo las Beta Test, que son las últimas pruebas que se realizan en los propios equipos del cliente y bajo cargas normales de trabajo.
- Una vez instalada la aplicación, pasamos a la fase de configuración.

En ella, asignamos los parámetros de funcionamiento normal de la empresa y probamos que la aplicación es operativa.

- Una vez se ha configurado, el siguiente y último paso es la fase de producción normal. La aplicación pasa a manos de los usuarios finales y se da comienzo a la explotación del software.

### MANUAL DE INSTALACIÓN:

La aplicación funciona con OpenJDK 17 o superior, se puede descargar desde <https://jdk.java.net/archive/>.

En el Anexo II incluyo los pasos necesarios para instalar OpenJDK en Windows.

La aplicación se entrega en un archivo comprimido que contiene el archivo Jar ejecutable y una carpeta con las librerías necesarias. Este archivo comprimido lo extraemos a un directorio y ejecutamos el archivo Jar para iniciar la aplicación.

La aplicación conecta con dos bases de datos, una que llamo *online* de donde lee los pedidos que se obtienen a diario de los diferentes Marketplace, y otra base de datos,

que llamo *marketbeezup*, donde almacena los pedidos junto con la información que se irá añadiendo para gestionar el envío del pedido.

He incluido dos scripts, *marketbeezup.sql* y *pedidos.slq*, para crear las dos bases de datos en el entorno de desarrollo y pruebas. En el entorno real de producción la base de datos Pedidos ya existe en un servidor y los pedidos entran mediante una integración que ya está desarrollada. La base de datos *marketbeezup* es necesario crearla en el servidor para que pueda funcionar la aplicación.

Cuando inicia la aplicación por primera vez, está configurada con unos parámetros por defecto para acceder a las dos bases de datos. Si la aplicación no puede conectar a ambas bases de datos nos dará el mensaje correspondiente. Luego podemos seleccionar el menú **Archivo > Configurar** para configurar los parámetros de conexión.

### **MANUAL DE USUARIO:**

La pantalla principal de la aplicación muestra el menú, la barra de herramientas de accesos rápidos y la tabla de pedidos.

Cuando se inicia la aplicación, los pedidos nuevos se registran automáticamente en la base de datos. La tabla de la pantalla principal muestra para cada pedido toda la información correspondiente: datos del pedido, datos de cada artículo del pedido y los registros correspondientes de envíos, compras, impresiones, etc.

Cada fila puede tener un color de texto diferente que informa del estado del pedido:

- Color azul: hay un pedido con varias líneas.
- Color rojo: el pedido ha sido anulado.
- Color verde: el pedido ha sido enviado.
- Color naranja: el pedido está pendiente de envío.
- Color negro: el estado es diferente de los anteriores

#### **1. Configurar la aplicación:**

Para configurar los parámetros de la aplicación debe seleccionar el menú **Archivo > Configuración**. En esta ventana puede configurar los parámetros de conexión con las bases de datos. También puede configurar el número de días anteriores a la fecha actual para filtrar los pedidos a mostrar en la ventana principal (90 días por defecto). Este valor influye en el tiempo de carga de los pedidos en pantalla principal.

#### **2. Filtrar pedidos:**

Puede filtrar los pedidos mostrados en la tabla seleccionando el menú **Editar > Filtro**

**Pedidos** o a través del **icono Filtro** de la barra de herramientas. Se puede filtrar seleccionando cualquiera de los valores o una combinación de varios. En las listas puede seleccionar varios valores dejando pulsada la tecla Ctrl y haciendo clic en cada valor a seleccionar.

### **3. Buscar pedido:**

También puede realizar **búsquedas de pedidos** por valores concretos: La **lista desplegable** situada en la barra de herramientas sirve para seleccionar el atributo de búsqueda y luego debe introducir el valor a buscar en el campo de búsqueda.

Para volver a mostrar todos los pedidos conforme el criterio de los filtros seleccionados puede hacer clic en el icono **Filtrar** en la barra de herramientas.

### **4. Registro Agencias de transporte, Almacenes y Provincias:**

Para registrar o eliminar una agencia de transporte seleccione **Editar > Agencia de transporte**. Para registrar una nueva agencia introduzca el nombre en el campo Agencia y seleccione **Registrar**. Una vez registrada la agencia se mostrará en la lista de la ventana. Para eliminar una agencia seleccione la fila correspondiente en la lista y luego **Eliminar**. También puede buscar una agencia determinada introduciendo su nombre en el campo de búsqueda.

