

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, el beneficio obtenido… 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 se 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

**Sistema informático:** Acorde a Sebastian K. Boell y Dubravka Cecez-Kecmanovic, podemos abordar la definición de sistema informático desde cuatro perspectivas distintas:

|  |  |
| --- | --- |
| **Punto de vista** | **Ejemplo de definición** |
| Tecnológico | El sistema utiliza el hardware y el software de la máquina, procedimientos manuales; modelos para el análisis, planificación, control y toma de decisiones; y una base de datos.  Se prioriza su en la informática orientada a las organizaciones. |
| Social | Un sistema informático es un sistema social que se ha incorporado a la informática.  El grado en el que la informática toma parte en la vida está aumentando drásticamente. Y así es que se considera un sistema social y no es posible diseñar un sistema de información robusto y efectivo que con semejante cantidad de tecnología no sea tratado como un sistema social. |
| Sociotécnico | En el campo de la informática se investigan el sistema tecnológico o el social. Incluso ambos separados y los fenómenos que ocurren cuando interactúan entre sí. |
| Procesos informáticos | Un sistema informático es un proceso de sistema cuyos procesos y actividades están orientados al proceso de información: capturar, transmitir, almacenar, recuperar, manipular y mostrar información. |

En este documento nos centraremos especialmente en el punto de vista tecnológico y en el punto de vista de los procesos informáticos.

**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.

**Vista (BD):** Una vista es una tabla virtual formada a partir de de los resultados de una consulta. Al igual que en una tabla normal, sobre ésta se pueden ejecutar queries.

**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 caso se ha utilizado para almacenar de forma global la información del usuario que ha iniciado sesión.

[**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[[4]](#footnote-4), 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/#:~: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 en los POJOs instanciados en para las clases DAO. Se permite 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 VegaGestión S.L., una consultora tecnológica de Viveiro. Esta empresa trabaja con el sistema Ahora ERP.

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

En mi estancia en la empresa descubrí un problema recurrente en los ERP. Este es la cantidad de GB que ocupa tan 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 llevar a cabo una digitalización sin necesidad de utilizar programas con un excedente de funcionalidades y que además tenga una interfaz sencilla e intuitiva.

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.

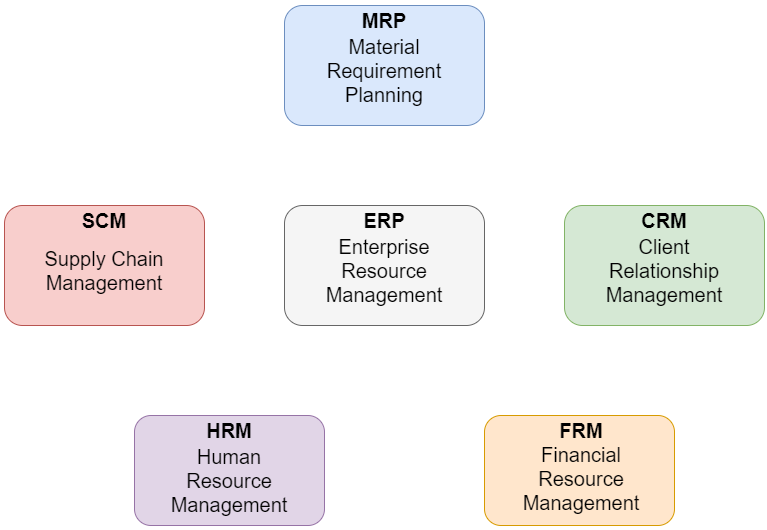


Imagen de elaboración propia usando la herramienta online [Draw.io](https://app.diagrams.net)

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.

Desde la app se gestionan empleados, precios de venta, costes, pedidos, productos y stocks. Así que podríamos decir que cuenta con los principales módulos para empezar a ser considerada un ERP.

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, como ya se ha mencionado en apartados anteriores.

VIABILIDAD TECNO-ECONÓMICA

Figurémonos que este producto se vende con un modelo de negocio similar a una empresa consultora tecnológica.

Supongamos también que el modus operandi (en un supuesto general) de la susodicha empresa es el siguiente:

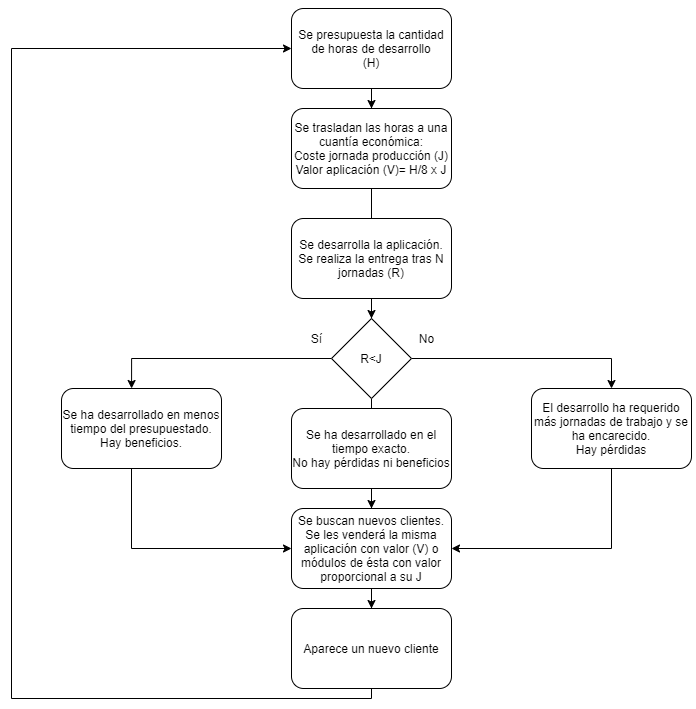


Imagen de elaboración propia usando la herramienta online [Draw.io](https://app.diagrams.net)

Basándonos en el modelo expuesto arriba, la aplicación no tiene por qué ser rentable en primera instancia. Aunque sería lo recomendable para maximizar el beneficio, ésta será rentable a largo plazo. Esto siempre y cuando los clientes demanden aplicaciones similares y se pueda reutilizar código.

