
INSTITUTO TECNOLÓGICO SUPERIOR DEL OCCIDENTE DEL ESTADO
DE HIDALGO

DIVISIÓN DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

“Software de display para monitoreo de línea”

INFORME TÉCNICO

Presenta

Carlos Alberto Barrera Lugo

Asesores:

Asesor industrial:

Ing. Roberto Juárez Angeles

Asesor académico:

Ing. Sergio Cruz Pérez

Contenido

Resumen	5
I. Introducción	7
II. Planteamiento del problema	9
III. Objetivos	11
3.1. General	11
3.2. Específicos	11
IV. Justificación	12
V. Marco Teórico	14
V 5.1. Marco Referencia	14
V 5.2. Marco Teórico	14
V.1.1 Teoría general de sistemas (TGS)	14
V.1.2 Ingeniería de sistemas computacionales	14
V.1.3 Ingeniería de software	15
V.1.4 Metodología ágil	15
V.1.5 Sistemas basados en transacciones	16
V.1.6 N-tier	16
V.1.7 Clean code	17
V.1.8 Sistema de control de versiones	18
V.1.9 Librerías	18
VI. Desarrollo del proyecto	19
VI.1.1 Requerimientos del sistema	19
VI.1.2 Análisis y diseño de la base de datos	20
VI.1.3 Análisis y diseño general del sistema	21
VI.2.1 Programación del sistema	25
VI.2.2 SAP ERP	25
VI.2.3 Escalabilidad	26
VI.2.4 Refactorización	26
VI.2.5 Procedimiento almacenado	27
VI.2.7 CRUD	28
VI.2.8 Interfaz de usuario	28
VI.2.9 Programación orientada a objetos	28
VI.2.10 IDE	29
VI.2.11 Visual Basic	29
VI.2.12 C#	30
VI.2.13 SQL Server	30
VI.2.14 ADO.NET	31
VI.2.15 DLL	31

VI.2.16 XML	32
VI.2.17 LINQ	32
VII. Resultados	35
VII.1 Objetivo específico 1.	35
VII.2 Objetivo específico 2.	36
VII.3 Objetivo específico 3.	37
VII.4 Objetivo específico 4.	37
VII.5 Objetivo específico 5.	38
VII.6 Objetivo específico 6.	39
VII.7 Objetivo específico 7.	39
VII.8 Objetivo específico 8.	40
VII.9 Objetivo específico 9.	41
VII.10 Objetivo específico 10.	41
VIII. Conclusiones	42
IX. Recomendaciones	43
X. Competencias desarrolladas	43
Bibliografía	45
XI. Anexos	49
XI.1.1 Cronograma de actividades	49
XI.1.2 Manual de usuario	50

Índice de Ilustraciones

<i>Ilustración 1 Paleta de colores que se implementó para el software</i>	23
<i>Ilustración 2 Tipografía usadas para el software</i>	23
<i>Ilustración 3 Proceso de ejecución con ADO.NET</i>	31
<i>Ilustración 4 Ejemplo de documentación XML</i>	32
<i>Ilustración 5 Linq para objetos</i>	33
<i>Ilustración 6 Linq a XML</i>	34
<i>Ilustración 7 Comparativa entre patrones de diseño</i>	35
<i>Ilustración 8 Programa original</i>	36
<i>Ilustración 9 Programa refactorizado</i>	36
<i>Ilustración 10 Marco de trabajo general</i>	37
<i>Ilustración 11 Aspecto gráfico original</i>	38
<i>Ilustración 12 Aspecto gráfico actualizado</i>	38
<i>Ilustración 13 Segmentación de los datos</i>	38
<i>Ilustración 14 Aportaciones al DLL SPEnginerCSharp</i>	39
<i>Ilustración 15 Ejemplo de código LINQ to XML</i>	40
<i>Ilustración 16 Contribuciones a los repositorios del proyecto</i>	40
<i>Ilustración 17 Operaciones CRUD</i>	41
<i>Ilustración 18 Cronograma de actividades</i>	49
<i>Ilustración 19 Inicio de la aplicación</i>	50
<i>Ilustración 20 Alertas y cargas</i>	51
<i>Ilustración 21 Menú principal</i>	51
<i>Ilustración 22 Botón flotante del menú principal</i>	52
<i>Ilustración 23 Ventana para editar datos</i>	52
<i>Ilustración 24 Datos actualizados con ayuda de las pestañas superiores</i>	53
<i>Ilustración 25 Campos estáticos</i>	54
<i>Ilustración 26 Campos dinámicos y controles para operaciones de alta, baja y modificaciones</i>	55
<i>Ilustración 27 Pestaña materiales con datos precargados</i>	55
<i>Ilustración 28 Pestaña materiales sin datos</i>	56
<i>Ilustración 29 Edición de datos en materiales</i>	56
<i>Ilustración 30 Tabla para registrar las actividades</i>	57
<i>Ilustración 31 Tablas de meta semanal</i>	57
<i>Ilustración 32 Tabla de documentos relacionados</i>	57
<i>Ilustración 33 Botones inferiores de la ventana editar datos</i>	58
<i>Ilustración 34 Menú flotante</i>	58

<i>Ilustración 35 Pantalla dinámica</i>	<i>58</i>
<i>Ilustración 36 Menú lateral izquierdo.....</i>	<i>59</i>
<i>Ilustración 37 Sección estática.....</i>	<i>59</i>
<i>Ilustración 38 Parte dinamica.....</i>	<i>60</i>
<i>Ilustración 39 Parte dinámica 2.....</i>	<i>60</i>
<i>Ilustración 40 Controles de tiempo.....</i>	<i>61</i>

Índice de Tablas

<i>Tabla 1. Privilegios por tipo de usuario</i>	¡Error! Marcador no definido.
<i>Tabla 2. Resoluciones de pantalla que soporta el software</i>	¡Error! Marcador no definido.

Resumen

Hasta este momento la empresa cuenta con un software que permite visualizar los procesos en línea de producción en tres pantallas distintas, este programa fue desarrollado en Visual Basic con el funcionamiento adecuado, sin embargo durante el inicio del año 2020 el gerente se dio cuenta de la falta de actualización de este y otros programas dentro de las áreas de la empresa por lo que se encuentra en un proceso de actualización e integración de un sistema SAP para controlar eficientemente sus datos.

Una necesidad que fue muy evidente y quizás la más importante fue que el software ya existente no es muy eficiente ya que implica una serie de malas prácticas de programación y una mala experiencia de usuario, tales como conflictos en el nombrado de los componentes, no es amigable para el usuario, no cuenta con estándares de programación, etc., a partir de este conjunto de malas prácticas de programación, procede el desarrollo del software.

El objetivo del proyecto fue el desarrollo de un software que sea eficiente, amigable, intuitivo y que posea un mejor manejo de datos, es decir que cumpla con experiencia de usuario (UI/UX) y una eficiente gestión de los datos de cada línea, que se adapte perfectamente a las necesidades que se puedan llegar a presentar, el software tiene el objetivo de gestionar los procesos de producción, administración de forma sencilla, con un diseño capaz de adaptarse a cualquier resolución de pantalla y que reduzca costos de mantenimiento e implementación.

Como parte del inicio del proyecto fue necesario realizar investigaciones preliminares de los términos y herramientas que se utilizarían para el desarrollo tales como: lenguaje de programación C#, modelo de n-capas (presentación, negocio, entidades y datos), Manejo de archivos XML, Material Design, ADO.NET (ActiveX Data Objects), se realizaron pruebas para mostrar una mejor calidad y fiabilidad ante diferentes situaciones, esto con el fin de desarrollar un software estable y robusto.

Una vez hecha la selección de herramientas para el desarrollo, se eligió la forma de trabajo, esta para reducir tiempos y optimizar la forma de trabajar con el fin de mostrar un mejor rendimiento en el proyecto, se optó por usar un sistema de controlador de versiones Git (para gestionar archivos locales) y GitHub (como el

almacenamiento remoto o repositorio), para mayor seguridad de datos bajo una llave de SSH y para la asignación de actividades durante el tiempo de residencia se implementó un cronograma de actividades.

Teniendo las bases para el desarrollo se comenzó a hacer el respectivo levantamiento de requerimientos, esta fue la parte más compleja del proyecto ya que como se trata de un software existente que no cumple con documentación de código ni documentación funcional, además de no seguir una buena estructura de programación y sin lineamientos para lo que debe contener cada una de las secciones. Para saber qué contenido deberían tener las demás secciones se investigó con el personal de producción y con el gerente para ver las necesidades de estos. Teniendo claro los requerimientos y las herramientas necesarias se desarrolló y se implementó exitosamente en la línea de producción de ACC MEX S.A. De C.V ubicada en Libramiento Sur-Poniente #808 Campestre Italiana, Santiago de Querétaro, Qro

El desarrollo de este proyecto fue posible gracias a que se siguieron principios para el desarrollo de sistemas basados en la ingeniería de software, implementación de una metodología ágil, conocimiento y desarrollo de un sistema basado en transacciones, buenas prácticas de programación, manejo de control de versiones y un buen manejo del tiempo para administrar las pruebas del software.

Palabras clave: ADO.NET, C#, LINQ to XML, SQL SERVER, Git.

I. Introducción

Debido a que la tecnología relacionada al desarrollo de software se ve cada vez más envuelta en las empresas, se ha notado una creciente necesidad por automatizar procesos relacionados al área industrial en donde la gestión de recursos es sumamente importante. El tiempo es un recurso que las empresas valoran por encima de cualquier otro, inclusive en algunos casos se puede valorar más que el dinero.

Gracias a la alta demanda en la automatización de procesos algunas empresas han optado por desarrollar subsistemas para llevar el control de algunos recursos, el problema con este tipo de soluciones surge cuando se piensa a futuro, pues generalmente el desarrollo de una solución barata y rápida se traduce en falta de escalabilidad y un aumento considerable de la deuda técnica, gracias a esto surgió la alternativa de crear un sistema monolítico en donde se puedan obtener recursos que a su vez serán consumidos por otras aplicaciones y así aumentar el rendimiento.

Si bien es cierto que actualmente este tipo de arquitecturas pueden carecer de escalabilidad cuando se pretende migrar a la nube, no deja de ser necesario comprender modelos básicos de arquitecturas y tratar de adaptarlas al crecimiento en la nube parece una vertiente que gana un mayor auge de practicantes, pues la migración a la nube puede ser un camino fácil o difícil dependiendo de la calidad con la que se escribió el código.

Dentro de este trabajo se podrán encontrar los detalles para elaborar un software de escritorio para la línea de producción en ACC MEX S.A de C.V. la cual pasará por un proceso de análisis detallado mientras se trate de definir el planteamiento que tendrá el problema por resolver, seguido de una serie de especificaciones generales que dan pauta a los objetivos de realizar un proyecto, para reafirmar la importancia del mismo se podrá consultar el apartado de Justificación.

Un punto muy importante para comprender los problemas de desarrollo es seguir un camino de aprendizaje, en este caso particular se mencionan las teorías en las que se basa el proyecto y que pueden presentarse como casos de oportunidad para seguir desarrollándose los temas que se mencionan en el marco de referencia van

relacionados a la teoría general de sistemas, ingeniería del software, metodologías ágiles, sistemas basados en transacciones, aplicación de una arquitectura de N capas (N-tier), aplicación de principios de código limpio y conocimiento sobre los sistemas de control de versiones. Además, se hace una mención especial de todas las tecnologías que sirven de apoyo en el desarrollo de este aplicativo entre las que destacan: ADO.NET y el uso de LINQ to XML

También se describe el área de la empresa donde se aplicó el proyecto, se describe su funcionamiento para comprender la problemática presentada; se presenta el cronograma de actividades explicando a detalle cada una de ellas, el tiempo que se requiere invertir para solucionar el problema y los medios que se utilizaron para que se cumplan.

Después se exponen los fundamentos teóricos donde se mencionan las bases que ayudaron a tomar una decisión para ofrecer una solución adecuada. Además, en este apartado se encuentra la terminología y definiciones de las distintas herramientas, estándares y tecnologías que fueron utilizadas (C#, ADO.NET, GIT, SQL SERVER, entre otros).

A lo largo del desarrollo del sistema se establece un plan de actividades a seguir describiendo la forma en la que cada una de las fases se van realizando en el tiempo que fueron planeados, todo esto basados en una metodología Ágil.

II. Planteamiento del problema

ACC MEX S. A. DE C. V. Es una empresa establecida en Libramiento Sur-Poniente #808 Campestre Italiana, Santiago de Querétaro, Qro. Se llevan a cabo distintos tipos de proyectos relacionados a la ingeniería en automatización y sistemas de control industrial, llevando a cabo proyectos en forma de soluciones en robótica, soluciones para visión computacional, desarrollo de software y/o mantenimiento. La empresa cuenta con varias áreas para poder llevar a cabo el proyecto y existe una en específico con el mismo nombre encargada de gestionar de la mejor forma los recursos de los cuales se disponen.

