



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA
SUPERIOR
Grado en Ingeniería en Informática



**TFG 20.15 del Grado en
Ingeniería Informática**

**Estudio e implementación de
una herramienta LOW CODE
para una aplicación web**



Presentado por
Antoni Lluís García Expósito
en Universidad de Burgos – 20 de enero
de 2021

Tutores: José Manuel Galán Ordax y
Virginia Ahedo García

D. José Manuel Galán Ordax y D^a Virginia Ahedo García, profesores del departamento de Ingeniería de Organización, área de Organización de Empresas

Expone:

Que el alumno Antoni Lluís García Expósito, con DNI 43145353K, ha realizado el Trabajo final del G^oIng.Informática titulado: Estudio e implementación de una herramienta LOW CODE para una aplicación web.

y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual, Se autoriza su presentación y defensa.

En Burgos a 20 de enero de 2021

Resumen

Este proyecto se basa en la creación de una aplicación para la gestión de un negocio (PYME) de venta al público.

El objetivo es que se pueda acceder a ella mediante web, sin la necesidad de que el usuario tenga que realizar una instalación en su terminal.

Para el desarrollo de la aplicación se ha hecho un pequeño estudio para saber qué tecnología cumple con los requisitos. La escogida ha sido una herramienta low code conocida como OpenXava.

Durante el desarrollo se afrontará la implementación de la aplicación con los requisitos iniciales, así como nuevos requisitos o funcionalidades que vayan surgiendo durante la implementación.

Descriptores

ERP, low code, Terminal Punto Venta, OpenXava, Java, Tomcat, Jaspersoft, github, aplicación web.

Abstract

This project is based on the creation of an application to be used to manage a retail business.

One of the main goals to be achieved is to avoid making the user have to download anything into a device.

The user must be able to access it through the web. Prior to the development of the application a small study has taken place to know which technology meets all the requirements. The chosen technology has been a low code tool known as OpenXava.

During the development the application will be implemented with the initial requirement. During the process new requirements and functions will be taken into account.

Keywords

ERP, low code, Point of Sale Terminal, OpenXava, Java, Tomcat, Jaspersoft, github, web application.

Información Indispensable

La aplicación está pensada para el uso en un solo terminal y por un solo usuario a la vez. Esto se debe a que está pensado para PYMES pequeñas, un sector olvidado por las empresas de software, ya que no le dan los mismos beneficios que a las grandes empresas.

Expongo esto porque a la hora de hacer pruebas por parte del tribunal, si se conectan dos usuarios a la vez puede haber inconsistencias en el tratamiento de datos.

Se añadirá en el apartado de futuros pasos, la implementación de una base de datos más completa y que la aplicación funcione en dos o más equipos a la vez. Pero no es el objetivo de este proyecto.

Por eso, si se quieren hacer pruebas con mayor seguridad, subire la carpeta comprimida del Tomcat, y solo se debaran seguir las instrucciones del readme de GitHub o en los anexos en el despliegue mediante Tomcat, para lanzar la aplicación en local, bajo la direccion de: localhost:8080/formes

Índice de contenido

1- Introducción.....	10
1.1- Introducción general	10
1.2- Estructura de la memoria	11
1.3 Estructura de los anexos	12
2- Objetivos del Proyecto	13
2.1- Objetivos Generales.....	13
2.2- Objetivos Académicos	13
2.3- Objetivos Técnicos.....	14
3- Conceptos Teóricos	15
3.1- Que es la tecnología Low Code	15
3.2- Utilidades de las herramientas Low Code	16
3.3- Características de una plataforma Low Code	17
3.4- Ventajas de usar Low Code(3)	18
3.4- Desventajas de usar Low Code(4)	19
3.5- Cuando y porque escoger una plataforma Low-Code.....	20
4- Estudios Relacionados.....	21
4.1- Aura Quantic(7).....	22
4.2- Mendix(9).....	23
5- Técnicas y herramientas.....	26
5.1- Herramientas.....	26
OpenXava:.....	26
OpenXava Studio:	26
Java.....	26
Apache Tomcat.....	27
JasperSoft Studio.....	27

HSQLDB.....	27
GitHub.....	27
ZenHub.....	28
Zotero.....	28
Word.....	28
5.2- Técnicas.....	29
Metodología.....	29
6- Aspectos Relevantes.....	30
6.1- Motivación personal.....	30
6.2- Elección de la herramienta de desarrollo.....	31
7- Conclusiones y líneas de trabajo futuras.....	34
7.1- Conclusiones.....	34
7.2- Líneas de trabajo futuras.....	35
8- Bibliografía.....	36

Índice de figuras

Ilustración 1 Árbol decisión Low Code.....	22
Ilustración 2 Tipo de Usuario y herramientas.....	23
Ilustración 3 Guía de precios módulos Odoo.....	25
Ilustración 4 Curso OpenXava.....	32

1- Introducción

1.1- Introducción general

En el mundo empresarial, una herramienta clave para el día a día de los negocios, es su aplicación de gestión, ya sea la de gestión de stock, clientes, facturas, empleados o la de venta al público.