Los almacenes y provincias se pueden registrar, editar o eliminar de una forma similar a las agencias de transporte. Puede acceder seleccionando **Editar > Almacenes** o **Editar > Provincias**. La diferencia es que en los almacenes puede modificar el nombre de un almacén registrado: Se puede hacer doble clic en el valor en la tabla y editarlo.

También puede modificar el nombre de una provincia de forma análoga a los almacenes.

### **5. Gestión de pedidos:**

Para registrar un nuevo pedido seleccione **Pedidos > Nuevo**. Primeramente, debe introducir todos los valores del pedido. Para cada artículo a registrar debe introducir los valores y seleccionar **Registrar**. En la tabla de la ventana Pedidos se muestra la lista de artículos que se van registrando. Para editar o borrar un artículo, debe seleccionarlo en la tabla y luego hacer clic en **Editar o Borrar**. Una vez editado el artículo seleccione **Registrar** para guardar los cambios. Para registrar el pedido junto con los artículos registrados, debe seleccionar **Guardar**.

Para editar un pedido registrado, seleccione el pedido en la tabla de pedidos y seleccione el menú **Pedidos > Editar**.

Para eliminar un pedido registrado, seleccione el pedido en la tabla de pedidos y seleccione el menú **Pedidos > Eliminar**.

Para marcar un pedido como anulado, debe seleccionar el pedido en la tabla de pedidos y luego el menú **Pedidos > Anular**.

Los cambios introducidos en los pedidos se mostrarán en la tabla de pedidos.

#### **6. Gestión de envíos:**

Para registrar el envío de un pedido, primero debemos seleccionar el pedido en la tabla de pedidos y luego seleccionar el menú **Envíos > Nuevo** o hacer clic en el **icono Envío** de la barra de herramientas. Luego introducimos la fecha de salida, el almacén y la agencia y hacemos clic en **Guardar**.

Podemos editar un envío registrado seleccionando el pedido en la tabla y luego **Envíos > Editar**. O podemos eliminar un envío registrado seleccionando el pedido correspondiente y luego **Envíos > Eliminar**. Si el pedido ya tenía registrada una fecha de impresión de albarán, también se eliminará cuando se elimine el envío.

Los cambios introducidos en los envíos se mostrarán en la tabla de pedidos.

#### **7. Gestión de compras:**

Para registrar una compra en un pedido, primero debemos seleccionar el pedido en la tabla de pedidos y a continuación seleccionar el menú **Compras > Nuevo** o hacer clic en el **icono Compra** de la barra de herramientas. Luego introducimos el identificador de compra, el proveedor, la fecha de compra, la fecha de entrada estimada y hacemos clic en **Guardar**.

Podemos editar una compra registrada seleccionando el pedido en la tabla y luego **Envíos > Editar**. O podemos eliminar un envío seleccionando el pedido correspondiente y luego **Envíos > Eliminar**.

Los cambios introducidos en las compras se mostrarán en la tabla de pedidos.

#### **8. Gestión de Tickets, Facturas y Albaranes de Venta.**

El registro, edición y eliminado de facturas y albaranes de venta se hace a través del menú **Ventas** de una forma análoga a la gestión de compras.

Si desea obtener **tickets de venta** debe seleccionar **Ventas > Obtener Ticket**. Luego debe introducir la fecha inicial, la fecha final y el Marketplace y hace clic en Obtener. En la ventana de selección de destino debe seleccionar la carpeta donde guardará los tickets generados.



## 9. Impresión y reimpresión de Albaranes de envío.

Para imprimir los albaranes de envío debe seleccionar el menú **Albaranes > Imprimir Albaranes Envío**. En la ventana de impresión se mostrarán los pedidos del día que tienen registrado un envío, no son dropshipping y aún no tienen marcada la fecha de impresión. El listado de pedidos a imprimir se puede filtrar por fecha, agencia de transporte o identificador de pedido. Para filtrar debe introducir los valores para filtrar y hacer clic en el icono **Filtrar**.

Para imprimir la lista mostrada debe hacer clic en el icono **Imprimir**. Cuando haya verificado la correcta impresión de los albaranes, o los haya guardado, seleccione **Marcar** para registrar la fecha de impresión.