Uno de los recursos clave para la empresa es el tiempo, pues un proyecto suele demorar mayor o menor tiempo si los procesos se gestionan adecuadamente, dicha gestión es un factor que se suele manejar por medio de programas propios o de terceros dentro de la empresa. El área de producción es uno de los sectores donde menos visibilidad del tiempo invertido por proyecto se tiene.

Se plantearon diversas necesidades, de las cuales se tomó como primordial: implementar mejoras en el software que permita que los trabajadores puedan visualizar los procesos en línea de producción en tres pantallas distintas y monitorear la línea de producción. Aunado a esto se requiere de un sistema de escritorio en el cual personal autorizado pueda gestionar los procesos de 3 diferentes líneas de producción, además de enviar alertas en las pantallas para identificar si el proceso se encuentra en: “proceso, detenido, alto”.

Hasta este momento la empresa cuenta con un software que permite visualizar los procesos en línea de producción en tres pantallas distintas; para el contexto temporal en que se desarrolló pudo resolver adecuadamente sus necesidades pues principalmente cuando el personal encargado se dirigía a revisar el estado de sus proyectos podían notar un avance sin dejar la incertidumbre de por medio.

Este programa fue desarrollado en Visual Basic con el funcionamiento adecuado, sin embargo, durante el inicio del año 2020 el gerente se dio cuenta de la falta de actualización y funcionalidad de este y otros programas dentro de las áreas de la empresa por lo que se encuentra en un proceso de actualización e integración de un sistema SAP.

Dentro de los problemas que se han encontrado está la necesidad de integrar sus aplicaciones al SAP, cambiar la estructura interna a un patrón de diseño escalable, delimitar los datos que entran o salen del origen de datos y llevar a cabo la implementación de las aplicaciones en producción.

III. Objetivos

3.1. General

Desarrollar e implementar una aplicación de escritorio para la línea de producción de ACC MEX S.A. de C.V, que permita visualizar los procesos en línea de producción en tres pantallas distintas y monitorear la línea de producción.

3.2. Específicos

- Reestructurar el patrón de diseño de la aplicación para aumentar su escalabilidad utilizando un modelo de n-capas (presentación, negocio, entidades y datos).
- Analizar los requerimientos respecto al sistema actual, conservando el objetivo principal de la aplicación (informar sobre el estado en producción).
- Conocer el entorno de desarrollo en C# para realizar una refactorización de código desde VB.
- Implementar un diseño orientado a UI/UX para facilitar la comprensión de la información
- Segmentar correctamente el origen de datos local con archivos de configuración y de base de datos utilizando ADO.NET para evitar redundancia.
- Crear e implementar una DLL para manipular procedimientos almacenados de una manera más amena.
- Implementar librerías específicas de C# para manipular componentes y colecciones de datos de XML.
- Manejar un control de versiones con Git para manipular los cambios y el trabajo remoto.
- Implementar operaciones CRUD a los módulos correspondientes tanto para componentes como para XML.
- Documentar constantemente los cambios que se realicen para conservar un código más limpio.

IV. Justificación

Previo a la actualización del software, el procedimiento para saber en qué estatus se encontraban los proyectos, por parte de trabajadores, era a partir de tres pantallas dispuestas de tal forma que en cada línea de producción se observa con claridad qué procesos se estaban trabajando, sin embargo, debido a la falta de diseño de interfaces era muy fácil perder la secuencia del proceso y optar por no consultar la información en la pantalla a menos de que se suscitara un paro en la producción.

Por otra parte, los clientes que iban a revisar el estado de los proyectos, e inversionistas que planeaban desarrollar sus ideas con la empresa se encontraban descontentos por que la información era difícil de procesar, pues las Interfaces no estaban dispuestas de la mejor manera y no existía un gráfico que mostrará de forma rápida que estaba sucediendo.

Este grado de incertidumbre se verá reducido de forma gradual una vez que se implementen las mejoras correspondientes, de las cuales se destacan:

- Una interfaz de usuario visualmente atractiva, intuitiva y llamativa, gracias a la selección de colores y componentes que permiten una mejor interpretación aún para personas poco experimentadas con los proyectos que se desarrollan.
- Una mayor fluidez al momento de navegar por las ventanas cuando que se desee editar la información, lo que significa un ahorro de tiempo en las interfaces de carga y un mejor control para realizar dichas ediciones
- Segmentación de la información gracias al método de guardado local por medio de un archivo XML evitando contar con datos redundantes en base de datos y a su vez liberar la carga de consultas al servidor, lo que significa un mejor rendimiento del SAP.

Todas estas mejoras se notarán gradualmente en la medida en que se satisfagan las pruebas propuestas por el equipo de desarrollo, el CTO y por supuesto los asesores a cargo. Con esta actualización además es posible que la empresa pueda calificar con los certificados que la avalen ante autoridades importantes en el rubro tecnológico.

Pensando acerca del futuro de esta y otro tipo de aplicaciones, se tiene en consideración la escalabilidad, pues en el código fuente se han realizado acciones adecuadas como:

- Programación orientada a objetos
- Selección de un patrón de diseño escalable y no monolítico.
- Uso de buenas prácticas de programación
- Reutilización de código en la medida de lo posible
- Documentación constante de cada elemento del código.

V. Marco Teórico

V 5.1. Marco Referencia

ACC MEX S.A. De C.V ubicada en Libramiento Sur-Poniente #808 Campestre Italiana, Santiago de Querétaro, Qro, creada para satisfacer las necesidades de control y distribución eléctrica de la industria nacional e internacional, fundada en marzo del año 2000

V 5.2. Marco Teórico

V.1.1 Teoría general de sistemas (TGS)

La observación de un concepto fundamental en un área determinada varía de acuerdo a los objetivos de la misma, es decir que cada objeto de estudio derivará de un elemento en particular, por ejemplo, una célula es la base para la biología, un dato lo es para la informática e incluso el pensamiento es la base para estudiar la lógica y el razonamiento. Para representar algunos tópicos complejos de áreas especializadas de la ciencia se volvió indispensable el uso de un modelo de construcción teórica que pueda establecer un nivel intermedio entre un contenido específico (que aporte poco o nulo significado general) y la parte más general (que pueda carecer de contenido) (Johansen Bertoglio, 1993). La presente investigación se sustenta en la TGS debido a que el aplicativo construido formará parte de un sistema de información, formalmente la TGS (Teoría General de Sistemas) pretende ser una forma sistemática y científica de aproximación y representación de la realidad, al mismo tiempo, como una orientación hacia una práctica estimulante para formas de trabajo transdisciplinarias (Arnold Cathalifaud & Osorio, 1998).

V.1.2 Ingeniería de sistemas computacionales

La TGS funge como un esqueleto para el estudio de las ramas de la ciencia y dentro de estas se encuentran diversas tendencias que apuntan a un contexto aplicativo combinando el pensamiento ingenieril se dio origen a dicha rama interdisciplinar que hace referencia a la planeación, diseño, evaluación y construcción científica de sistemas hombre – máquina (Johansen Bertoglio, 1993). Debido al cambio generacional y a la aparición de los primeros ordenadores surgió una vertiente en la rama de la ingeniería de sistemas que involucra precisamente las ciencias de la

computación adquiriendo los principios de creación hacia elementos no tangibles pero claramente visibles, este apartado teórico es la base fundamental para la creación ya sea de un sistema o parte del mismo basado en los elementos de las ciencias de la computación como lo son la programación, paradigmas, estructuras de datos, arquitectura, entre otros elementos detallados más adelante.

V.1.3 Ingeniería de software

Esta rama de las ciencias de la computación aporta un enfoque orientado a la construcción de aplicaciones, es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales (Pressman, 2010). Otra definición que abarca más aspectos es aquella desarrollada por la IEEE (The Institute of Electrical and Electronics Engineers, 1990) en la que se dice la ingeniería de software es la aplicación de un enfoque sistémico, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir la aplicación de la ingeniería al software (The Institute of Electrical and Electronics Engineers, 1990), se puede deducir que la repercusión de este campo de la ingeniería impacta en la forma de gestionar los recursos para llevar a cabo la concepción del software.

V.1.4 Metodología ágil

La metodología de trabajo que se emplea en el desarrollo de software robusto está orientado a modelos estructurados y bien documentados, que en un principio funcionan para equipos expertos y muy bien coordinados, pero cuando se trata de ciclos de desarrollo cortos estas estructuras clásicas de desarrollo presentan ciertos problemas, las metodologías ágiles provienen del manifiesto ágil y pretende optimizar el proceso de creación de software, se valora entonces:

- Al individuo y sus interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- Desarrollar software que funcione más que la documentación del mismo.
- La colaboración con el cliente más que la negociación de su contrato.
- Responde a los cambios más que seguir con el plan establecido.

Además, se siguen doce principios para seguir realmente una metodología ágil (López Menéndez de Jiménez, 2015). Dentro de las metodologías ágiles se encuentran Kanban y LEAN como metodologías menos específicas y por tanto más fáciles de adaptar a un equipo de trabajo con relativa facilidad, dentro de este proyecto se pretende adoptar principios de ambas metodologías, especialmente de Kanban, pero sin seguir pautas rígidas (Project Management Institute, 2017).

V.1.5 Sistemas basados en transacciones

Son sistemas generalmente localizados en un ordenador específico el cual permite registrar y llevar a cabo transacciones (consultas, altas y bajas) que son necesarias para el desarrollo de una empresa, el objetivo principal de este tipo de sistemas es automatizar el procesamiento de los volúmenes de información mismos que se almacenan en una base de datos o archivos específicos de colecciones (Pítsica Marqués, 2001). El proyecto que se pretende llevar a cabo está basado en este tipo de sistema de información debido a que ayuda a automatizar un proceso de visualización de datos con ayuda de un servidor para realizar operaciones de tipo CRUD en un archivo específico evitando la redundancia del guardado de información.

V.1.6 N-tier

El desarrollo de un producto de software tiene una implicación muy importante en la estructura o patrón de diseño para implementar una solución, según Refactoring Guru (2014-2020) estos patrones de diseño son considerados como planos prefabricados que se pueden personalizar para resolver un problema un problema de diseño del código de una aplicación, los patrones de diseño cuentan con una serie de secciones específicas como lo son:

- Un propósito
- La motivación
- Estructura
- Ejemplo de código

Partiendo de esta premisa Mark Richards (Richards, 2015) explica que los patrones de diseño de arquitecturas albergan una serie de elementos que las convierten en

más o menos escalables, más o menos difíciles de implementar y más o menos difíciles de comprender, la arquitectura de N capas o N-tier tiene la característica de ser un buen elemento de partida a la hora de diseñar software de poco impacto, el autor incluso expresa que es un buen punto de partida para cualquiera que incursione llevando el desarrollo de aplicaciones a un nivel más organizado.

El patrón de N capas no define un límite en específico para organizar la estructura del proyecto, sin embargo, es importante mencionar que es idóneo usar 4 capas que pueden tener nombres específicos o se puede optar por nombrarlos como:

- Presentación (presentation).
- Negocios (bussiness).
- Entidades (Entities).
- Base de datos (database)

En el presente proyecto se optó por nombrarlas de la misma manera a excepción de la capa de base de datos a la que se le recortó el nombre a “Datos”, además se siguió el principio de nivel de aislamiento que determina que algunas capas pueden estar abiertas para saltar un nivel de capa e ir a otro, este diseño de capas se caracteriza por ser bueno en desarrollo y pruebas.

V.1.7 Clean code

Es una serie de pensamientos que tratan de dar un estándar adecuado para generar una alta mantenibilidad del código que se desarrolle, esta manera de desarrollar código permite obtener beneficios como la facilidad de desarrollar pruebas que sean del tipo que sean, si se adoptan este tipo de medidas además será mucho más sencillo que cualquier otra persona pueda comprender qué está pasando en el software (Suryanarayana, Samarthayam, & Sharma, 2015). Las prácticas de clean code no solo abordan cómo evitar malas prácticas si no que a su vez se abordan temas de deuda técnica o código hediondo que pudieran comprometer el desarrollo futuro del código, dentro de la aplicación que se trabajó se detectó una deuda técnica orientada a: el diseño, pues carecía de un estándar y todo parecía ejecutarse en una misma clase que combinaba la parte de la interfaz de usuario, el código, debido a que se encontraban múltiples carencias sobre el conocimiento del lenguaje, estándares para nombrar elementos del código y

carencia de estructuras de datos para manipular la información, en cuanto a la documentación se encontró un gran déficit, pues no existía una manera de conocer cómo funcionaba cada parte importante del código ya que se incluían únicamente comentarios relacionados a partes fáciles de comprender e incluso un tanto obvias.