Durante muchos años esta gestión se hizo a mano, en cuadernos y archivadores que se amontonaban en el desván. Esto cambió cuando aparecieron los primeros ordenadores y con ellos los programas de gestión que necesitaban instalación y mantenimiento, tanto el programa como los terminales.

Actualmente hay miles de programas de gestión para negocios de venta al público. Los hay de todo tipo y precio, unos más orientados al sector primario, otros al secundario y otros al terciario. Pero todos coinciden en una cosa: son muy complejos. Los desarrolladores de estos programas programan todas las funcionalidades posibles para poder llegar a más público lo cual es una razón comprensible.

Lo anterior puede resultar tan conveniente como inconveniente en ocasiones. No todos los usuarios son expertos en la utilización del ordenador y esto puede provocar que la experiencia del usuario sea mala y un mal feedback compromete a la empresa desarrolladora.

Por eso es importante la creación de un programa de gestión de fácil utilización para los usuarios y qué mejor que una aplicación que no necesita ni siquiera instalación. Simplemente con acceder a una web ya la tenemos desplegada.

En este proyecto se ha desarrollado una aplicación web para la gestión de una PYME principalmente orientada a la venta de ropa, pero adaptable a cualquier otro ámbito. Dicha aplicación contiene las funcionalidades básicas y necesarias para que el usuario pueda gestionar todo su negocio.

La herramienta también es importante ya que si se quisieran usar las técnicas tradicionales de programación necesitaríamos un equipo para el desarrollo, así que optaremos por usar las nuevas tecnologías low code para la realización de la aplicación.

1.2- Estructura de la memoria

- Introducción: Pequeña descripción del proyecto con su contexto social. Contiene la estructura de la memoria y de los anexos.
- Objetivos del proyecto: Se expondrán los objetivos del proyecto, tanto personales como académicos.
- Conceptos teóricos: Explicación de los conceptos necesarios para el entendimiento del proyecto.
- Estudios Relacionados: Se analizarán otras herramientas low code y el motivo por el cual se ha escogido OpenXava.
- Técnicas y herramientas: Se listarán las herramientas y las metodologías usadas en el proyecto.
- Aspectos relevantes: Se verán los aspectos más relevantes que han surgido en el desarrollo del proyecto.
- Conclusiones y líneas de trabajo futuras: Se listarán las ideas futuras del proyecto y como tiene que ir avanzando y se expondrán las conclusiones del proyecto.

1.3 Estructura de los anexos

- Plan de proyecto software: Planificación temporal y viabilidad económica y legal.
- Especificación de requisitos: Objetivos y requisitos establecidos al comienzo del proyecto.
- Especificación de diseño: Recoge los diseños de datos, arquitectónicos y de interfaces.
- Manual del programador: Explica los conceptos más técnicos del proyecto como su instalación y la ejecución.
- Manual de usuario: Es el manual de cómo utilizar la aplicación.

2- Objetivos del Proyecto

En este apartado se detallarán los objetivos que se nos marcamos en el proyecto en cuanto a funcionalidad, los objetivos académicos y los objetivos técnicos.

2.1- Objetivos Generales

- Escoger la herramienta low code para el desarrollo de la aplicación.
- Aprender a usar dicha tecnología.
- Desarrollar una aplicación de gestión ERP.
- Desplegar la aplicación en línea.
- La aplicación debe poder gestionar los siguientes campos:
 - Gestión de los productos.
 - Gestión de empleados y clientes.
 - Apartado de venta al público.
- Implementación de un pequeño MRP

2.2- Objetivos Académicos

Como objetivos académicos me propuse analizar un problema que me surgió y posteriormente descubrir cómo resolverlo de la mejor manera posible. Para ello se debía hacer tanto un análisis del problema como un estudio de mercado.

Una vez hecho este análisis me marqué como objetivo académico aprender un nuevo sistema de programación y un nuevo framework de trabajo. Y para finalizar los objetivos académicos también me propuse aprender cómo se despliega una aplicación web, ya que en el grado no se trabaja con este tipo de aplicaciones.

2.3- Objetivos Técnicos

- Usar OpenXava como framework de programación.
- Programar las funcionalidades de un ERP
- La aplicación debe tener una interfaz *user friendly*.
- Usar GitHub como gestor de versiones.
- Usar Zenhub para la gestión del scrum.
- Almacenar los datos introducidos en una base de datos.
- Usar Zotero como gestor de referencias bibliográficas.

3- Conceptos Teóricos

3.1- Que es la tecnología Low Code

El concepto Low Code se podría traducir como código bajo o, mejor dicho, bajo nivel de código, es decir poca programación. Evitando así siempre que sea posible que el usuario tenga que programar de la manera tradicional.