Para reimprimir albaranes ya impresos (tienen fecha de reimpresión marcada), debe seleccionar el menú **Albaranes > Reimprimir Albaranes Envío**. El listado de albaranes se puede filtrar de manera análoga a la impresión de albaranes. Para reimprimir un albarán debe seleccionarlo en la lista y hacer clic en el icono **Imprimir**. Para seleccionar varios albaranes mantenga presionada la tecla **Ctrl** mientras los selecciona.

## 10. Observaciones.

Para registrar una o varias observaciones, primeramente, debe seleccionar un pedido en la tabla de pedidos y luego seleccionar el menú **Observaciones > Obtener**.

Luego introduzca el título de la observación, la descripción y seleccione **Registrar**.

La observación aparecerá en el listado de observaciones.

La descripción de una observación se puede editar haciendo doble clic en el texto de la lista.

Para eliminar una observación debe seleccionarla en la lista y hacer clic en **Eliminar**. También se pueden buscar observaciones por título introduciendo el texto a buscar en el campo de búsqueda.

## 7. CONCLUSIONES

### a. Dificultades encontradas y soluciones adoptadas.

La parte más complicada de este proyecto ha sido el filtrado de la tabla de pedidos que aparece en la pantalla principal de la aplicación. Como solución he optado por concatenar a cada consulta las cláusulas WHERE necesarias para realizar el filtrado requerido. Esta solución se podrá mejorar mediante el uso de algún framework como

puede ser Hibernate. A continuación, detallo un ejemplo de la solución adoptada:

```
public static String obtenerConsultaPedidos(Filtro filtro) {
    StringBuilder consulta = new StringBuilder();
    StringBuilder predicadoAux;
    List<String> predicados = new ArrayList<>();
    if (filtro.isObservaciones()) { // Registros con observaciones almacenadas
        consulta.append(CONSULTA_PEDIDOS_SOLO_CON_OBSERVACIONES);
    } else { // Registros con y sin observaciones registradas
        consulta.append(CONSULTA_PEDIDOS);
    }
    if (filtro.getMarketplace() != null) { // Si hay market seleccionado se añade el
predicado correspondiente
        predicadoAux = new StringBuilder();
        predicadoAux.append(" pedidos.marketplace IN (?)");
        for (int i = 1; i < filtro.getMarketplace().length; i++) {
            predicadoAux.append(", ?"); // Se añaden tantos parámetros como nº de
market
        }
        predicadoAux.append(")");
        predicados.add(predicadoAux.toString());
    }
    if (filtro.getTiendas() != null) {
        predicadoAux = new StringBuilder();
        predicadoAux.append(" pedidos.tienda IN (?)");
        for (int i = 1; i < filtro.getTiendas().length; i++) {
            predicadoAux.append(", ?");
        }
        predicadoAux.append(")");
        predicados.add(predicadoAux.toString());
    }
    ...
}
```

#### b. Propuestas de mejora del proyecto.

Una posible funcionalidad es que la aplicación pueda registrar los datos relativos a los rappels e incentivos de venta. Para los rappels sería necesario registrar: familia, subfamilia, marca y porcentaje. Para los incentivos de venta se registraría: código de artículo, proveedor, importe, fecha de inicio y fecha de fin de la promoción.

Mediante esta nueva funcionalidad el usuario podría obtener un informe de rentabilidad para cada pedido mediante la siguiente fórmula:

$$\text{Beneficio} = \text{Importe} - (\text{PUC} + \text{Porte} + \text{Comisión} - \text{Rappel} - \text{Incentivo})$$

PUC es el precio de última compra, que habría que obtener del ERP. El porte y la

comisión ya los tenemos registrados en la base de datos de pedidos.

#### c. Ofertas de mantenimiento.

Se ofrece al cliente un plan de mantenimiento que puede incluir los siguientes servicios:

- Corrección de los errores que no se hayan detectado durante la fase de pruebas.
- Mejorar el rendimiento.
- Mejorar el diseño.
- Mantener actualizada la aplicación para que funcione con los diferentes sistemas con los que interactúa.
- Mantener actualizadas las librerías de la aplicación para que funcione con las actualizaciones que se produzcan en el sistema operativo sobre el que funciona.
- Programación de nuevas funcionalidades a medida.
- Realización de copias de la base de datos y restauración en caso de ser requerido.