V.1.8 Sistema de control de versiones

Una parte crucial para llevar un mejor control sobre el desarrollo del proyecto es cuando se implementa un sistema de control de versiones, pues este tipo de sistema puede registrar los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo (Git, 2014) si se realiza un análisis más profundo sobre cómo funciona el control de versiones, se puede notar que existen diversos tipos de sistemas de entre los cuales destaca uno, principalmente por su alta disponibilidad y capacidad de colaboración, este es denominado como *sistema de control de versiones distribuido* en el cual se basa Git el cual destaca debido a que cuenta con una alta demanda por la comunidad de desarrolladores alrededor del mundo debido a que no controla cada cambio específico sino que más bien hace una especie de captura de todos los archivos cada vez que se guardan se optó por utilizar esta tecnología para llevar el control del proyecto debido a que la colaboración aunada con GitHub puede ser altamente eficiente para trabajar en equipo.

V.1.9 Librerías

- **Guna. UI2. WinForms 2.0.1.2**

Guna UI es la suite para crear una interfaz de usuario de aplicación de escritorio innovadora. Es para desarrolladores que se dirigen a la plataforma .NET Windows Forms. Guna UI garantiza un desarrollo más rápido y una productividad mejorada. Permite a los desarrolladores manipular y crear un excelente diseño gráfico.

VI. Desarrollo del proyecto

Se realizó una reunión y se organizó el cronograma de actividades para llevar a cabo el desarrollo del proyecto, para la estimación de las semanas de trabajo por mes y por etapa, en el [ANEXO I. Cronograma](#) se muestra detalle cómo se tiene organizado el tiempo para desarrollar los productos.

Durante esta reunión se mencionó sobre el plan de desarrollo, los entregables y el objetivo del proyecto.

Además, se mencionaron los requerimientos para el desarrollo del proyecto como lo son: el diseño, la funcionalidad y los posibles cambios del sistema, para el desarrollo de este, los integrantes del proyecto aportamos ideas para el prototipo.

VI.1.1 Requerimientos del sistema

Para comenzar, es necesario conocer cuáles son las necesidades de los posibles usuarios a quienes va dirigido el software, para lo cual se realizaron una serie de investigaciones que tenían como objetivo conseguir toda la información relevante respecto a la línea de producción. Posteriormente y tomando en cuenta el funcionamiento del primer prototipo existente, el levantamiento de requerimientos señaló que el software tendría que tener la facilidad y funciones del poder ingresar, modificar, eliminar y consultar información del contenido de los procesos a manejar en cada línea de producción y que cada área específica contará con un nivel de privilegios inferior a un usuario de tipo personal autorizado, el cual tendrá acceso a todo el sistema; que en este caso sería el personal autorizado o un Administrador general, el cual recibirá toda la capacitación para este a su vez, capacitar al personal de de trabajo de cada línea de producción, esto con el fin de tener un control sobre las personas que tendrán acceso a la información y publicación de contenido de cada proceso.

Es por ello que una pregunta importante fue quien tendrá el acceso al sistema y qué acciones tendrán permitido realizar. Los permisos quedaron de la siguiente manera:

Tipo de usuario	Permisos
-----------------	----------

Personal autorizado	<ul style="list-style-type: none"> • Total acceso al Sistema Administrativo del contenido y configuración.
Trabajador de línea de producción	<ul style="list-style-type: none"> • Visualizar la información presentada mediante una pantalla correspondiente a su línea. • Acceso al control del cronómetro dinámico para registrar tiempo total de trabajo, tiempo de descanso, mediante comunicación serial. • Acceso a enviar capturas de pantalla, mediante correo Gmail, accionadas mediante comunicación serial.

Tabla 1. Privilegios por tipo de usuario.

Contar con un Personal autorizado de contenido dinámico: Este usuario permitirá publicar procesos, en el sistema, esta información será almacenada en una base de datos.

En ocasiones con este tipo de usuario los contenidos del sistema podrán ser editados, editar el diseño del sistema activando o desactivando el modo oscuro, administrar los datos de los procesos, menús, cambiar el estado en el que se encuentra el proceso, seleccionar los documentos relacionados, lista de actividades generales, materiales que utilizará y configurarlos, capturar pantalla y decidir si guardar la imagen en una unidad de almacenamiento de la computadora o bien ser enviado a un correo electrónico Gmail.

Los cambios realizados por medio del personal autorizado son reflejados inmediatamente en el sistema. Esto aumenta considerablemente el dinamismo de esta actividad.

Contar con un trabajador de línea: Este usuario permitirá solo visualizar la información mostrada en la pantalla y accionar un control de hardware que será conectado medio conexión serial para interactuar con los cronómetros.

VI.1.2 Análisis y diseño de la base de datos

Una parte muy importante para el desarrollo de este sistema fue conocer la base de datos empleada por el usuario de personal autorizado para poder eficientar los procesos y el uso que se le daría en la programación de las consultas, al tener un

prototipo previo del sistema conlleva a tener ya definida una base de datos existente, por lo que nos dimos a la tarea de analizarla para así poder usarla en el proyecto.

VI.1.3 Análisis y diseño general del sistema

Una vez realizados los requerimientos y la investigación que se enfocó en las necesidades generales de los procesos de la línea de producción, tanto en monitoreo para el usuario de personal autorizado y la visualización de la información para el usuario de trabajador de línea de producción.

- **Modularidad**

Se diseñó una estructura con múltiples módulos posibles para el software, claro hay módulos que son obligatorios tales como el módulo de Menú, Editar datos, Ventana dinámica, Comunicación serial, Material Design, enviar captura de pantalla por correo, cronometraje de tiempos de trabajo real por parte del trabajador de línea de producción y otro no obligatorios como el módulo de modo oscuro. Estas decisiones se toman en el levantamiento de requerimientos

- **Módulos del sistema**

- a) Menú principal: El menú principal cuenta con tres botones grandes de color azul en donde se llevan a cabo las principales operaciones del programa, además de existir un cuarto botón flotante para activar algunas funciones tales como (Salir, Modo oscuro, Ayuda, Captura de pantalla).
- b) Editar Datos: El apartado para editar datos cuenta con una serie de pestañas que realizan funcionalidades específicas para cada caso. En este módulo se define la configuración y el monitoreo de los datos a manejar, el cual solo tiene acceso el usuario de personal autorizado.
- c) Ventana Dinámica: El apartado de Ventana Dinámica tiene la funcionalidad de mostrar información, para ello cuenta con un conjunto de tablas, barras progresivas y un cronómetro que controla el tiempo de trabajo real y tiempo total de descansos durante la jornada laboral, este módulo tiene acceso ambos usuarios.

- d) Comunicación Serial: Permite conectar un módulo de arduino a la computadora para que así el trabajador pueda tener acceso al cronómetro y al capturar pantallas.
- e) Material Design: El software cuenta con un diseño moderno, limpio y agradable para la visualización, además que se implementaron efectos de animación modernos y la creación de alertas personalizadas.
- f) Captura de pantalla: Permite a ambos usuarios realizar capturas de pantalla para posteriormente ser enviadas automáticamente a un correo electrónico predeterminado.
- g) Cronómetro: El usuario de trabajador de línea tiene acceso total a un cronómetro el cual almacena el tiempo total de trabajo, tiempo total de descanso, pausar el cronómetro en caso de que pare la producción cuenta con dos paneles de historial registrando cada movimiento en tiempo real.
- h) Modo Oscuro: Permite la configuración de cambio de colores de los componentes en modo claro y modo oscuro, para evitar el cansancio de la vista.

- **Guía de estilos**

Una vez concluido el diseño general de la estructura del sistema se prosiguió al diseño de la guía de estilos, es decir un documento donde se hizo la unificación de criterios de todas las características de estilización que debe presentar el sistema. Desde el tamaño y tipo de fuentes tipográficas, colorimetría etc. Esto para lograr que en todo el software haya armonización, estandarización, uniformidad, tipificación, homogenización, unificación y normalización.



Ilustración 1. Paleta de Colores que se implementó para el software. (gráfico propio)

Tipografía

Licencia

Montserrat Thin

Libre

Montserrat Medium

Libre

Montserrat Bold

Libre

Ilustración 2. Tipografía usadas para software. (gráfico propio)

- **Diseño Adaptable**

Cabe de mencionar que existen diferentes resoluciones de pantallas, se integró la funcionalidad de responsivo de acuerdo a las resoluciones de pantalla de escritorio, soporta:

SIGLAS	NOMBRE	RESOLUCIÓN	DESCRIPCIÓN
--------	--------	------------	-------------

SD	Standard Definition	640 x 480 píxeles	Uno de los primeros estándares de baja resolución. También se conoce como VGA, pero SD es un nombre más utilizado ahora.
QHD	Quarter of High Definition	960 x 540 píxeles	Significa literalmente un cuarto de alta definición, y ha sido utilizado en dispositivos de gamas bajas
HD	High Definition	1.280 x 720 píxeles	Es el primer estándar de alta resolución, y también se conoce como 720p. No llega a ser alta definición total, pero es como un paso previo muy utilizado. Es como la primera resolución considerada HD.
FHD	Full HD o Full High Definition	1.920 x 1.080 píxeles	También conocida como 1080p, es el estándar de alta definición total.
QHD	Quad High Definition	2.560 x 1.440 píxeles	También se conoce como 1440p o 2K, y es un estándar bastante utilizado en smartphones que quieren ir un poco más allá del Full HD.

UHD	Ultra High Definition	3.840 x 2.160 píxeles	Este es el famoso 4K, que también puedes encontrar como 2160p. Es un estándar de alta resolución muy superior al Full HD, y que poco a poco se empieza a convertir en normal en televisores de gamas medias y altas.
UHD 8K	Ultra High Definition 8K	7.680 × 4.320 píxeles	El 8K o 4320p es el estándar de máxima resolución que se está empezando a ver cada vez más en televisores de gamas superiores. Aspira a ser el sucesor del 4K.

Tabla 2. Resoluciones de pantalla que soporta el software.

VI.2.1 Programación del sistema

Debido a protocolos de seguridad y bajo control de cláusulas de contrato, no es posible mostrar el código fuente del software.

Dentro de este apartado el lector podrá encontrar los conceptos puntuales que dan origen al desarrollo del sistema y que son realmente esenciales para comprender el objetivo de llevar a cabo la implementación del software, pasando de lo más genérico a conceptos muy específicos.

VI.2.2 SAP ERP

La relación del uso de sistemas informáticos para entornos empresariales ha estado presente desde el principio de la era digital ya que cuenta con un factor de innovación y captación del público impresionantes, existen diversas corrientes del pensamiento, teorías y principalmente estrategias de marketing que apuntan a la

comercialización con las empresas pudiendo traer valor agregado a elementos que ya existen (como la regulación del tiempo, gestión de recursos, control de personal, etc.) es por este motivo que algunos equipos de trabajo en el área de desarrollo han optado por desarrollar software específico para empresas. Cuando se habla de SAP (System Analysis and Program Development) se hace referencia a la compañía que, en 1972, cinco trabajadores de IBM dieron a conocer al mundo gracias a la idea de crear software con estándares empresariales que integran todos los procesos de negocio y permitieran procesar datos en tiempo real (SAP, 2020). Por su parte, la abreviatura ERP (Enterprise Resource Planning) hace alusión al software que tiene por objetivo facilitar la gestión de todos los recursos de la empresa a través de la integración de la información de los distintos departamentos y áreas funcionales (Suárez Rey, 2010).

VI.2.3 Escalabilidad

Es importante recalcar que una aplicación sea pequeña o grande en el momento que se despliega pasados algunos años pueden surgir más necesidades para resolver y es el deber del equipo de desarrollo dejar un sistema escalable. La escalabilidad es la habilidad que tiene un sistema para adaptarse al crecimiento sin perder calidad en los servicios que se ofrecen, es de suma importancia mencionar que la capacidad de crecimiento puede darse de dos maneras: scale in (escalado vertical) que hace referencia al crecimiento en un solo nodo, por otro lado, el scale out (escalado horizontal) que se enfoca en el aumento de nodos para liberar la carga del sistema (Gonzalez Godino, 2013).

VI.2.4 Refactorización

Cuando se diseña el software normalmente se debería pensar en crearlo orientado hacia la mantenibilidad del código fuente, una de las principales reglas para llevar a cabo este cometido se trata de escribir partes simples para lograr un cambio en la estructura interna para lograr que este sea más sencillo de comprender y sobre todo fácil de modificar sin alterar el comportamiento normal. Generalmente las técnicas de refactorización de código se aplican después de que el funcionamiento de un módulo ha sido desarrollado, pasa el tiempo y al revisar el código se

encuentra una difícil comprensión del funcionamiento o simplemente el producto ha crecido y necesita una mejor forma de escalar (Dooley, 2017).

