

C.P.R. Liceo “La Paz”

Proyecto Fin de Ciclo

Desarrollo de Aplicaciones Multiplataforma

Autor : Gabriel Rodríguez Díaz  
Tutor: Jesús Ángel Pérez-Roca Fernández

ÍNDICE

RESUMEN

La aplicación consiste en una solución **multiplataforma**. **Escalable** y **modular** para una empresa dedicada a venta de productos, ya sea de forma mayorista o minorista. También sirve para empresas cuyo modelo de negocio sea la fabricación de productos.

Aunque se puede utilizar cualquiera de sus módulos de forma aislada si así se desea.

La aplicación cuenta con módulos de gestión de productos distribuidos en secciones, lo que hace posible su funcionamiento aunque la empresa entre en nuevos mercados.

Respecto a los productos, se almacenan su precio, su coste y su tipo de IVA; facilitando así la contabilidad. También se almacenan otros datos relevantes: referencia, modelo…

Se almacenan pedidos, donde cada línea contiene las unidades del producto en el pedido, el cliente y el día en que lo ha realizado.

Se puede mantener la jerarquía de la empresa mediante los perfiles. Un empleado tiene asignado un perfil y las funciones que podrá desempeñar en la aplicación variarán en función del perfil que tenga asignado.

Mediante la aplicación se puede consultar la información en gráficos y también se pueden generar reportes y facturas.

La base de datos cumple con la [RLOPD artículo 93.3](https://www.boe.es/buscar/pdf/2008/BOE-A-2008-979-consolidado.pdf) asegurando así la confidencialidad de las credenciales almacenadas en la base datos.

PALABRAS CLAVE

**Escalabilidad:** Que la aplicación sea escalable significa que es capaz de soportar una serie de cambios en el modelo de negocio.

A nivel técnico Esto implica que cualquier campo de selección dentro de un formulario recogerá información de base de datos.

[**JavaFX**](https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784)**:**  Consiste en una tecnología que permite integrar una serie de paquetes multimedia en tu aplicación Java. Es una alternativa más estética (aunque menos completa) a su precursor, Java Swing\*[[1]](#footnote-1).

Como su nombre indica, se utilizan archivos XML (.fxml) para diseñar las GUI. También permite el uso de CSS.

**SceneBuilder:** Herramienta complementaria a JavaFX que permite diseñar la GUI de forma gráfica arrastrando componentes.

**JFoenix:** Librería de JavaFX que implementa componentes basados en el estándar [Material Design](https://material.io/design/introduction) de Google.

**SQL Server:** Es un sistema gestor de base de datos relacionales desarrollado por Microsoft y que utiliza el lenguaje TransactSQL\*[[2]](#footnote-2). El lenguaje TSQL nos permite añadir lógica de negocio en la base de datos, lo que lo hace un gestor muy potente.

En este caso se han explotado funciones propias de SQL Server mediante procedimientos almacenados.

**Procedimiento almacenado:** Consiste en una función almacenada en la propia base de datos que se ejecuta utilizando el motor de la misma.

**Singleton\***[[3]](#footnote-3)**:** Es un patrón de diseño que nos permite crear una instancia de un objeto durante toda la vida de la aplicación. En este casi se ha utilizado para almacenar de forma global la información del usuario que se ha logeado.

[**DAO**](https://www.oscarblancarteblog.com/2018/12/10/data-access-object-dao-pattern/#:~:text=Para%20esto%2C%20tenemos%20el%20patrón,al%20resto%20de%20la%20aplicación.)**:** Consiste en un patrón de diseño que propone la división de la lógica de negocio de la lógica de acceso a datos. Dejando así un código más limpio donde la clase DAO se encargaría exclusivamente de las operaciones CRUD trabajando con POJOs.

[**Hashing & Salting**](https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/)**:** Son dos técnicas de criptografía. El hashing consistiría en aplicar una función de cifrado a una cadena de caracteres.

Ésta podría ser descifrada mediante Rainbow Tables, con lo cual se le aplica Salting: Es decir, a la cadena inicial se le concatena otra cadena aleatoria para cifrar aún más la información.

**[Overload](https://www.sitepoint.com/quick-guide-to-polymorphism-in-java/" \l ":~:text=Static%20Polymorphism%3A,%2Forder%2Fnumber%20of%20parameters.&text=For%20the%20third%20and%20fourth,change%20of%20order%20of%20parameters.)**: Es una característica de Java que nos permite alcanzar el polimorfismo estático.