Por ello el usuario tendrá a disposición un entorno grafico con elementos visuales para desarrollar su aplicación. Este deberá ir arrastrando y soltando elementos para ir desarrollando su aplicación, esta técnica se conoce como drag and drop(1)

En 2014 dos analistas de la agencia de investigación de mercados "Forester Research" acuñaron el termino Low Code en su informe de nuevas tecnologías, pero realmente, dicha tecnología, solo es una evolución de la tecnología ya existente desde los 90' conocida como RAD (Rapid Application Development), Esta se fundamenta en un proceso de desarrollo de software. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE. Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.(2)

3.2- Utilidades de las herramientas Low Code

- **Modelado con métodos gráficos**

Mediante las interfaces gráficas el usuario podrá trabajar de manera intuitiva siguiendo patrones modulares, mediante la técnica mencionada anteriormente "Drag and Drop". Así cada usuario podrá ver casi al instante el resultado final.

- **Trabajar en Cloud**

Casi todas las plataformas dan la posibilidad de manejar y editar sus aplicaciones desde la nube que ya incluye métodos gráficos para que la implementación se haga de manera mucho más rápida.

- **Reutilización**

Una de las premisas más importante en el Low code es el ahorro de tiempo, por eso se da la utilización de plantillas, widgets u otras funcionalidades similares hacen que se reutilicen muchos de los recursos.

3.3- Características de una plataforma Low Code

Actualmente existen muchas plataformas que permiten desarrollar aplicaciones con esta tecnología. Pero para saber cuál se adapta más a nosotros y a nuestro proyecto hay que mirar qué características nos convienen. (3)

- **Despliegue:**

Hay que saber si el desarrollo tiene que ser en local, si lo queremos en la nube o si nos conviene más un híbrido de los dos.

- **Tipo de desarrollador:**

Con qué tipo de desarrollador contamos, si se trata de un desarrollador con conocimientos de programación e interesado en la tecnología RAD o si contamos con un trabajador que tiene conocimientos técnicos o de diseño o por último, si se da el caso de un usuario final que en realidad no tiene ningún conocimiento técnico.

- **Orientación sectorial**

Dependiendo de en qué sector trabajamos hay plataformas que se adaptan más al sector.

- **“Post-Venta”**

Otro factor clave a la hora de elegir es tener en cuenta que la plataforma nos ofrezca un servicio de soporte y atención al cliente adecuado a nuestras necesidades tanto actuales como futuras.

3.4- Ventajas de usar Low Code(3)

- **Velocidad:**

Al evitar en gran parte la programación tradicional, el tiempo a la hora de realizar aplicaciones completas o prototipos se reduce de manera exponencial. Esto ayuda a los programadores a poder centrarse en los requisitos más importantes del proyecto.

- **Sencillez:**

Suelen ser plataformas en las cuales el aprendizaje es intuitivo y carece de la necesidad de conocimientos previos. El código generado se hace de forma automática mediante la ordenación de los elementos gráficos.

- **Reducción de costes:**

Este apartado va ligado al primero, ya que, al reducir el tiempo, el coste se ve disminuido. También se evitan las formaciones de los usuarios por el punto de sencillez.

- **Flexibilidad:**

Las herramientas Low code suelen dar la opción de ser desarrolladas en diferentes entornos adaptándose al que conozcan los usuarios. Además, a la hora de implementar una nueva funcionalidad que sea novedosa será mucho más fácil.

3.4- Desventajas de usar Low Code(4)

- **Velocidad:**

A pesar de que pueda sonar paradójico, la velocidad no siempre es una buena aliada. Apresurar el desarrollo a veces es perjudicial ya que el querer sacar adelante con demasiado ímpetu puede causar fallos en el rumbo inicial de un proyecto.

- **Limitación:**

Está claro que las herramientas Low code están en constante evolución y cada vez hay más tipo de personalizaciones, pero nunca se llegará a ser tan completa como una programación manual, a la hora de diseño o funcionalidad tienen unas limitaciones muy fuertes. Dependerá de la plataforma que se haya contratado que se tengan más o menos estilos, plantillas, funcionalidades.

- **Esperanza de vida:**

Este aspecto negativo se tiene que coger con pinzas a causa de los motivos que ahora se presentarán.

Es verdad que si queremos que una aplicación sea beneficiosa a largo plazo es mejor una aplicación típica, con su equipo de desarrolladores expertos detrás. En cambio, las Low-code están más orientadas a corto/medio plazo. Y es aquí donde se tiene que coger con pinzas ya que según el proyecto o situación es mejor hacer una aplicación con Low code que dure por ejemplo dos años y luego rehacerla o empezar una nueva con las nuevas tecnologías existentes en dicho momento. En contraposición a haber invertido una gran cuantía económica en una aplicación tradicional que dure más tiempo, pero cuya evolución es más cara.