VI.2.5 Procedimiento almacenado

Es un grupo de una o más instrucciones en lenguaje SQL los procedimientos se asemejan a las construcciones de otros lenguajes de programación, porque pueden: aceptar parámetros de entrada y devolver varios valores en forma de parámetros de salida al programa que realiza la llamada, contener instrucciones de programación que realicen operaciones en la base de datos, inclusive pueden contener llamadas a otros procedimientos, devolver un valor de estado a un programa que realiza una llamada para indicar si la operación se ha realizado correctamente o se han producido errores, y el motivo de estos, el uso de procedimientos almacenados acarrea una serie de beneficios entre los cuales destacan el bajo uso del tráfico de red, una mayor seguridad ante inyecciones SQL (además es posible cifrar un procedimiento almacenado), mejora el mantenimiento y la usabilidad debido a que un solo procedimiento almacenado puede ser utilizado múltiples veces (Microsoft, 2020).

VI.2.6 Colección de datos

Para que un programa cuente con una alta eficiencia debe incluirse un buen manejo de la memoria aplicando estructuras de datos o colecciones de datos que no son más que un conjunto de técnicas que aumentan considerablemente la productividad del programa, reduciendo en elevado grado, el tiempo requerido para escribir, verificar, depurar y mantener los programas. El término estructura de datos hace referencia a un conjunto de datos que, por medio de un nombre, identifican un espacio en memoria, teniendo ciertas características como la organización y estructuración, permitiendo realizar operaciones definidas en ellas. Las estructuras de datos pueden ser de dos tipos: estructuras de datos estáticas (aquellas que cuentan con un tamaño definido), estructuras de datos dinámicas (aquellas en las que el tamaño puede variar en tiempo de ejecución) (Luna López, 2012).

VI.2.7 CRUD

Es un acrónimo para las maneras en las que se puede operar sobre información almacenada. Es un mnemónico para las cuatro funciones del almacenamiento persistente. CRUD usualmente se refiere a operaciones llevadas a cabo en una base de datos o un almacén de datos, este acrónimo deriva de las palabras: Create (Crear), Read (Leer), Update (Actualizar) y Delete (Eliminar) (Mozilla and contributors, 2019).

VI.2.8 Interfaz de usuario

Para que un usuario ordinario se pueda comunicar con un ordenador o aplicación del ordenador hace falta contar con un conjunto de elementos de hardware y software que presenten la información de una forma entendible apoyándose de gráficos, colores, tipografía presentados en forma de ayuda, manuales, tutoriales, etc. Si la interfaz de usuario está perfectamente diseñada el usuario podrá navegar por la aplicación u ordenador sin problema alguno encontrando las respuestas a cada acción que realice (Sebastián Gómez, 2000).

VI.2.9 Programación orientada a objetos

Conocido como un paradigma de programación, son un conjunto de técnicas que nos permiten incrementar enormemente nuestro proceso de producción de software; aumentando drásticamente nuestra productividad por un lado y permitiéndonos abordar proyectos de mucha mayor envergadura (Morero, 2000), la POO se encarga de definir un objeto como una serie de métodos y atributos que se encontrarán definidos en una clase (abstracción de la realidad, sirve como una especie de plano) al utilizar un paradigma de programación orientado a objetos es posible observar un grado de reusabilidad en el código que van de la mano con otra serie de conceptos que son:

- **Herencia:** Destaca la cualidad de contar con clases padre e hijo para reutilizar elementos de las clases derivadas.
- **Encapsulación:** Cuando se reúnen todos los elementos pertenecientes a una misma entidad.

- **Polimorfismo:** Comportamientos similares asociados a objetos distintos que pueden compartir este funcionamiento con elementos distintivos de cada uno.

VI.2.10 IDE

Un entorno de desarrollo integrado (IDE) es un sistema de software para el diseño de aplicaciones que combina herramientas del desarrollador comunes en una sola interfaz gráfica de usuario (GUI) (Red Hat Inc., 2020). Generalmente, un IDE cuenta con las siguientes características:

- **Editor de código fuente:** editor de texto que ayuda a escribir el código de software con funciones como el resaltado de la sintaxis con indicaciones visuales, el relleno automático específico del lenguaje y la comprobación de errores a medida que se escribe el código.
- **Automatización de compilación local:** herramientas que automatizan tareas sencillas e iterativas como parte de la creación de una compilación local del software para su uso por parte del desarrollador, como la compilación del código fuente de la computadora en un código binario, el empaquetado del código binario y la ejecución de pruebas automatizadas.
- **Depurador:** programa que sirve para probar otros programas y mostrar la ubicación de un error en el código original de forma gráfica.

VI.2.11 Visual Basic

Es un lenguaje de programación orientado a objetos desarrollado por Microsoft, utilizar visual basic puede aumentar la velocidad de creación de aplicaciones con .NET, la palabra Visual hace referencia al método de creación de la GUI (método que otros lenguajes ya han optado gracias a los IDEs), por otra parte la palabra BASIC es una referencia al lenguaje del mismo nombre que significa “Código de instrucciones simbólicas multipropósito para novatos” (Microsoft, 2020). Esta forma de desarrollar aplicaciones tiene algunas limitantes como la mantenibilidad del código que aumenta en relación a la envergadura de los proyectos y temas de escalabilidad previamente mencionados, cabe mencionar que este lenguaje fue la base para crear la lógica del primer programa para la empresa.

VI.2.12 C#

Es el lenguaje de programación por excelencia desarrollado por Microsoft que corre tanto en .NET core como en .NET framework, está orientado al paradigma de programación orientado a objetos pero comparte una clara similitud con C y Java, este lenguaje ha sido utilizado para desarrollar aplicaciones web, móviles, juegos, etc. (Microsoft, 2020). Este lenguaje es ampliamente utilizado junto al IDE Visual Studio un entorno de desarrollo en donde es posible aplicar todas las técnicas de estructura de un patrón de diseño y a su vez buenas prácticas de programación, el IDE cuenta también con herramientas de depuración y testeo lo que lleva a pensar en un desarrollo bastante completo.

VI.2.13 SQL Server

A medida que las aplicaciones informáticas comenzaron a popularizarse, se observó la vertiente de contar con datos redundantes para un conjunto de aplicaciones específico, por ejemplo un nombre de producto podría estar en el inventario y al mismo tiempo en el detalle de venta desde ese momento fue necesario comenzar a utilizar conjuntos de ficheros interrelacionados que implementan estructuras complejas que a su vez, podían ser compartidos por varios procesos de manera simultánea a este descubrimiento se le conoce como bases de datos, de una manera más formal se puede decir que una base de datos es la representación integrada de los conjuntos de entidades instancia correspondientes a las diferentes entidades tipo del sistema de información y de sus interrelaciones. Esta representación informática (o conjunto estructurado de datos) debe poder ser utilizada de forma compartida por muchos usuarios de distintos tipos (Camps Paré, y otros, 2005). Para llevar a cabo el manejo de una base de datos más tarde surgió el concepto de DBMS (Data Base Management System) que básicamente es un gestor de base de datos percibido por el usuario como tablas que contienen dos dimensiones en donde se representa la información concepto al que posteriormente se le agrego una parte más complejo pero altamente eficiente para trabajar con un gran conjunto de datos al que se le aplican conceptos más avanzados sobre atomicidad, consistencia, aislamiento y durabilidad (Gómez Fuentes, 2013). SQL Server es un RDBMS (Relational Data Base Management System) desarrollado por Microsoft que a partir de 2016 ha dado soporte no solo para Windows sino que

también para Linux y contenedores lo que permite que pueda escalar de una forma natural, la propia empresa optó por utilizar este RDBMS debido a sus buena integración con C# y su facilidad para crear conexiones hacia el servidor (Gorman, y otros, 2019).

VI.2.14 ADO.NET

Para llevar a cabo la conexión entre una aplicación y una base de datos se han implementado diversos métodos de conexión de acuerdo a las necesidades del desarrollador, para seguir el estándar ya establecido en la empresa se optó por utilizar conexión de tipo ADO.NET, es importante mencionar que este tipo de conexión es la forma más simple de conectarse a una base de datos, una conexión simple puede utilizar una sola referencia a System.Data.SqlClient y satisfacer todas sus necesidades (Stack Overflow Comunidad, 2017), el modo de acceso para una aplicación se puede apreciar en la figura posterior (ver ilustración 3).

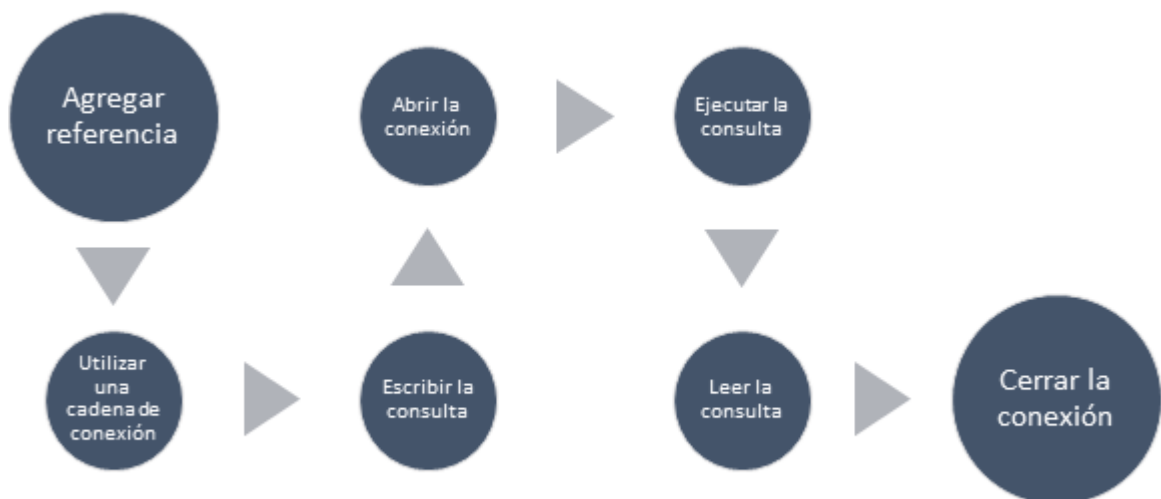


Ilustración 3. Proceso de ejecución con ADO.NET (gráfico propio).

VI.2.15 DLL

Es una librería que contiene datos que pueden ser usados por más de un programa al mismo tiempo, las siglas significan biblioteca de enlace dinámico (Dynamic Link Library) esta forma de crear pequeños fragmentos de código que puede ser

reutilizable aumenta en gran medida la eficiencia de la memoria debido a que el DLL carga únicamente cuando se necesita la funcionalidad, otro de los beneficios está en que se promueve una arquitectura modular que permite que los cambios se hagan con mayor facilidad, los desarrolladores que implementan nuevas funcionalidades además tendrán un mejor control sobre qué implementar y la característica principal por la cual se destaca es su facilidad de implementación ya que basta con tener la referencia para ocupar la funcionalidad del DLL (Microsoft, 2020).

VI.2.16 XML

El Lenguaje de Marcado Extendido es un tipo de especial de lenguaje utilizado principalmente para guardar información en un modo más sencillo (Dykes & Tittel, 2005), se compone principalmente de etiquetas propias con nombres en de cadena o tokens especiales, cada tag o etiqueta debe contener un cierre y puede contener cero o más atributos que identifiquen a la etiqueta con un valor único. Otra característica importante del lenguaje de marcado es que puede albergar comentarios para facilitar su comprensión y una característica notoria es que una etiqueta puede tener el mismo nombre y dentro contener elementos similares o distintos, es el mismo caso para el contenido de cada etiqueta (Landa Cosio N. A., 2018) (ver ilustración 4).

```
<book>
  <title>Night Fall</title>
  <author>Demille, Nelson</author>
  <publisher>Warner</publisher>
  <price>$26.95</price>
  <contentType>Fiction</contentType>
  <format>Hardback</format>
  <isbn>0446576638</isbn>
</book>
```

Ilustración 4. Ejemplo de documento XML (obtenido de Dykes, L., & Tittel, E. (2005). Ejemplo de clasificación con XML. Recuperado de XML for Dummies. Hoboken: Wiley Publishing).