En la aplicación se explota especialmente en los POJOs de la clase DAO. Permitiendo así utilizar distintas instancias de los POJOs en función de los parámetros requeridos por la lógica de negocio.

MOTIVACIÓN

Tanto siendo alumno de FP medio como FP superior realicé mi FCT en una consultora tecnológica de Viveiro, VegaGestión S.L.

Allí fue donde vi por primera vez un ERP funcionando y descubrí la capacidad que tenían de aumentar la productividad de una empresa.

El ERP que se utiliza allí es Ahora ERP. El problema que veo en muchos ERP es la cantidad de GB que ocupa sólo para instalarlo. Además de eso, al ser herramientas tan potentes, en muchos casos tienen más funcionalidades de las necesarias y se ven tablas vacías en la base de datos.

Con mi aplicación pretendo crear una solución sencilla: Una herramienta que permita una digitalización sin necesidad de utilizar programas con demasiadas funcionalidades y una interfaz en la que los usuarios se pierdan.

De ahí que todos los menús y los controles funcionen igual en todo el programa.

Creo que los desarrolladores debemos estar dispuestos a dar soporte, pero siempre intentar que no sea necesario en ningún caso. Evitemos los errores y las interfaces poco intuitivas.

ESTADO DEL ARTE

La aplicación, a pesar de no ser tan completa como otras soluciones del mercado, no deja de ser un ERP.

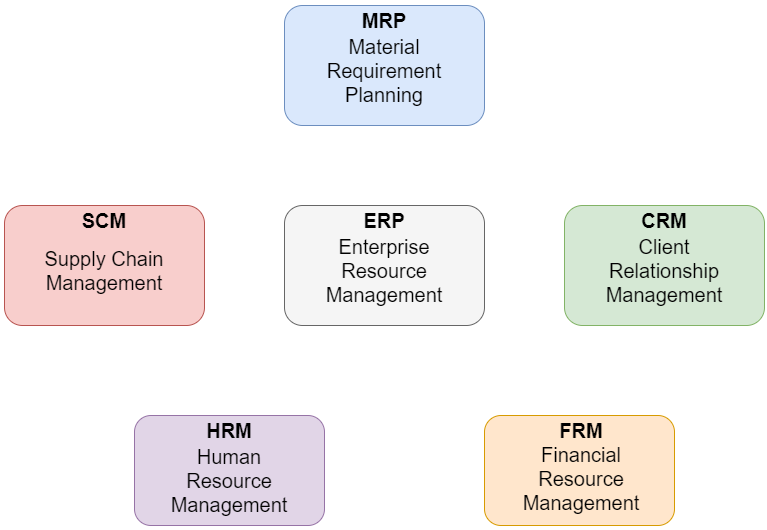


Imagen de elaboración propia usando la herramienta Draw.io

Desde la app se gestionan empleados, precios de venta, costes, pedidos, productos y stocks. Podríamos decir que cuenta con los principales módulos.

Los ERP existen aproximadamente desde los años 90 y en el mercado hay muchas soluciones. Pasando de las más potentes y asentadas como SAP a otras más humildes como Syspro.

Existen incluso ERP web como Flexygo de la empresa valenciana Ahora.

El elemento común en todas estas soluciones es que son herramientas muy pesadas que requieren de técnicos especializados para implementarlas.

La alternativa que ofrece la aplicación aquí expuesta es la sencillez de la misma, la facilidad de instalación con tutoriales guiados y lo ligera que es.

Otras aplicaciones como Ahora ERP trabajan con bases de datos de varios GB y contienen multitud de tablas que no siempre se usan.

VIABILIDAD TECNO-ECONÓMICA

DIAGRAMAS

BIBLIOGRAFÍA

CONCLUSIONES

LÍNEAS FUTURAS DE INVESTIGACIÓN

1. [Comparativa entre Java Swing y JavaFX](https://www.educba.com/java-swing-vs-java-fx/) [↑](#footnote-ref-1)
2. [Comparativa entre SQL y TSQL](https://www.geeksforgeeks.org/difference-between-sql-and-t-sql/) [↑](#footnote-ref-2)
3. Véase “Introducción a los patrones de diseño” de O.J. Blancarte Iturralde. Pág. 68/69 [↑](#footnote-ref-3)