3.5- Cuando y porque escoger una plataforma Low-Code

Haciendo una búsqueda superficial se puede ver qué tipos de usuario usan esta tecnología. Tras dicha búsqueda se puede observar que quienes más tiendan a utilizar las plataformas en cuestión son emprendedores y *bloggers*. A pesar de ello, se ha de decir que a día de hoy cada vez se encuentran más PYMES que cuentan con su aplicación web o móvil, su programa de facturación, sus estudios de mercado y ventas, creadas con estas tecnologías.

Incluso en estos últimos años ha aparecido algún sector público que para realizar alguna tarea específica ha echado mano de una plataforma Low-code.

Dicho esto, se puede deducir que escogeremos una herramienta Low-code cuando nos enfrentemos a una petición o a un trabajo que necesite una entrega rápida, que requiera una transformación o modernización digital inmediata.

Según estudios en el 2024 el 70% de las grandes empresas multinacionales utilizarán esta herramienta, dada su estructura ágil, y a la necesidad de BPNⁱ para el estudio de los mercados y el aumento de beneficios, posicionamiento y evolución empresarial.(5,6)

4- Estudios Relacionados

En el mercado actual existe una infinidad de estas plataformas que ofrecen una implementación en Low code. Dependiendo de las necesidades que tengamos podremos ir filtrando para encontrar la que mejor se adapte a nosotros.

Una cosa que conviene tener clara, es que es posible que la seleccionada no sea la mejor herramienta del mercado, pero sí es la que más se ajuste a los objetivos, el presupuesto, las necesidades o simplemente que sea la que más nos atrae.

Vamos a ver una pequeña lista de diferentes plataformas de Low Code:

4.1- Aura Quantic(7)

Esta plataforma nos ofrece bastantes productos interesantes a la hora de escoger una plataforma, ya que no solo se dedica al Low Code, sino que también se especializa con herramientas No-Code, BPM e incluso con MXDP. En su web nos proporcionan un árbol muy intuitivo para saber cuál de sus productos nos podría interesar más:(8)



Ilustración 1 Árbol decisión Low Code

En cuanto a su interfaz se puede afirmar que es muy completa y que permite una gran variedad de diseños y opciones para la configuración de nuestra aplicación (Formularios en múltiples capas, set de estilos CSS, firmas digitales, creador de informes, etc.). Eso en cuanto a la interfaz, en la de lógica ofrece las funcionalidades más novedosas e interesantes del mercado (flujo de procesos, machine learning, IA, Case Management, etc). Y por último, en la capa de Datos, BBDD, podemos encontrar las características más usadas, modelos entidad relación, modelado gráfico del modelo de datos, Almacenamiento en DFS (Distributed File System), etc.)

Aura Quantic es una alternativa muy interesante para un proyecto grande, si se quiere evolucionar digitalmente para llegar a un gran mercado es una de las mejores alternativas. Además, sus planes no son excesivamente caros de modo que se puede encontrar el básico por 24€ mes, el medio que asciende a los 37€ mes y por último existe la opción de un plan personalizado.

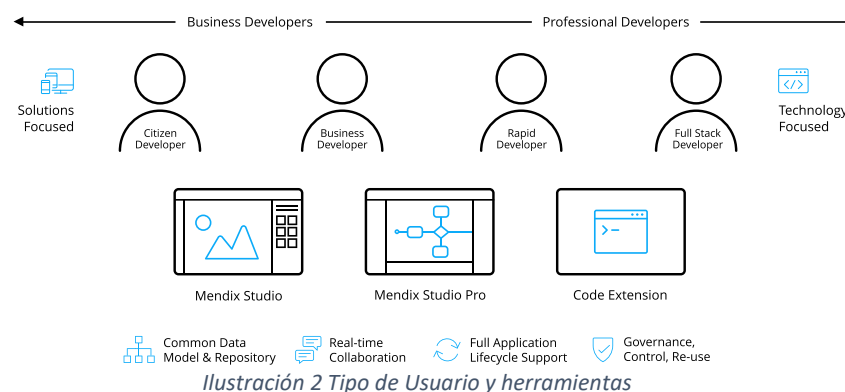
Las diferencias entre los planes son pequeñas funcionalidades y sobre todo afectan al tamaño y al número de usuarios que pueden soportar. Así que, a la hora de escoger, nuestra decisión se debe basar en la necesidad en aquel momento, ya que se puede hacer un cambio de plan en medio del proyecto si se argumenta.

4.2- Mendix(9)

Mendix es otra de las grandes plataformas del Low Code, en este caso sí que se especializa únicamente en crear aplicaciones Low-Code, Así pues, si se escogiese se tendría una plataforma con una gran experiencia en el desarrollo de código bajo.

Es verdad que esta opción requiere unos mínimos conocimientos técnicos lo cual es observable con el simple hecho de navegar por su web, que además es poco intuitiva. Ofrece un producto muy completo con las funcionalidades más necesarias aparte de una personalización muy interesante con dos estudios de diseño para ofrecer la mejor interfaz posible (Mendix Studio y Mendix Studio Pro).