VI.2.17 LINQ

Para comprender de qué se trata este concepto es necesario explicar en primer lugar lo que es el mapeo Objeto/Relacional (ORM), se refiere a la persistencia

automatizada y transparente de las tablas en una base de datos relacional, usando metadatos que definen el mapeo entre los objetos y la base de datos, dichos objetos pueden estar orientados únicamente a una estructura de datos (arreglo, matriz, etc.), un origen de datos de BD o inclusive utilizarse con XML, este último apartado requiere de algunos elementos especiales para funcionar listados a continuación:

- **xObject:** Clase abstracta que es la base para todo el contenido XML.
- **XNode:** Clase base para la mayoría del contenido XML, excepto atributos.
- **XContainer:** Define miembros para trabajar con sus hijos y es la clase padre de XElement y XDocument
- **XElement:** Define a un elemento XML.
- **XDocument:** Representa la raíz de un árbol XML, en realidad envuelve a un XElement que actúa como raíz y lo podemos usar para adicionar la declaración e instrucciones de procesamiento.

Es importante mencionar además que es necesario utilizar System.Xml.Linq y no System.Linq (Landa Cosio N. , 2018), Linq es la base lógica del negocio por lo que es importante comprenderlo si se requieren realizar modificaciones a ese nivel de comprensión, a continuación se listan ejemplos de escritura (ver ilustración 5 y 6).

```
//Se crea un arreglo con datos
int[] numeros = { 1, 5, 9, 8, 6, 3, 2};
//Se realiza el query
IEnumerable<int> valores = from n in numeros
                           where n > 3 && n < 9
                           select n;
//Se muestran los resultado por consola
foreach (int i in valores)
{
    Console.WriteLine(i);
}
```

Ilustración 5. Linq para objetos (gráfico de Javier David Banda Villeda).

```
//Construcción funcional para XML
XElement documento = new XElement("Alumnos",
    new XElement("Ana", new XAttribute("ID", "10100"),
        new XElement("Curso", "Administracion"),
        new XElement("Promedio", "10")
    ),//Fin de ANA el primer elemento
    new XElement("Luis", new XAttribute("ID", "35350"),
        new XElement("Curso", "Programacion"),
        new XElement("Promedio", "9.5")
    )//Fin de Luis
);//Fin de alumnos
```

Ilustración 6. Linq to XML bajo escritura funcional (gráfico de Javier David Banda Villeda).

VII. Resultados

En el presente apartado se hará mención de los comentarios finales que dejó como resultado el desarrollo del proyecto, se mencionan algunas conclusiones relacionadas con los objetivos planteados al inicio del presente documento, posteriormente se darán a conocer algunas recomendaciones para aquellas personas que deseen retomar el desarrollo para la empresa, se muestra la bibliografía consultada y que hace referencia a algunos temas específicos de la investigación relacionada con la parte teórica, se encuentra además un glosario para identificar aquellos conceptos que no son tan sencillos de interpretar para el público general, en el apartado final se muestra material que sirve de referencia al proyecto, pero que no puede estar descrito en ninguno de los apartados principales, como es el caso del manual de usuario, descripción de clases y arquitectura.

A continuación, se enlista el requerimiento específico y la acción que se toma en cuenta, posteriormente se mostrará una imagen en donde se visualice el cambio en el desarrollo del programa.

VII.1 Objetivo específico 1.

Reestructurar el patrón de diseño de la aplicación para aumentar su escalabilidad utilizando un modelo de n-capas (presentación, negocio, entidades y datos). Con ayuda de Visual Studio se realizó una estructura basada en bibliotecas de clases en las que se siguiera dicho patrón de clases, esta fue la base del desarrollo básicamente sin un patrón de diseño la refactorización de la aplicación no hubiese tenido sentido (ver ilustración 7).

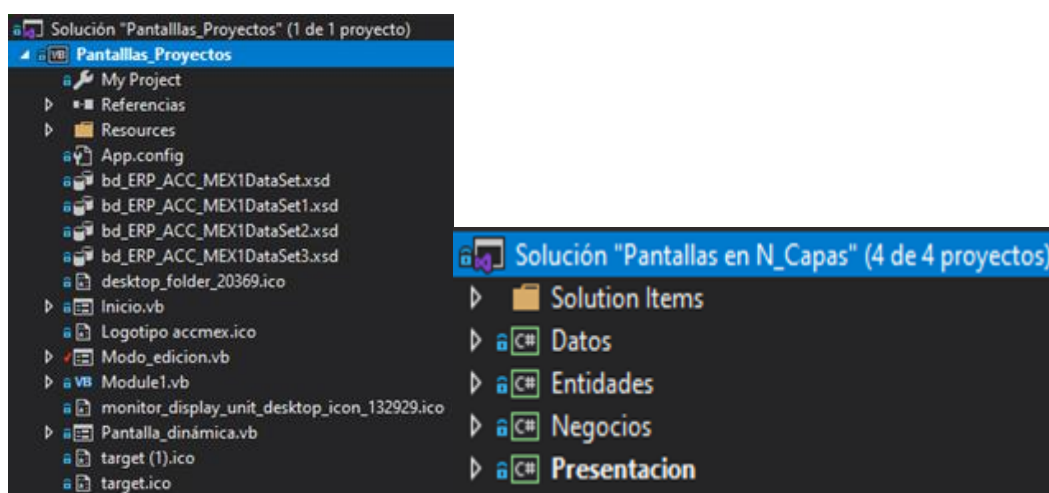


Ilustración 7. Comparativa entre patrones de diseño (derechos reservados a ACC MEX S.A. de C.V.).

VII.2 Objetivo específico 2.

Analizar los requerimientos respecto al sistema actual, conservando el objetivo principal de la aplicación (informar sobre el estado en producción). Se conservó gran parte de la funcionalidad del programa, tratando de visualizarlo como una estructura de bloques para abstraer cada parte de la funcionalidad y volver a crearla (ver ilustración 8 y 9).

The screenshot shows the 'Modo edición' window of the ACC MEX software. The header includes the ACC MEX logo, the date 'jueves, 26 de noviembre de 2020 11:36:19 a. m.', and the Mexican flag. The main area is divided into several sections:

- Project Information:** Includes fields for 'Proyecto' (TABLEROS SILAO PCNS01-0622U-), 'ADP' (sdAS), 'COP' (asd), 'Cliente' (asd), and 'Lider de proyecto' (250 Victor Anaya).
- Timeline and Status:** Includes 'Fecha de inicio' (martes, 20 de agosto de 2019), 'Fecha de entrega' (martes, 20 de agosto de 2019), 'Estado del proyecto general' (Por iniciar), 'Proceso Actual' (Montaje y ensamble 1), and 'Proceso Siguiente' (Montaje y ensamble 1).
- Progress and Comments:** Includes 'Piezas totales de proyecto' (2), 'Piezas terminadas de proyecto' (1), a progress bar at 50%, and a 'Comentarios del proyecto' field with the text 'asd'.
- Table:** A table with columns 'Tareas de línea', 'Responsable', 'Estatus', 'Prioridad', and 'Fecha'. It contains one row with 'asd' as the task, '200 German Tomas' as the responsible person, 'En proceso' as the status, 'Alta' as the priority, and 'asdasd' as the date.

At the bottom, there are buttons for 'Editar', 'Guardar y actualizar', 'Limpiar datos', 'Salir', 'Avanza', 'Atro', and 'Paso de E'.

Ilustración 8. Programa original (derechos reservados a ACC MEX S.A. de C.V.)

The screenshot shows the refactored version of the ACC MEX software. The header includes the ACC MEX logo, the date 'jueves, 26 de noviembre de 2020', the time '11:46:38', and the Mexican flag. The main area is divided into several sections:

- Project Information:** Includes fields for 'Proyecto' (CCM BEMS), 'O.P.', 'L.A.B.' (Cliente recoge), 'Cliente', and 'Lider de proyecto'.
- Timeline and Status:** Includes 'Fecha de inicio' (miércoles, 4 de septiembre de 20), 'Fecha de entrega' (viernes, 29 de noviembre de 20), 'Estado del proyecto' (Stand by), 'Proceso actual' (Montaje y ensamble 2), and 'Proceso siguiente' (Montaje y ensamble 2).
- Progress and Comments:** Includes 'Piezas totales del proyecto' (0), 'Piezas terminadas del proyecto' (0), a progress bar at 20%, and a 'Comentario' field with the text 'ENTREGA CON NOTIFICACION DE CAMBIO DE BUS PF'.
- Table:** A table with columns 'Tareas de Línea', 'Responsable', 'Foto', 'Estatus', 'Prioridad', and 'Fecha'. It contains one row with 'asd' as the task, '200 German Tomas' as the responsible person, 'En proceso' as the status, 'Alta' as the priority, and 'asdasd' as the date.

At the bottom, there are buttons for 'Editar', 'Guardar y actualizar', 'Limpiar datos', 'Salir', 'Avanza', 'Atro', and 'Paso de E'.

Ilustración 9. Programa refactorizado, se conservaron muchos aspectos funcionales (derechos reservados a ACC MEX S.A. de C.V.).

VII.3 Objetivo específico 3.

Conocer el entorno de desarrollo en C# para realizar una refactorización de código desde VB. Fue necesario realizar una capacitación por cuenta propia de cada miembro del equipo de trabajo en el entorno de desarrollo de Visual Studio y las tecnologías que utiliza para el desarrollo de aplicaciones de escritorio, el lenguaje de Visual Basic solo sirvió de referencia para conocer cómo estaba construida la programación (ver ilustración 10).

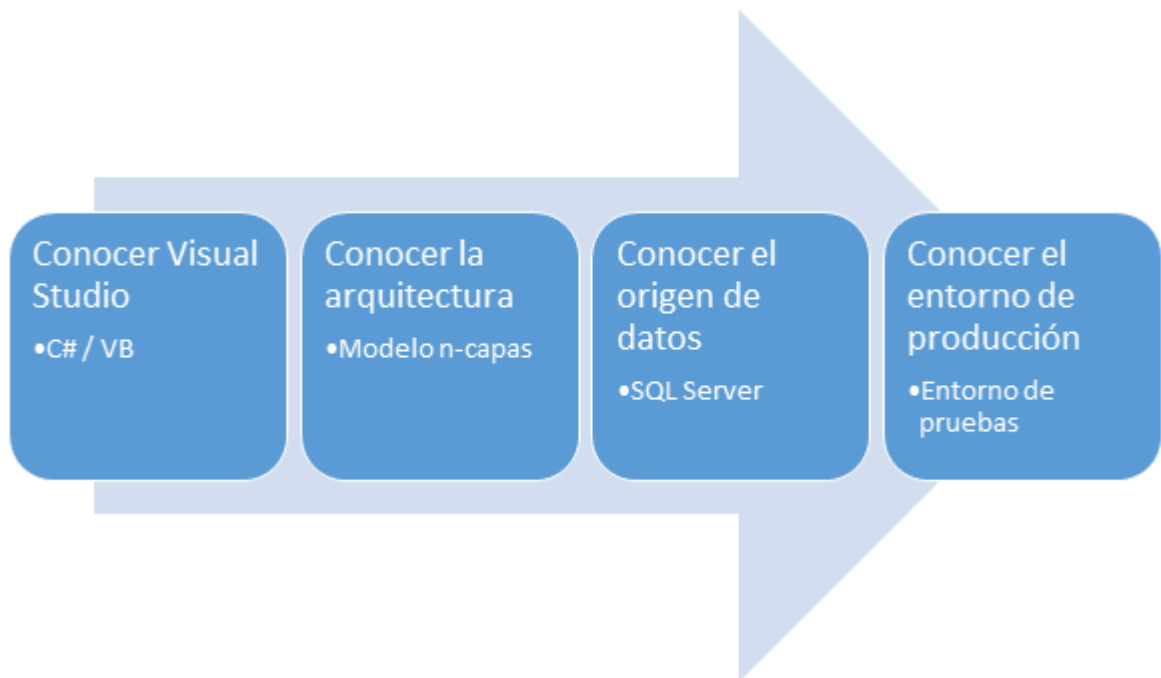


Ilustración 10. Marco de trabajo general (gráfico propio).

VII.4 Objetivo específico 4.

Implementar un diseño orientado a UI/UX para facilitar la comprensión de la información. El diseño orientado hacia una experiencia que los usuarios encuentren familiar e intuitiva facilita muchos temas de comprensión es por ello que la interfaz tiene una paleta de colores que facilite encontrar los elementos dispuestos en pantalla, controles con algunas animaciones para resaltar elementos y uso de programación asíncrona (ver ilustración 11 y 12).