En función de los servicios que necesite el cliente se elaborará un presupuesto detallado.

### **8. REFERENCIAS BIBLIOGRÁFICAS.**

<https://www.ine.es>

<https://www.bbvaresearch.com/publicaciones/situacion-espana-segundo-trimestre-2022/>

<http://www.ipyme.org/es-ES/creaciondelaempresa/Paginas/Tramites.aspx>

<http://www.ipyme.org/Publicaciones/SCivilCreacion.pdf>

[https://portal.mineco.gob.es/RecursosNoticia/mineco/prensa/noticias/2021/210127\\_np\\_pyme.pdf](https://portal.mineco.gob.es/RecursosNoticia/mineco/prensa/noticias/2021/210127_np_pyme.pdf)

[https://www.fauca.org/wp-content/uploads/2017/10/manual\\_prl-oficinas-y-despachos-MC-Mutual.pdf](https://www.fauca.org/wp-content/uploads/2017/10/manual_prl-oficinas-y-despachos-MC-Mutual.pdf)

<https://www.formdev.com/flatlaf/>

<https://www.mysql.com/>

<https://www.jaspersoft.com/es>

<https://www.oracle.com/es/index.html>

<https://toedter.com/jcalendar/>

## **9. ANEXOS.**

### **Anexo I. Albarán.**

El albarán debe incluir el logotipo de la tienda que realiza el pedido. La empresa trabaja con varias tiendas, y cada una tendrá su logotipo. En el pie del documento debe incluir los datos de la tienda: dirección web, CIF, dirección fiscal y teléfono. El albarán debe incluir la fecha de pedido, el identificador del Marketplace o Web, el número de pedido, el almacén de salida, la agencia de transporte, los datos del cliente, los datos del producto y la cantidad. La cantidad es importante destacarla para que el personal de almacén no cometa errores al preparar el pedido.



# Logotipo

31 marzo 2022

## Albarán de entrega

FAN - 002Y11Y19B020

**Manuel Rivallo Bejarano**

Ronda del Serrallo 85

13170 Miguelturra

Número pedido: 002Y11Y19B020

Fecha pedido: 31 marzo 2022

REFERENCIA	DESCRIPCIÓN	UNIDADES
6927828020833	Frigorífico Balary 3TS333WT – Blanco, 199cm, Clase C	<b>1</b>

003 - SEUR

[www.electrodomestico.es](http://www.electrodomestico.es)

Electrodoméstico SL – F11111111

Ctra. Gorrión 33 - 13001 - Ciudad Real - 611 333 489

## Anexo II. Instalar OpenJDK 17 en Windows 10

- Descomprimir el paquete de Java OpenJDK en una carpeta de la unidad C. Ej.:

c:\Program Files\jdk-17.0.2

- Crear esta variable de entorno:

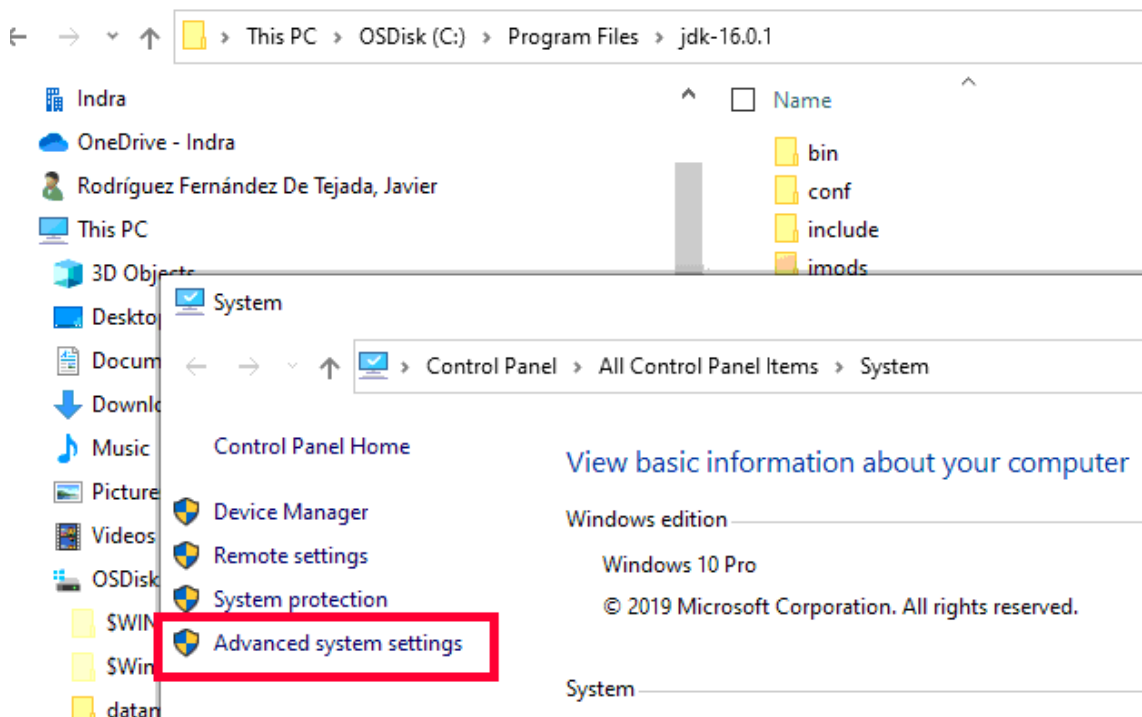
JAVA\_HOME=c:\Program Files\jdk-17.0.2

- Añadir a la variable PATH la ruta:

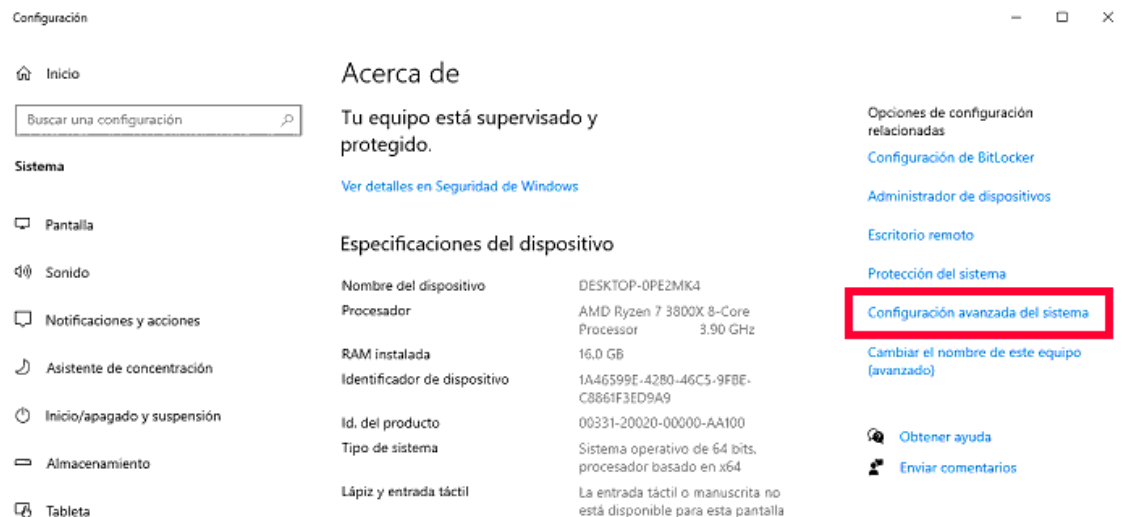
c:\Program Files\jdk-17.0.2\bin

Hay 3 formas posibles para llegar a la ventana de Windows 10 donde podemos crear las variables de entorno.