Otro factor a favor de esta plataforma es su funcionalidad de extensión de código para usuarios con un nivel alto de programación, ya que permite retocar el código si fuera necesario. Esta característica es muy importante ya que una de las desventajas del Low Code es que el código autogenerado suele estar escondido y no es posible la vista por parte de los usuarios.



Mendix es la plataforma ideal para un gran abanico de usuarios que va desde los desarrolladores más nuevos que no poseen un gran conocimiento en programación hasta los más expertos programadores.

En cuanto a los precios, Mendix ofrece un plan gratuito, un poco limitado, pero que para iniciarse resulta adecuado. El primer plan básico que se puede adquirir asciende a los 1875€ mensuales y que permite una sola aplicación. En los demás planes no aparece el precio, pero nos podemos hacer a la idea viendo el precio del básico.

El precio puede resultar excesivo para cierto tipo de clientes, como PYMES o autónomos, pero para una empresa grande es una de las alternativas más interesantes del mercado, gracias a su gran soporte técnico.

4.2- Odoo(10)

Si se tuviese que definir Odoo en una palabra seria modularidad. Con esta plataforma nos encontramos con la mayor fuente de distintos módulos para la creación de nuestra aplicación web. Su estilo de trabajo es el de: por qué motivo un usuario tiene que tener todas las funcionalidades si hay más de la mitad que no va a usar. Por eso Odoo ofrece la personalización de las herramientas que queremos y nos permite crear aplicaciones con lo que necesitamos, dejando la puerta abierta a que, si en un futuro necesitamos otra funcionalidad, la implementación sea automática.

Odoo separa sus funcionalidades en 4 apartados, cada uno con su propio eslogan para hacer más atractivo al cliente la contratación.

- **Sitios Web** (Cree una estupenda experiencia para los usuarios)

Se nos permite crear un sinfín de tipos de web distintas, adaptando las necesidades de cada uno de nuestros requisitos, desde blogs, foros, chats en vivo hasta la incorporación de e-commerce y e-learning.

- **Ventas (Aumente sus índices de éxito)**

En caso de necesitar la creación de una aplicación web para la gestión de ventas, productos etc. es sin duda una de las mejores opciones del mercado, ya que nos permite crear un TPV muy completo que cuenta con las últimas tecnologías como la firma digital o la posibilidad de enlazarte con Amazon para poder vender en la mayor tienda del mundo.

- **Operaciones (La clave está en la eficiencia)**

Y no todo es para fin externo, también nos ofrece la opción de poder crear una aplicación para el uso y la gestión interna de la empresa.

- **Herramientas de productividad** (Herramienta estupenda = persona contenta)

Este es el apartado donde más destaca Odoo, ya que es de gran importancia el aumento de la productividad, y contar con herramientas para la realización de estudios de marketing, encuestas o una simple agenda, nos da la oportunidad de crear un producto para destacar por delante de nuestra competencia.

Como hemos hecho anteriormente ahora valoraremos sus planes, pero en este caso, al estar basado en módulos, los planes son altamente personalizables y por tanto se paga únicamente por los que necesites.

CRM 12,00 EUR / mes	Facturación 6,00 EUR / mes	Ventas 6,00 EUR / mes
Web 12,00 EUR / mes	Comercio electrónico 6,00 EUR / mes	Punto de venta 12,00 EUR / mes
Contabilidad 12,00 EUR / mes	Proyecto 12,00 EUR / mes	Inventario 18,00 EUR / mes
Fabricación 24,00 EUR / mes	Compra 6,00 EUR / mes	Hojas de horas 6,00 EUR / mes
Marketing electrónico 6,00 EUR / mes	Gastos 6,00 EUR / mes	Eventos 6,00 EUR / mes
Ausencias 6,00 EUR / mes	Contratación 6,00 EUR / mes	Valoración 6,00 EUR / mes
Suscripciones 12,00 EUR / mes	Firmar 12,00 EUR / mes	Mantenimiento 12,00 EUR / mes
Calidad 12,00 EUR / mes	Studio 36,00 EUR / mes	Servicio de asistencia 12,00 EUR / mes
Product Lifecycle Manag... 12,00 EUR / mes	Citas 6,00 EUR / mes	Automatización de mark... 18,00 EUR / mes
Documentos 12,00 EUR / mes	IoT Boxes 15,00 EUR / month / per box	Aprobaciones 6,00 EUR / mes
Consolidation 24,00 EUR / mes	Recomendación de empl... 6,00 EUR / mes	Servicio de campo 12,00 EUR / mes
Planificación 6,00 EUR / mes	Alquiler 12,00 EUR / mes	Marketing Social 18,00 EUR / mes
Aprendizaje en línea 12,00 EUR / mes		

Ilustración 3 Guía de precios módulos Odoo

5- Técnicas y herramientas

5.1- Herramientas

OpenXava:

Herramienta Open Source basada en la tecnología low code que cuenta con una gran cantidad de documentación oficial y no oficial que puede ser extraída de foros y blogs. En la web oficial nos encontramos con un apartado de documentación que nos permite hacer desde un curso de introducción hasta documentación de cómo configurarse para todo tipo de BBDD (MySQL, PostgreSQL, DB2 entre muchas), a diferentes servidores.