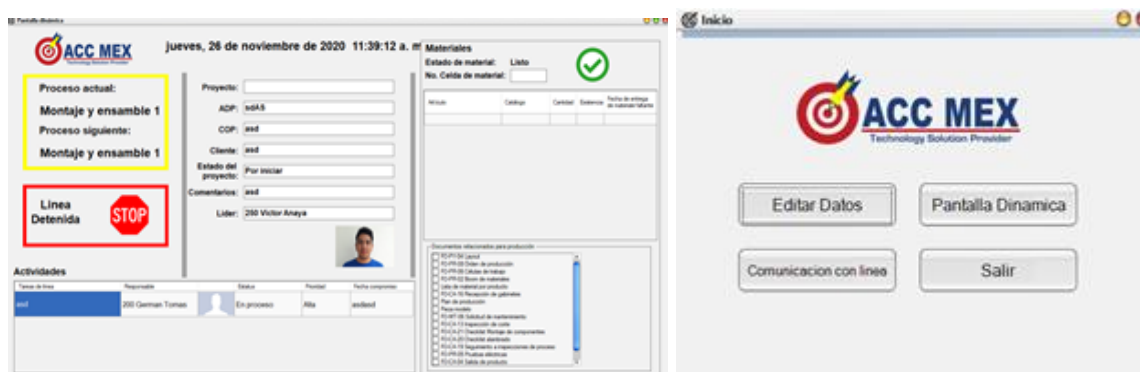


Ilustración 11. Aspecto gráfico original (derechos reservados a ACC MEX S.A. de C.V.).

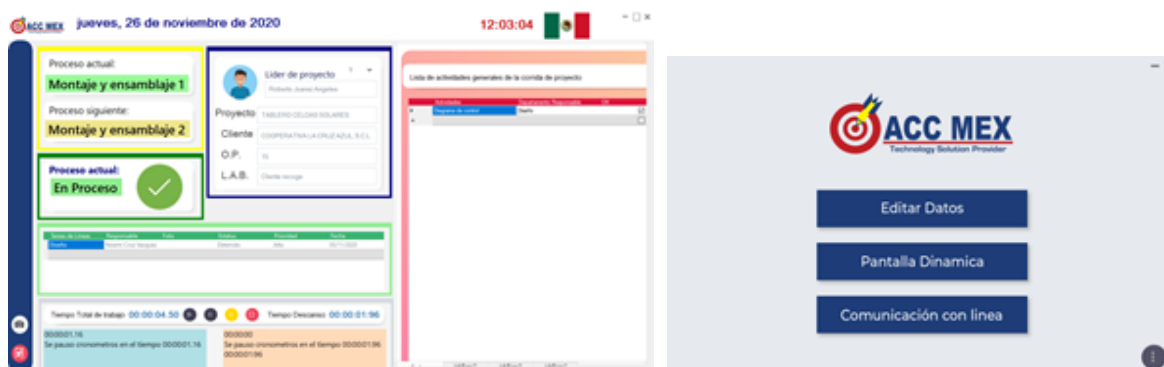


Ilustración 12. Apartado gráfico actualizado (derechos reservados a ACC MEX S.A. de C.V.).

VII.5 Objetivo específico 5.

Segmentar correctamente el origen de datos local con archivos de configuración y de base de datos utilizando ADO.NET para evitar redundancia. Existe una pequeña diferencia entre el origen de datos y el lugar donde se almacenarán los datos, en este proyecto particular la primera parte se realiza con datos de base de datos y archivos de configuración que contienen datos predeterminados (ver ilustración 13).

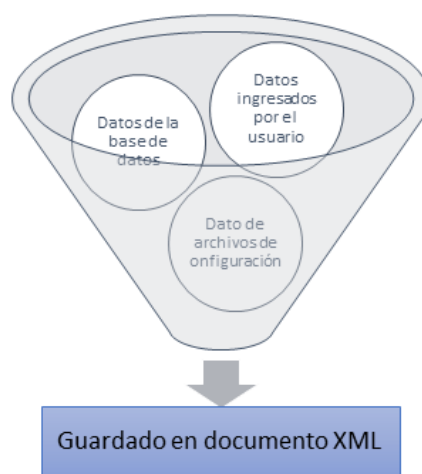


Ilustración 13. Segmentación de los datos (gráfico propio).

VII.6 Objetivo específico 6.

Crear e implementar un DLL para manipular procedimientos almacenados de una manera más amena. El uso de procedimientos almacenados demanda una conexión activa al servidor que posteriormente se cierra, utilizar ADO.NET como método de conexión puede llegar a mantener viva la sesión en más de una vez si no se manipulan correctamente los recursos, es por ello que se desarrolló un DLL que pudiera manipular ese acceso a datos de forma más sencilla y sobre todo asegurándose de que se elimina la conexión a SQL Server, un punto importante es que gracias al DLL es posible implementar otro tipo de conexión hacia otra base de datos en caso de que se quisiera migrar (ver ilustración 14).

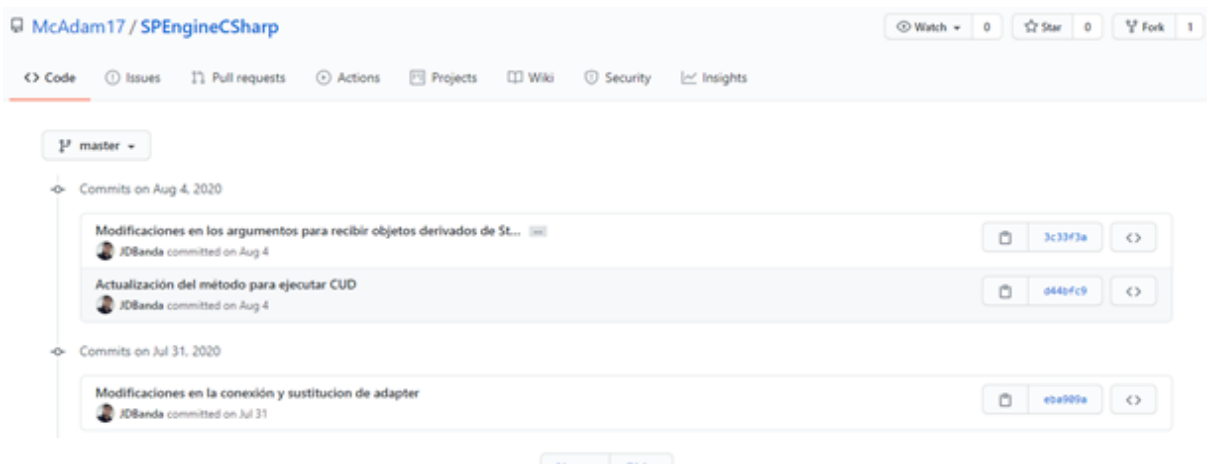


Ilustración 14. Aportaciones al DLL SPEngineCSharp (gráfico de Javier David Banda Villeda).

VII.7 Objetivo específico 7.

Implementar librerías específicas de C# para manipular componentes y colecciones de datos de XML. Se utilizaron las librerías que permiten llevar a cabo LINQ to XML, programación asíncrona, componentes de la GUI, entre otras librerías (ver ilustración 15).


```

public static List<ent_MetaSemanalXML> Cargar(string path, string pantalla, List<
ent_MetaSemanalXML> lista)
{
    try
    {
        //Leer el archivo
        XElement xElement = XElement.Load(path);
        //Objeto
        ent_MetaSemanalXML ent_;
        //Consulta LINQ
        var result = xElement.Descendants(pantalla).Descendants("Metas")
            .Elements().Select(a => a.Elements()).ToList();
        //Iterar sobre los elementos
        foreach (var item in result)
        {
            ent_ = new ent_MetaSemanalXML
            {
                MetaSemanal = item.ElementAt(0).Value,
                okMetaSemanal = Convert.ToBoolean(item.ElementAt(1).Value)
            };
            lista.Add(ent_);
        }
        return lista;
    }
    catch (Exception)
    {
        return null;
        throw;
    }
}

```

Ilustración 15. Ejemplo de código con LINQ to XML (derechos reservados a ACC MEX S.A. de C.V.).

VII.8 Objetivo específico 8.

Manejar un control de versiones con Git para manipular los cambios y el trabajo remoto. Git y Github se convirtieron en herramientas indispensables para trabajar debido a la facilidad para manipular los cambios, mostrarlos a otras personas relacionadas con el entorno de trabajo, realizar revisiones de código y detectar errores (ver ilustración 16).

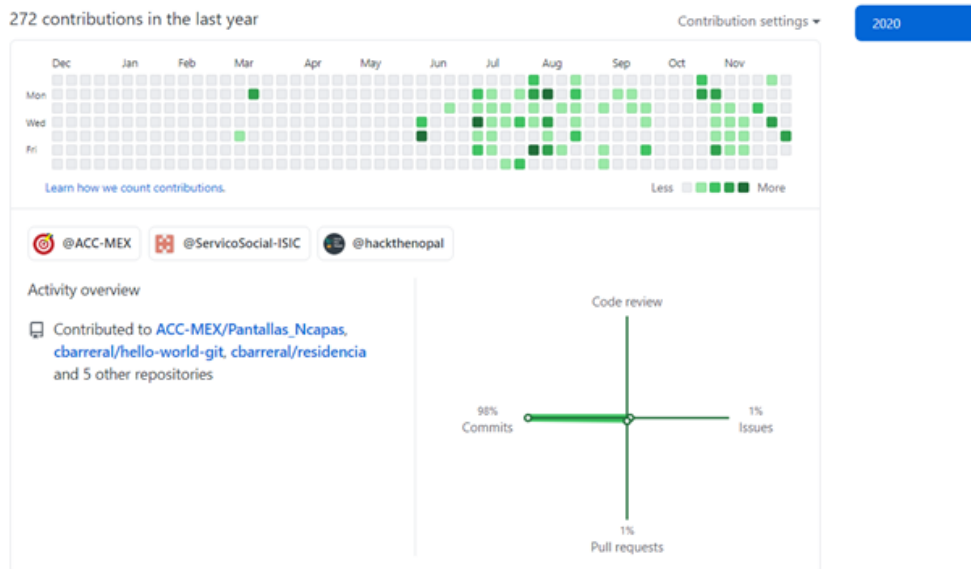


Ilustración 16. Contribuciones a los repositorios del proyecto (gráfico propio).

VII.9 Objetivo específico 9.

Implementar operaciones CRUD a los módulos correspondientes tanto para componentes como para XML. Las operaciones denominadas CRUD ya descritas en el documento, se realizan a nivel de frontend y de backend, al cumplir con este objetivo se cuenta con la mayor parte de la funcionalidad del proyecto (ver ilustración 17).



Ilustración 17. Operaciones CRUD (derechos reservados a ACC MEX S.A. de C.V.).

VII.10 Objetivo específico 10.

Documentar constantemente los cambios que se realicen para conservar un código más limpio. La documentación de código además de realizarse en el manual de usuario (consultar anexos XI.1.2), también se realiza dentro del propio código explicando cual es la funcionalidad clave de los métodos y a su vez en GitHub se pueden notar como se llevaron a cabo los cambios y de que se trataron.

VIII. Conclusiones

Se puede determinar que el proyecto terminó adecuadamente incluyendo la fase de pruebas, lo cual dio la pauta a las personas encargadas del área de proyectos de la empresa para implementar el sistema en producción, se desconoce con exactitud qué impacto positivo pueda tener frente a los inversionistas, debido a temas de contingencia sanitaria que acontecen en el periodo de tiempo que se presenta el proyecto, sin embargo se puede mencionar que se ha cumplido correctamente con el desarrollo de una aplicación para la línea de producción en ACC MEX S.A De C.V. y que cuenta con características que la vuelven competente y sencilla de mantener.

La realización de este proyecto permitió a los participantes (Carlos Alberto Barrera Lugo y Javier David Banda Villeda) comprender el trabajo por etapas y el control de los mismos, ya que esto a corto, mediano y largo plazo conllevan a agilizar las tareas y optimización de las mismas, todo comienza desde llevar una documentación adecuada, fijar tiempos y respetarlos, saber cómo se va y hacia donde es vital para ello registrar y adherirse a un control del progreso es incondicional, analizar y reflexionar al final de cada etapa las experiencias ganadas y lo que se puede mejorar, mantener una buena actitud y resolver inconsistencias o detalles del proyecto antes de que estos sean problemas, ayudan a tener una buena práctica de trabajo que apunta al éxito de cada proyecto.

IX. Recomendaciones

Es muy cierto que las tecnologías para crear aplicaciones de escritorio, entre las más usadas son Windows Forms y Windows Presentation Foundation (WPF), Windows Forms es la forma tradicional de crear formularios o vistas, en cambio WPF se basa en XAML es decir una notación xml para describir interfaces de usuario, con WPF se puede realizar diseños muy avanzados sin recurrir a GDI (Graphics Device Interface) , incluyendo una mejor presentación, además que WPF es el sucesor de Windows Forms

X. Competencias desarrolladas

Para poder llevar a cabo este proyecto fue necesario desarrollar las competencias para conocer el entorno de trabajo de Visual Studio y poder entender cómo se integran las herramientas para el trabajo colaborativo, en el caso de este proyecto la herramienta colaborativa utilizada es Git (para gestionar archivos locales) y GitHub (como el almacenamiento remoto o repositorio).