Para ello se recomendaría hacer una serie de estudios de mercado previos.

El presupuesto del desarrollo de este programa fue de 26 horas. Tal y como figura en el DOG[[5]](#footnote-5).

Haciendo una serie de búsquedas en páginas de empleo, pongamos, por ejemplo [indeed.com](https://es.indeed.com/career/desarrollador-junior/salaries) vemos que el salario de un programador junior oscila entre los 18.000 y 20.000€ euros brutos al año.

Averigüemos ahora el valor de cada jornada de producción (J)

J = 18.000 ÷ 12 ÷ 30 = 50€

Calculemos ahora el coste figurado de la aplicación (V)

V = 26 ÷ 8 X J = 162,5€

Sin embargo, este programa ha llevado más horas de las presupuestadas, consecuentemente, se buscaría la rentabilidad a largo plazo propiciada por la modularidad en el código de la aplicación.

Estas serían las bases sobre las que se ha trabajado respecto a la viabilidad económica en la producción. Pasemos ahora a otras formas de facturación.

Este producto, recordemos, ofrecido por una consultoría, es un sistema informático. Todo sistema informático es vulnerable y está sujeto a errores en mayor o menor medida.

La consultoría informática es una empresa que, además de crear aplicaciones, lleva el soporte de las mismas.

El modelo de negocio planteado consistiría en una licencia de mantenimiento que se cobraría en todo caso. En esta licencia entraría la solución de errores.

Por otro lado existe la bolsa de horas. Dicha bolsa de horas se vendería a cierto precio, y se consumirían las horas en cada petición de soporte para modificar características de la aplicación.

Con este modelo de negocio ya tendríamos una nueva fuente de ingresos a corto, medio y largo plazo, pues funcionaría durante todo el ciclo de vida de la aplicación.

Otra fuente de ingresos serían las formaciones. Con la aplicación se entregarán unos cursos en vídeo donde se explica cómo usar la aplicación. Para aquellas personas que requieran de una enseñanza personalizada se ofertarán clases para aprender a manejar el software.

A continuación se desglosarán los requisitos hardware de la implementación. Opcionalmente, la consultora podrá obtener beneficio del coste de los dispositivos.

**Requisitos hardware**

Atendiendo a las recomendaciones mínimas de Microsoft y aumentando el baremo para un rendimiento aceptable.

**Servidor:**

OS: Windows Server 2019

SGBD: SQL SERVER 2017

|  |  |  |
| --- | --- | --- |
| **Componente** | **Características** | **Mínimo recomendado** |
| Memoria RAM | 16GB DDR4 | 6GB |
| SSD  Disco duro | 256GB  1TB | 50GB |
| CPU | Intel Xeon E-2124 3,3GHz  4 núcleos, 4 hilos | CPU x64 2GHz |
| Adaptador de Red | Tarjeta de red 1Gb | Tarjeta de red 1Gb |
| Fuente de alimentación | Fuente de alimentación 650w |
| Refrigeración | 2 ventiladores de 120mm |
| Placa base | Adaptada al socket de la CPU |
| GPU | Integrada de la CPU |
| Caja | Genérica ATX |

Esta sería una configuración recomendada para una máquina que vaya a funcionar como servidor de una aplicación.

El precio de las licencias de Windows Server 2019 y SQL SERVER 2017 oscilaría los 275€ sin IVA

No obstante el precio es muy variable. Seguramente se podrían adquirir claves válidas por menos precios.

El precio de la máquina rondaría los 425€ sin IVA. Cantidad también variable.

El total de la configuración serían 700€ aproximadamente.

**Cliente:** Para el cliente serviría cualquier portátil de uso cotidiano o cualquier ordenador de empresa.

**Viabilidad de cara al futuro:**

La aplicación, como se detallará en el apartado de líneas futuras de desarrollo, contará con un backoffice. Dicho backoffice se presupuestará de una forma más realista y será, por tanto, otra fuente de ingresos del producto.

Otras líneas futuras de desarrollo sería la creación de dos app móviles:

* Venta de una aplicación de Android e IOS orientada a los clientes, desde donde podrían hacer sus pedidos.
* Venta de un port de la aplicación a Android e IOS. El haber usado JavaFX nos facilitará el trabajo, pues la integración con el Proyecto Gluon permite realizar este tipo de tareas con facilidad.

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. Blancarte, O.J. 2016: Introducción a los patrones de diseño: Un enfoque práctico, Autoeditado\*, pp. (68/69), disponible en: [Reactiveprogramming.io](https://reactiveprogramming.io/books/design-patterns/es), [Amazon](https://www.amazon.es/Introducción-los-patrones-diseño-práctico/dp/1539619214) [↑](#footnote-ref-3)
4. Son tablas que contienen los resultados de una función de cifrado (hash) para cada cadena de caracteres. [↑](#footnote-ref-4)
5. [DOG Nº109 Miércoles 8 de junio de 2011. Anexo VI. Tabla 1,Fila 16](https://www.xunta.gal/dog/Publicados/2011/20110608/AnuncioC3F1-250511-2127_es.html) [↑](#footnote-ref-5)