OpenXava Studio:

Entorno de programación de OpenXava, está basado en Eclipse, de hecho, es una versión *lite* del mismo. Ofrece casi todas las funcionalidades de eclipse, pero la diferencia es que ya te viene con todas las funcionalidades de OpenXava instaladas.

Para la descarga de este entorno tienes que registrarte en la web de OpenXava y hacer la petición de descarga, entonces se te enviará un enlace con el archivo.

Java

Es un lenguaje de programación comercializado por primera vez en el 1995 por Sun Microsystems y más tarde en el 2010 fue adquirido por Oracle. Esta adquisición lo propulsó para ser uno de los lenguajes más usados y populares. Sobre todo, usado para la programación de aplicaciones de cliente-servidor.

Java posee una biblioteca de librerías inmensa hecho que le permite ser usada en casi todos los ámbitos de la programación. Es un lenguaje orientado a objetos e independiente a las plataformas, es decir que se puede ejecutar en cualquier plataforma.(11)

Apache Tomcat

Es el servidor web más popular para las aplicaciones escritas en Java, ya que está desarrollado por los mismos desarrolladores. Puede funcionar de manera autónoma, pero se suele utilizar con un servidor web Apache para dar más soporte al usuario y al mismo tiempo aumentar las prestaciones. Es un servicio gratuito y de código abierto

JasperSoft Studio:

Es el software sugerido por el propio OpenXava para la realización de los informes de la aplicación en formato rjxml.

Se puede conectar con la base de datos creada por nuestro proyecto o crear un formulario con parámetros y campos que a posteriori desde el entorno de OpenXava studio se pueden enlazar a los datos que se tengan en la base de datos.

A parte, este software ofrece la posibilidad de crear el diagrama de clases que conforma tu proyecto a modo de ingeniería inversa.

HSQldb

Es un gestor de base de datos relacional escrito íntegramente en java, de hecho, es el gestor por defecto de las aplicaciones escritas en Java.

Es similar a SQLite, ya que se trata de un gestor ligero con un tiempo de arranque mínimo y una gran velocidad en las operaciones SQL básicas. Gracias a la velocidad y a su máximo de 8GB de capacidad la convierten en una base de datos muy usada para aplicaciones simples.

En caso de necesitar más capacidad de BBDD valore usar otras opciones como MySQL o Postgres. (12)

GitHub

Es un portal creado para alojar código de aplicaciones de los programadores del mundo, pero no solo sirve para subir el código, sino que también es útil para saber más sobre la aplicación, documentación, feedback, etc.

Como el propio nombre indica el portal usa la tecnología de control de versiones Git creada por Linus Torvalds, funcionalidad que incrementa su importancia, ya que permite controlar las versiones de tu proyecto. Y en caso de trabajar en equipo, se pueden ver las modificaciones de cada uno de los integrantes.

ZenHub

Se trata de un portal de gestión de proyectos integrado en GitHub en el que se nos permite controlar el avance del proyecto mediante la gestión de las pequeñas tareas en las que separamos nuestro proyecto.

Nos aporta un seguimiento con gráficos y estadísticas de la vida de nuestro proyecto lo cual es muy interesante.

Zotero

Es el gestor de referencias que me ha recomendado mi tutor, ya que es un gestor de referencias de código libre muy completo.

El principal fuerte es la integración con los principales editores de texto como Word, OpenOffice o LaTeX, así se facilita mucho la aplicación de citas y bibliografía a nuestros documentos.

Word

Es el editor de textos por defecto de Microsoft, permite la modificación de documentos y textos, al igual que su personalización. Al ser el procesador de textos mas usado en el mundo, cuenta con todas las funcionalidades necesarias para la escritura.

5.2- Técnicas

Metodología

La metodología escogida ha sido SCRUM, basada en la utilización de una gestión ágil del proyecto. Separando las diferentes partes del proyecto en Sprints.

Para ello se acordó realizar entregas cada 10 o 15 días dependiendo de la dificultad del *sprint*. La dificultad se ha calculado mediante el tiempo que preveo en la realización de cada una de las tareas, este tiempo o dificultad está expresada en los storie points de cada tarea, equivaliendo 1 unidad a una hora.

Cada Sprint cuenta con un nombre propio definido como *milestone* para así saber en qué fase del proyecto estamos de forma más natural.