A su vez es muy importante comprender que el lenguaje de programación principal es C# del cual se necesitan conceptos teóricos - prácticos de programación orientada a objetos, comprensión de clases estáticas y abstractas, además se requieren algunos fundamentos de programación funcional. C# contiene una variedad de herramientas para integrar la base de datos en sus aplicaciones, por este motivo es necesario conocer el funcionamiento de ADO.Net y comunicaciones con puertos seriales. Para optimizar procesos de carga es necesario implementar programación asíncrona.

En cuestión de diseño de arquitectura de software es necesario comprender que el proyecto deberá seguir una estructura de 4 capas:

- Datos: Comunicación con base de datos.
- Entidades: Abstracción de los elementos físicos en propiedades.
- Negocios: Lógica de negocios del programa donde se realizan operaciones y se preparan los datos para ser mostrados.
- Presentación: Manipulación de los elementos visuales.

Para llevar a cabo la implementación del software será necesario realizar una serie de pruebas funcionales y considerar la retroalimentación de los usuarios para adaptar el programa a sus necesidades, este proyecto solamente será utilizado por la línea de producción y su funcionamiento está limitado a la visualización de información.

Referente al diseño gráfico y la experiencia de usuario (UI/UX) es necesario comprender que el proyecto necesita una fluidez de componentes y una agradable combinación de colores para lograr que el ambiente sea intuitivo para el usuario y brindar una experiencia favorable a la interacción del usuario con el software dando como resultado una percepción positiva.

Bibliografía

- Anderson, D. J., & Carmichel, A. (2016). *Essential Kanban Condensed*. Seattle: LEAN Kanban University Press.
- Arnold Cathalifaud, M., & Osorio, F. (1998). Introducción a los Conceptos Básicos de la Teoría General de Sistemas. *Cinta de Moebio*(3). Obtenido de <https://www.redalyc.org/pdf/101/10100306.pdf>
- Git. (2014). *Documentation: 1.1 Inicio - Sobre el control de versiones*. Obtenido de Git: <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones>
- Microsoft. (2018). *Documentation: Get started with WPF*. Obtenido de Microsoft:
<https://docs.microsoft.com/en-us/visualstudio/designers/getting-started-with-pf?view=vs-2019>
- ACC MEX S.A C.V. Technology Solution Provider. Obtenido de ACC MEX S.A C.V:
<https://accmex.com.mx/>
- Microsoft. (2018). *Documentation: Windows GDI*. Obtenido de Microsoft:
<https://docs.microsoft.com/en-us/windows/win32/gdi/windows-gdi>
- Gómez Fuentes, M. d. (2013). *Notas del curso BASES DE DATOS*. CDMX: Casa abierta al tiempo UAM Cuajimalpa.
- Gonzalez Godino, F. (2013). *Diseño de una arquitectura escalable y de alta disponibilidad para un sistema middleware*. Obtenido de Universidad Politécnica de Madrid:
http://oa.upm.es/32647/1/PFC_FERNANDO_GODINO_GONZALEZ.pdf
- Gorman, K., Hirt, A., Noderer, D., Rowland-Jones, J., Sirpal, A., Ryan, D., & Woody, B. (2019). *Introducing Microsoft SQL Server 2019. Reliability, scalability, and security both on premises and in the cloud*. Birmingham: Packt Publishing.
- Johansen Bertoglio, O. (1993). *Introducción general a la teoría general de sistemas*. Santiago: LIMUSA GRUPO NORIEGA EDITORES.
- Landa Cosio, N. (25 de Febrero de 2018). LINQ C# en español, básico. *Este es un curso básico de LINQ con C#, nos enfocamos principalmente en LINQ para objetos y XML*. Obtenido de <https://www.youtube.com/playlist?list=PLM-p96nOrGcYGfYpStN3jnaTulosO3V5>

- Landa Cosio, N. A. (2018). Tutorial XML en español. *Curso básico de XML para entenderlo de forma rápida y sencilla*. Obtenido de <https://www.youtube.com/playlist?list=PLM-p96nOrGcYb96AMy3VdUN8fo-sVAx7K>
- López Menéndez de Jiménez, R. E. (Diciembre de 2015). Metodologías Ágiles de Desarrollo de Software Aplicadas a la Gestión de Proyectos Empresariales. *ITCA-FEPADE*, 6-11. Obtenido de <https://core.ac.uk/download/pdf/80296686.pdf>
- Luna López, C. A. (2012). *Estructura de datos*. Estado de México: Red tercer milenio.
- Merizalde Andrade, C. E., & Toledo Vallejo, A. F. (2013). *Repositorio Institucional de la Universidad Politécnica Salesiana*. Obtenido de Desarrollo e implementación de un sistema de escritorio para el control de producción: <https://dspace.ups.edu.ec/bitstream/123456789/4229/1/UPS-GT000382.pdf>
- Microsoft. (2020). *.NET*. Obtenido de Microsoft Documentation: <https://docs.microsoft.com/es-es>
- Mompó Camarasa, Á. (2017). *Repositorio Institucional Universitat Politècnica de València*. Obtenido de Estudio de sistema de control de versiones y uso de los mismos en proyectos de integración continua: <https://riunet.upv.es/bitstream/handle/10251/87104/MOMP%C3%93%20-%20Estudio%20de%20sistema%20de%20control%20de%20versiones%20y%20uso%20de%20los%20mismos%20en%20proyectos%20de%20integraci%C3%B3....pdf?sequence=1>
- Morero, F. (2000). *Introducción a la OOP*. Obtenido de Laboratorio de computación de la universidad Magallanes: <https://kataix.umag.cl/~ruribe/Utilidades/Introduccion%20a%20la%20Programacion%20Orientada%20a%20Objetos.pdf>
- Mozilla and contributors. (23 de Marzo de 2019). *CRUD*. Obtenido de MDN web docs: <https://developer.mozilla.org/es/docs/Glossary/CRUD>
- Pérez Filoche, Á. (Enero de 2020). *Archivo Digital Universidad Politécnica de Madrid*. Obtenido de Evoluciones y refactorizaciones de una aplicación Android para alfabetización digital: http://oa.upm.es/58085/1/TFG_ALVARO_PEREZ_FILOCHE.pdf
- Pítsica Marques, M. (2001). *SISTEMA DE INFORMACIÓN PARA LA GESTIÓN APLICADO EN LAS ENTIDADES FINANCIERAS*:

ESTUDIO EMPIRICO SANTA CATARINA. Obtenido de Universidad Complutense Madrid: <http://webs.ucm.es/BUCM/tesis//cee/ucm-t25237.pdf>

Pressman, R. (2010). *Ingeniería del software. Un enfoque práctico*. Ciudad de México: McGrawll Hill.

Project Management Institute. (2017). *AGILE PRACTICE GUIDE*. Pennsylvania: PMI Book Service.

Red Hat Inc. (2020). *El concepto de IDE*. Obtenido de MiddleWare: <https://www.redhat.com/es/topics/middleware/what-is-ide>

Refactoring Guru. (2014-2020). *Patrones de diseño*. Obtenido de Refactoring Guru: <https://refactoring.guru/es/design-patterns/what-is-pattern>

Richards, M. (2015). *Software architecture patterns*. Sebastopol, CA: O'Reilly Media, Inc.

SAP. (2020). *SAP History*. Obtenido de SAP: A 48-year history of success: <https://www.sap.com/corporate/en/company/history.html>

Sebastián Gómez, L. (2000). *Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación*. Obtenido de ResearchGate: https://www.researchgate.net/publication/228877430_Disenio_de_Intefaces_de_Usuario_Principios_Prototipos_y_Heuristics_para_Evaluacion

Stack Overflow Comunity. (07 de Marzo de 2017). *C# Notes for professionals*. Obtenido de GoalKicker.com: <https://goalkicker.com/CSharpBook/>

Suárez Rey, C. (2010). *Sistemas integrados de gestión*. Obtenido de Grupo Jean Monet de Competencia y desarrollo regional: http://www.gcd.udc.es/subido/catedra/presentaciones/economia_competencia_ii/nota_tecnica_sistemas_de_gestion_erp_carlos_suarez_rey_17-03-2010.pdf

Suryanarayana, G., Samarthayam, G., & Sharma, T. (2015). *Refactoring for software design smells. Managing technical debt*. Waltham, MA: Elsevier.

The Institute of Electrical and Electronics Engineers. (28 de Septiembre de 1990). IEEE Standard Glossary of Software Engineering Terminology. *IEEE Std 610.12-1990*. New York, New York, United States of America: IEEE Standards.

Yépez Llerena, E. D., & Armijos Guillen, K. F. (2020). *Repositorio de la UNACH*. Obtenido de APLICACIÓN DE LA METODOLOGÍA KANBAN EN EL DESARROLLO DEL SOFTWARE PARA GENERACIÓN, VALIDACIÓN Y ACTUALIZACIÓN DE REACTIVOS, INTEGRADO AL SISTEMA INFORMÁTICO DE CONTROL ACADÉMICO UNACH.: <http://dspace.unach.edu.ec/bitstream/51000/6457/1/APLICACION%20DE%20LA%20METODOLOGIA%20KANBAN%20EN%20EL%20DESARROLLO%20DEL%20SOFTWARE%20PARA%20GENERACION.pdf>

XI. Anexos

XI.1.1 Cronograma de actividades

Actividades	Julio					Agosto					Septiembre					Octubre					Noviembre				
	4	1	2	3	4	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Fases																									
INVESTIGACIÓN PRELIMINAR																									
Reconocer repositorios de GitHub, Configurar entorno de desarrollo para ingresar bajo SSH																									
Investigación profunda de GitHub																									
Investigación de Manejo de Windows Forms																									
Investigar y capacitación de la arquitectura de N-Capas																									
Investigar y comprender los estándares de Google de Material Desing, Optimización de elementos UI																									
Comprender la forma de trabajo en Visual Studio para pasar de Visual Studio a C#																									
Investigar e implementar la conexión con ADO .Net desde el proyecto nuevo																									
Investigación de LINQ para el tratamiento de manejo de datos																									
DETERMINACION DE REQUERIMIENTOS																									
Recopilación de información																									
Analizar la estructura del programa actual y planear un plan de refactorización de código																									
DISEÑO DEL SISTEMA																									
Creación de diseño en boceto para la creación de interfaz grafica																									
Creación de recursos para el software en Adobe Ilustrador cs 6 2020																									
Diseño de estructura de XML																									
DESARROLLO DEL SOFTWARE																									
Integración de librerías de estándares de Material desing, e implementar UX (experiencia de Usuario)																									
Creación de Formularios Windows Form usando el diseño de boceto																									
Apoyo en desarrollo de DLL que permite acceder a los procedimientos almacenados de forma mas amena																									
Integración de componentes adicionales a los formularios (Animaciones, Efectos, Diseño, Alertas)																									
Codificación funcional del sistema																									
Creación de módulos (captura de pantalla, enviar captura por Gmail) e implementación de archivos de configuración																									
Implementar archivos de configuración																									
Creación de la serialización de datos para guardar en un archivo XML																									
Refactorización de métodos que operan con componentes e integración de optimización de código y metos asincronos																									
PRUEBAS DEL SISTEMA																									
Trabajar en pruebas unitarias para implementar correctamente el DLL																									
Pruebas de guardado de datos en un archivo XML																									
Pruebas de captura de pantalla																									
Pruebas de envío de correo por Gmail																									
Implementación de pull request a la rama master del repositorio																									
IMPLEMENTACIÓN Y EVALUACIÓN																									
Entrega de primer avance																									
Entrega de segundo avance																									
Entrega de final al cliente																									
Implementación del sistema																									
DOCUMENTACIÓN																									
Firmar carta de confidencialidad en ACCMEX																									
Primer reporte y evaluación de residencia																									
Segundo reporte y evaluación de residencia																									
Tercer reporte y evaluación de residencia																									
Informe técnico																									
Manual de usuario																									

Ilustración 18. Cronograma de actividades. (grafico propio)

XI.1.2 Manual de usuario

En este anexo se desarrolla el manual técnico para comprender el funcionamiento adecuado del software.

Una vez que el proyecto ha sido iniciado desde la solución de Visual Studio o como un archivo ejecutable, se mostrará la siguiente pantalla (ver ilustración 22), el tamaño de la ventana varía de acuerdo a la resolución del equipo.



Ilustración 19. Inicio de la aplicación.

En algunas ocasiones se muestran cuadros de diálogo para visualizar la carga entre interfaces, si el usuario da clic en aceptar, simplemente se eliminarán lo mismo sucede con las notificaciones en la parte inferior derecha de la pantalla (ver ilustración 20).