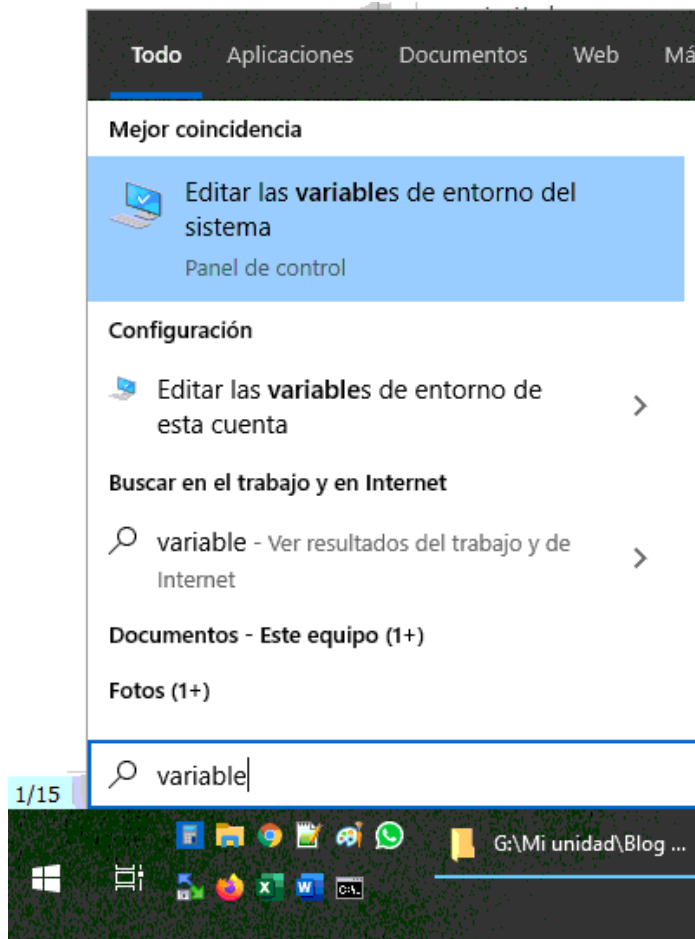
- Desde Mi PC > botón derecho del ratón > propiedades > Y en la ventana que se muestra, **Configuración avanzada del sistema**. (Ventana de tipo 1)



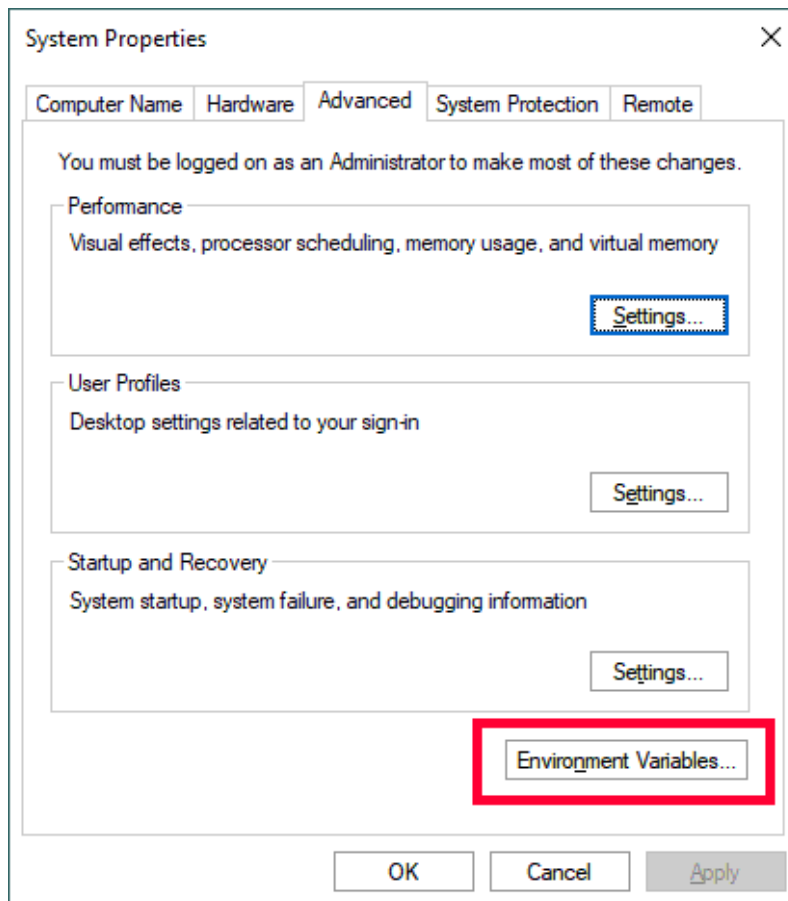
- Desde Mi PC > botón derecho del ratón > propiedades. Y en la ventana que se muestra, **Configuración avanzada del sistema**. (Ventana de tipo 2)



- Pinchar en el menú inicio de Windows 10. Escribir la palabra "variable", y se nos debe mostrar la opción de **Editar las variables de entorno del sistema**.



Cualquiera de las tres opciones posibles, nos llevan a esta ventana, donde seleccionaremos **Variables de entorno**.