Otro punto para valorar es que al tratarse de una aplicación con un plazo de entrega relativamente corto, se van a concentrar muchas tareas en cada uno de los *sprints* y no se realizarán tantos como me hubiese gustado.

En cada revisión se hablaba de los problemas que había tenido en el anterior al igual que se planeaba con los tutores el próximo sprint, ya fuese añadiendo funcionalidades o simplemente valorando en qué se podía mejorar el proyecto.

6- Aspectos Relevantes

En este apartado explicaremos los aspectos más relevantes del proyecto, como la motivación personal para la elección de este proyecto, la elección de la herramienta, la formación necesaria para la implementación de la aplicación y la resolución de los problemas técnicos.

6.1- Motivación personal

La elección de mi TFG fue bastante fácil, la única dificultad fue encontrar a los tutores idóneos. La facilidad de la elección viene del punto que en casa tenemos una pequeña tienda de ropa que gestiona exclusivamente mi madre.

En la tienda tenemos un software de gestión que nos ha venido heredado de la antigua propietaria de la tienda, se trata de un software excesivamente completo porque tiene implementadas funcionalidades que nunca se han usado o que ni entran en nuestro sector.

En cuanto a las funcionalidades que sí que se utilizan destaca lo complejo que resulta su uso por la cantidad de opciones que ofrece. Esto desde el punto de vista de un usuario que tiene buen nivel de informática es positivo, pero hay que tener en cuenta que pocos poseen dicho nivel.

Por eso se me ocurrió proponer una aplicación pensada para pequeños negocios de barrio que permita que los usuarios que no tengan una base de informática muy elevada puedan realizar un buen uso de ella.

Al proponérselo a José Manuel y a Virginia ellos enseguida dieron su visto bueno y me animaron a completarlo con funcionalidades que no había pensado o ni conocía.

6.2- Elección de la herramienta de desarrollo

Esta también fue una decisión muy fácil, no sé si es por influencia paterna o por ideología siempre he sido partidario del software de código abierto. Así que en análisis del proyecto empecé a estudiar diferentes tipos de herramientas low code. A pesar de que la mayoría son de código cerrado finalmente encontré OpenXava.

OpenXava me ofrecía todo lo que buscaba: ser de código abierto, estar desarrollada en España, tener un soporte técnico muy completo y estar basada en el lenguaje que más hemos visto en el grado como es Java.

Estudié brevemente si era factible el uso de su framework para el desarrollo de lo que tenía en mente para mi aplicación y puede ver cómo en gran parte sí que me ofrecía la posibilidad de llevarlo a cabo.

Otra de las cualidades por las cuales me decanté por OpenXava es que la mayoría de las herramientas low code usan la tecnología Drag&Drop, pero Openxava no, con Openxava se tiene que escribir código. Esto último me pareció más adecuado para la realización de un Trabajo de Final de Grado en informática.

6.3- Formación

En este apartado tuve la gran suerte de que OpenXava ofrece un curso de formación de 9 sesiones que es muy completo y funcional.

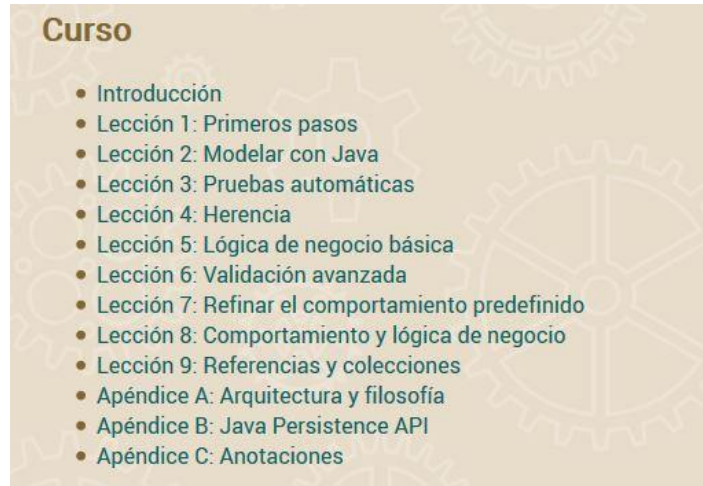


Ilustración 4 Curso OpenXava

En cada una de las sesiones nos enseñan diferentes puntos que me han servido para la implementación final.

A parte, tiene otros tutoriales para la configuración con otras bases de datos u otros servidores entre otros tutoriales.

A la hora de ir programando ha sido de gran ayuda la API propia del lenguaje: <https://www.openxava.org/OpenXavaDoc/apidocs/> en la que nos salen todas las funcionalidades posibles del framework.

6.4- Resolución de problemas técnicos

La mayoría de los problemas técnicos se han resuelto con éxito gracias en la mayoría de los casos al foro oficial de dudas de OpenXava, el foro está alojado en la plataforma SourceForge.