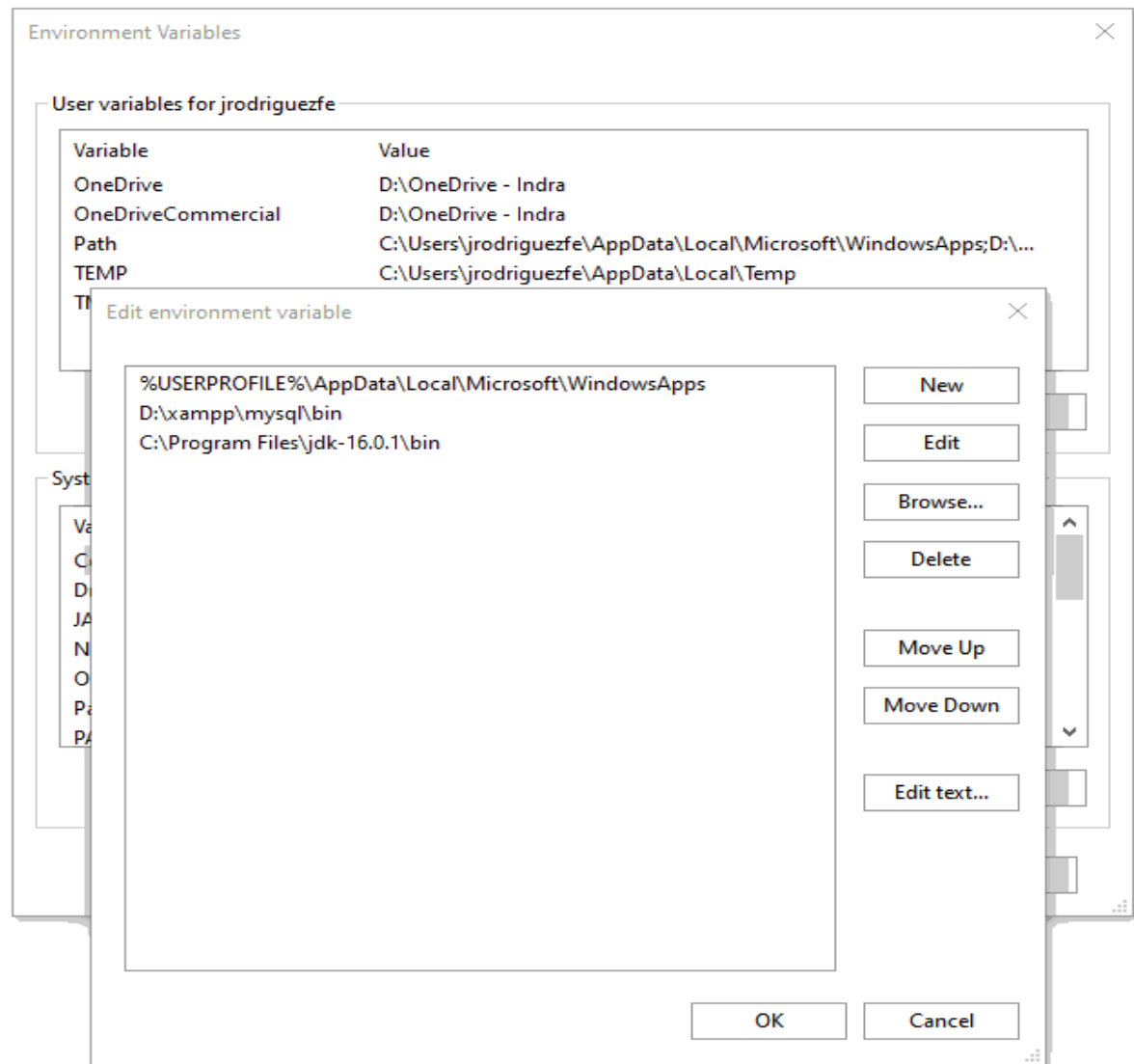


Ahora crearemos la variable de entorno antes mencionada. Le damos a nueva, e introducimos el nombre de la variable **JAVA\_HOME** y la ruta donde lo hemos descomprimido.