<https://sourceforge.net/p/openxava/discussion/437013/>

En él, los usuarios de OpenXava cuelgan sus dudas o errores y la propia comunidad comparte sus conocimientos para ayudar. En caso de no obtener respuesta por parte de la comunidad, el creador de OpenXava (Javier Paniza), resuelve en la medida de lo posible las dudas al cabo de dos días.

7- Conclusiones y líneas de trabajo futuras

Una vez finalizado el proyecto es hora de hacer balance.

7.1- Conclusiones

Por una parte, estoy muy contento por la adquisición de conocimientos que he experimentado y porque la gran mayoría de los objetivos iniciales se han cumplido de manera satisfactoria.

Al ser una aplicación desarrollada en unos dos meses el resultado lo considero muy bueno y completo. Al principio del proyecto tenía la idea de compaginar la aplicación con la creación de una tienda online y que fuese todo junto como proyecto, pero por suerte, gracias a mis tutores fuimos precavidos y optamos por prescindir de la tienda online.

En cuanto al resultado de la aplicación tengo una sensación un poco agri dulce. La aplicación funciona muy bien y cumple con los requisitos que nos planteamos al principio del proyecto.

No obstante, la parte menos positiva ha sido tener ideas para diferentes funcionalidades y ver que no he sido capaz de plasmarlas, ya sea porque al ser una herramienta low code te limita un poco en cuanto al desarrollo o por no tener los conocimientos necesarios para la implementación.

El factor tiempo también ha jugado en mi contra, ya que los mayores avances y cuando más he aprendido ha sido en las últimas semanas que es cuando más cómodo he estado desarrollando la aplicación.

Como conclusión final, diría que la experiencia de mi TFG ha sido muy positiva ya que he aprendido a desarrollar una aplicación sin un guion dado por un profesor. Supongo que las partes negativas vienen dadas por el espíritu perfeccionista o por la ambición de querer siempre más.

7.2- Líneas de trabajo futuras

- Mejorar el punto pedido para que los pedidos se gestionen solos.
- Migrar a una base de datos más robusta como MySQL.
- Crear informes personalizados para los pedidos.
- En caso de mejorar a OpenXava Pro, la implementación de creaciones de usuarios.
- Alojar la aplicación en un servidor de verdad y no en un pc.

8- Bibliografía

1. Drag & Drop [Internet]. NeoAttack. [citado 3 de noviembre de 2020]. Disponible en: <https://neoattack.com/neowiki/drag-drop/>
2. Metodología RAD [Internet]. Metodología RAD. [citado 4 de noviembre de 2020]. Disponible en: <http://metodologiarad.weebly.com/>
3. Ventajas Low Code [Internet]. IONOS Digitalguide. [citado 3 de noviembre de 2020]. Disponible en: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-el-low-code/>
4. Manaure A the A/ A. Contraste low-code [Internet]. CIOAL The Standard IT. 2020 [citado 4 de noviembre de 2020]. Disponible en: <http://thestandardcio.com/2020/08/25/low-code-pros-y-contras/>
5. Martí T. Razones para elegir low-code [Internet]. AuraPortal. 2020 [citado 4 de noviembre de 2020]. Disponible en: <https://www.auraquantic.com/es/cinco-razones-para-elegir-low-code/>
6. Desarrollo ágil de software. En: Wikipedia, la enciclopedia libre [Internet]. 2020 [citado 4 de noviembre de 2020]. Disponible en: https://es.wikipedia.org/w/index.php?title=Desarrollo_%C3%A1gil_de_software&oldid=130060551
7. AuraPortal es ahora AuraQuantic. Plataforma de Aplicaciones Low-code e iBPMS [Internet]. AuraPortal. [citado 5 de noviembre de 2020]. Disponible en: <https://www.auraquantic.com/es/>
8. El árbol de decisión para las categorías Low-Code [Internet]. AuraPortal. [citado 5 de noviembre de 2020]. Disponible en: <https://www.auraquantic.com/es/que-es-low-code/>
9. Low-code Application Development Platform - Build Apps Fast & Efficiently | Mendix [Internet]. [citado 5 de noviembre de 2020]. Disponible en: <https://www.mendix.com/>
10. ERP y CRM de código abierto | Odoo [Internet]. Odoo S.A. [citado 5 de noviembre de 2020]. Disponible en: <https://www.odoo.com/>
11. Java (lenguaje de programación). En: Wikipedia, la enciclopedia libre [Internet]. 2021 [citado 17 de enero de 2021]. Disponible en: [https://es.wikipedia.org/w/index.php?title=Java_\(lenguaje_de_programaci%C3%B3n\)&oldid=132315990](https://es.wikipedia.org/w/index.php?title=Java_(lenguaje_de_programaci%C3%B3n)&oldid=132315990)
12. HSQldb. En: Wikipedia, la enciclopedia libre [Internet]. 2020 [citado 17 de enero de 2021]. Disponible en: <https://es.wikipedia.org/w/index.php?title=HSQLDB&oldid=126741102>