Ahora debemos editar la variable de entorno **PATH** por si queremos hacer uso de Java desde cualquier carpeta de MS-DOS. Para ello, editamos la variable PATH, y añadimos un valor más al listado que ya tenga, con la misma ruta que hemos puesto en JAVA\_HOME añadiendo la carpeta \bin. Ejemplo:

c:\Program Files\jdk-17.0.2\bin





Para verificar que todo ha ido bien, abrimos una ventana de MS-DOS y escribimos:

`java -version`

Y nos debe mostrar la versión de Java OpenJDK que hemos instalado. Algo así:

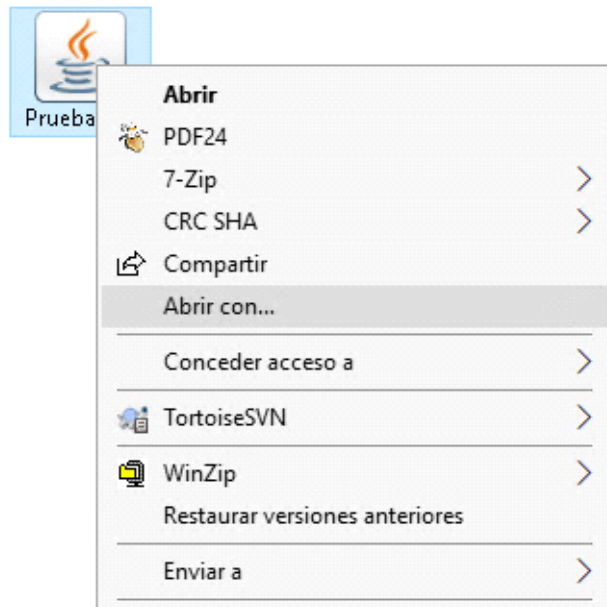
```
Command Prompt
Microsoft Windows [Version 10.0.18363.1500]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\jrodriguezfe>java -version
openjdk version "16.0.1" 2021-04-20
OpenJDK Runtime Environment (build 16.0.1+9-24)
OpenJDK 64-Bit Server VM (build 16.0.1+9-24, mixed mode, sharing)

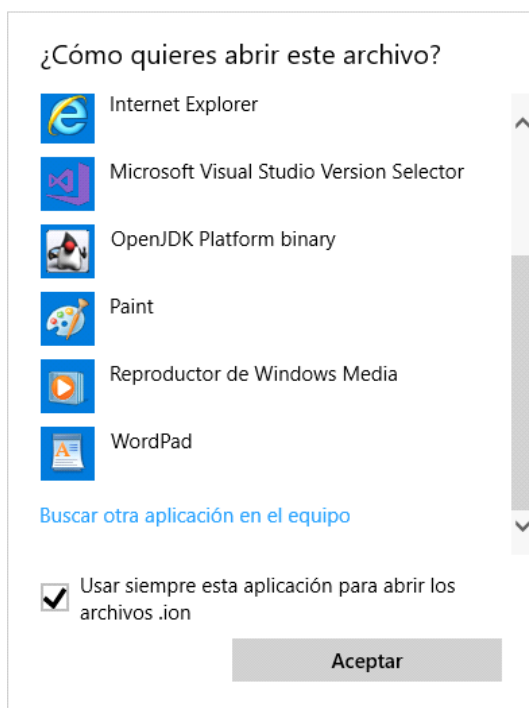
C:\Users\jrodriguezfe>
```

**Para que la aplicación se pueda abrir haciendo doble clic sobre el archivo .jar:**

Pulsamos el botón derecho sobre el archivo .jar y hacemos clic en **Abrir con...**:



Ahora marcamos la opción **Usar siempre esta aplicación para abrir los archivos .jar** y seleccionamos **Buscar otra aplicación en el equipo**. Luego navegamos hasta la carpeta donde se encuentra el paquete OpenJDK, en este ejemplo c:\Program Files\jdk-17.0.2\bin, y seleccionamos el archivo javaw.exe para ejecutar el archivo .jar (para las aplicaciones de consola deberíamos seleccionar el archivo java.exe).



Con esto debería asociarse los archivos .jar con Java. También debemos verificar que el registro de Windows tenga los valores correctos.

Windows Registry Editor Version 5.00

```
[HKEY_CURRENT_USER\SOFTWARE\Classes\.jar]
@="jarfilelocal"
```

```
[HKEY_CURRENT_USER\SOFTWARE\Classes\jarfilelocal]
@="Executable Jar File Local"
```