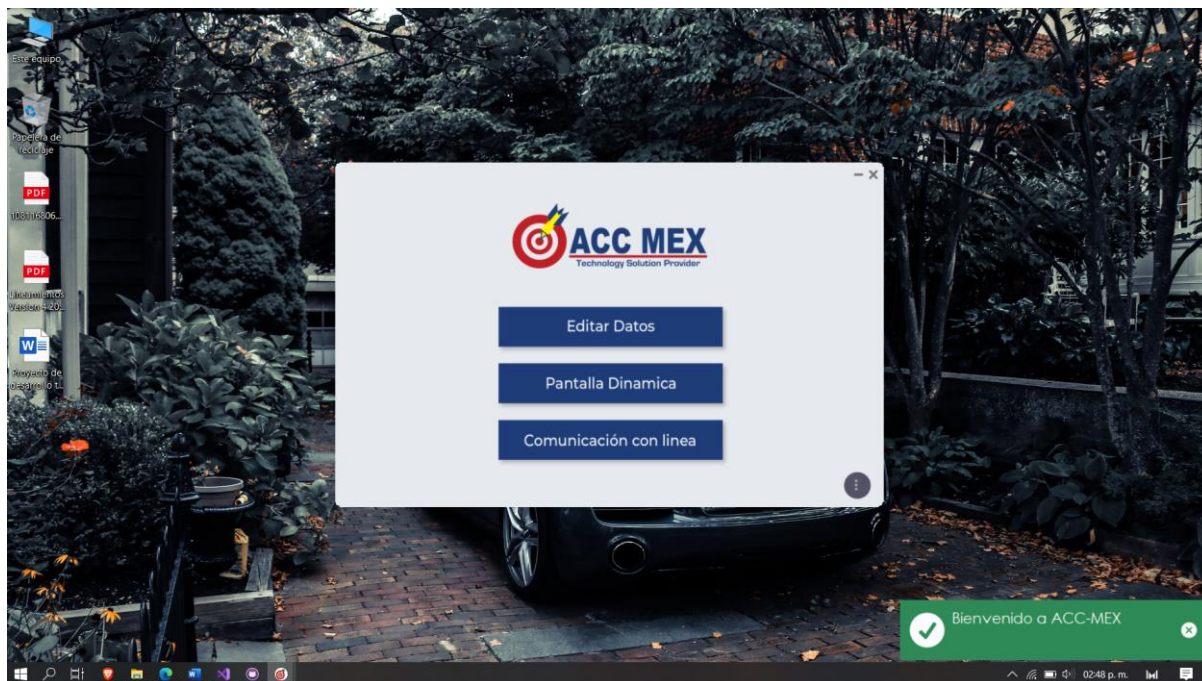


Ilustración 20. Alertas y cargas.

El menú principal cuenta con tres botones grandes de color azul en donde se llevan a cabo las principales operaciones del programa, además de existir un cuarto botón flotante para activar algunas funciones (ver ilustración 21).

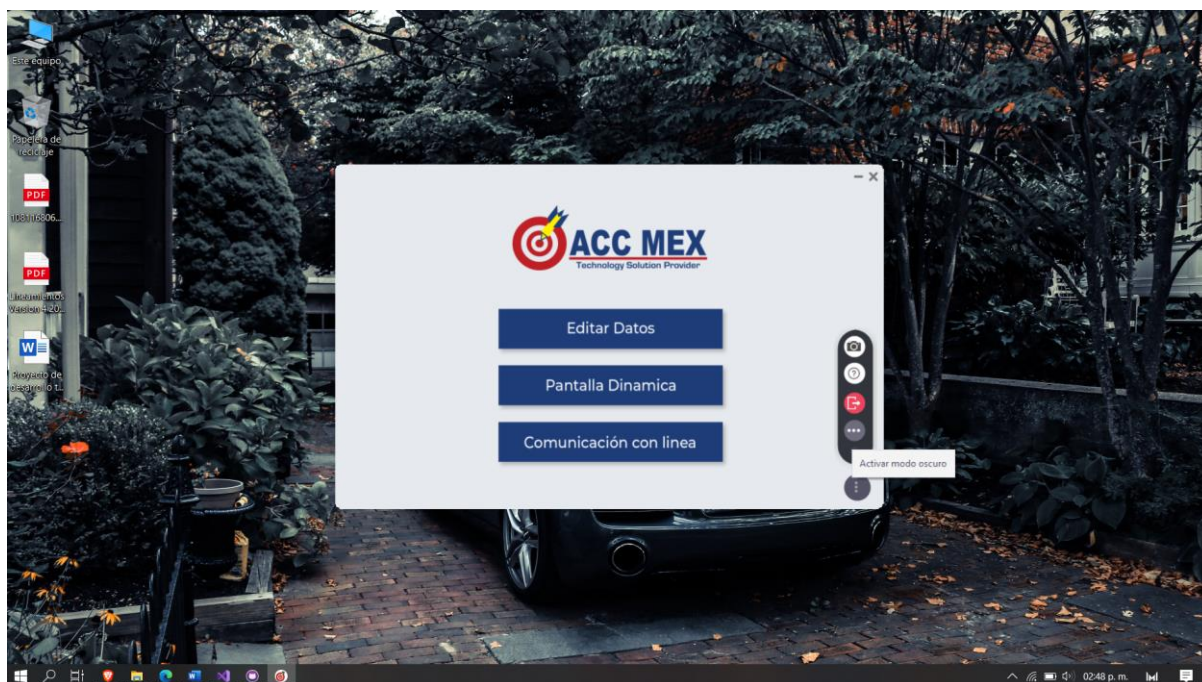


Ilustración 21. Menú principal.

El botón flotante puede ser activado o desactivado pulsando en los tres puntos verticales (E), para cambiar a un tema de color oscuro se utiliza el botón con los

tres puntos dispuesto de manera horizontal (D), si se desea salir de la aplicación sin presionar la pequeña letra x en la parte superior derecha es posible realizarlo con el botón rojo que tiene el icono de puerta (C), cuando se requiera solicitar ayuda se puede presionar el botón con signo de interrogación (B) y el último botón colocado en la parte superior realiza una captura de pantalla (A) estos botones se pueden utilizar en la pantalla principal únicamente, en otras pantallas existen los mismos elementos solo que con distintas funcionalidades (ver ilustración 22).

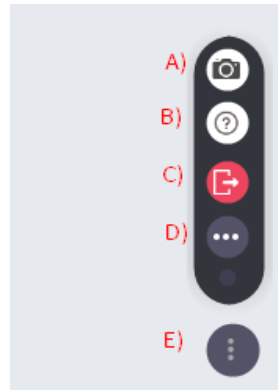


Ilustración 22. Botón flotante del menú principal.

El apartado para editar datos cuenta con una serie de pestañas (ver ilustración 23) que realizan funcionalidades específicas para cada caso, existen dos formas de cambiar entre pantallas, la primera de ellas actualiza el contenido de la pantalla (localizado en la parte superior) y el segundo no actualiza el contenido, lo que quiere decir que si se desean conservar datos antes de guardarlos esta podría ser la mejor alternativa (ver ilustración 24).

Ilustración 23. Ventana para editar datos.



Ilustración 24. Datos actualizados con ayuda de las pestañas superiores.

El primer apartado contiene varios campos por llenar (ver ilustración 25), algunos de ellos son datos precargados, lo que quiere decir que ya han sido registrados en base de datos y solo se están mostrando, si se realiza una edición solo se verá reflejada en el archivo local de almacenamiento, los datos mostrados se listan a continuación:

- a) Combo box en el que se selecciona el proyecto en el cual se quiera trabajar, algunos datos se precargan con esta selección, su origen es la base de datos lo que significa que no pueden ser borrados.
- b) (Orden de producción) hace referencia al Id asignado para producción.
- c) (Libre a bordo) existen tres opciones predeterminadas, Cliente recoge, ACC MEX envía y ACC MEX paquetería.
- d) (por motivos de confidencialidad no se coloca el nombre), este dato está precargado.
- e) Dato precargado para el nombre del líder
- f) Dato precargado para la fecha de inicio
- g) Dato precargado para la fecha entrega

- h) Combobox para seleccionar el proceso actual
- i) Combobox para seleccionar el proceso siguiente
- j) Dato precargado en el que se muestra un comentario
- k) Combobox para seleccionar el estado del proyecto
- l) Selector de número para indicar el total de piezas
- m) Selector de número para indicar el número de piezas terminadas

Ilustración 25. Campos estáticos de la pestaña Datos.

En el apartado posterior del programa (ver ilustración 26) se puede observar una serie de componentes que controlan otro tipo de acciones y datos explicados a continuación:

- a) Combo box en donde se selecciona el número de pantalla a presentar, al cambiar el número los datos se actualizarán automáticamente al de la pantalla seleccionada, en caso de estar vacío mostrará un mensaje y en caso de estar lleno mostrará otro mensaje, es importante mencionar que si se colocan datos y se cambia el índice estos no se guardarán y cambiará a otra pantalla.
- b) Las tareas de línea son tareas asignadas por el personal, este campo es de texto.
- c) Combobox para seleccionar el responsable contiene a todo el personal de ACC MEX registrado en el ERP.

- d) El apartado de foto no se guarda, es una funcionalidad que no se implementó.
- e) Combobox estatus del proyecto cambia el valor del selector por detenido, en proceso, esperando revisión y listo.
- f) Combobox que almacena datos de un archivo de configuración y tiene los valores alta, media y baja.
- g) Campo de texto en el que se puede almacenar la fecha en la que se asignó la tarea de línea.

The screenshot shows a form with several dynamic controls. At the top, there is a label 'No. de proyecto en pantalla' followed by the value '2' and a dropdown menu with 'A)' selected. Below this is a table with columns: 'Tareas de Línea', 'Responsable', 'Foto', 'Estatus', 'Prioridad', and 'Fecha'. Each column has a corresponding dropdown menu with a red letter label: 'B)', 'C)', 'D)', 'E)', 'F)', and 'G)' respectively.

Ilustración 26. Campos dinámicos y controles para operaciones de alta, baja y modificaciones.

La pestaña siguiente tiene por nombre Materiales, en esta se muestran los materiales disponibles del proyecto que se seleccionó en la pestaña datos (ver ilustración 27), en caso de no tener materiales relacionados mostrará una ventana en blanco del lado derecho (ver ilustración 28).

The screenshot shows the 'Materiales' tab. On the left, there is a form with fields for 'Fecha de pedido', 'Cantidad', and 'Seleccionar'. On the right, there is a table with the following data:

Id	Marca	Artículo	NoCatálogo	FechaEntrega
PCCA32-6710-20	ABB	DISYUNTOR DE COR.	2C5F204005R3630	10/07/2020 05:12 p. m.
PCCA32-6710-20	ABB	INTERRUPTOR TMA...	1SDA074643R1	10/07/2020 05:12 p. m.

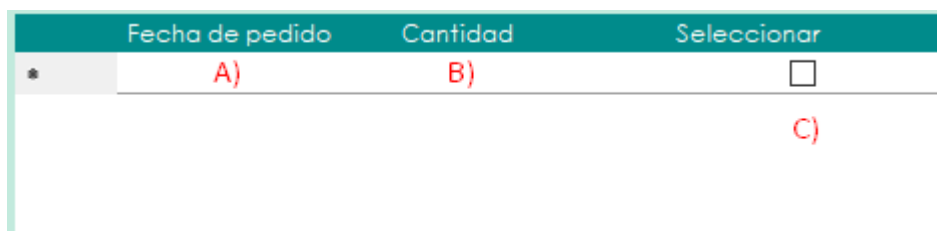
Ilustración 27. Pestaña Materiales con datos precargados.

The screenshot shows the 'Materiales' tab with the same form on the left as in the previous illustration. However, the table on the right is empty, indicating that no materials are currently displayed.

Ilustración 28. Pestaña Materiales sin datos.

La tabla en donde se pueden editar datos será la del lado izquierdo (ver ilustración 29), se guardarán únicamente las filas que sean iguales o menores al número de filas precargadas en la tabla derecha, al utilizar el programa se puede observar que no hay un límite para agregar las filas, sin embargo, las que superen el número no se guardarán, los campos guardados son:

- a) Campo de texto denominado fecha de pedido puede almacenar una fecha o cadena para indicar cuándo se entregará.
- b) Campo de texto denominado cantidad hace referencia al número de piezas que se pedirán.
- c) Campo seleccionable para indicar si se realizará o no la selección de materiales.



	Fecha de pedido	Cantidad	Seleccionar
*	A)	B)	<input type="checkbox"/>
			C)

Ilustración 29. Edición de datos en Materiales.

Dentro de la pestaña corrida del proyecto existen dos tablas, en la primera (ver ilustración 30) se especifican los campos que se almacenan relacionados a las actividades de la corrida del proyecto:

- a) Combo box precargado de un archivo de configuración en el cual se lista una serie de actividades que podría desempeñar un departamento.
- b) Combobox precargado gracias a la base de datos que muestra los departamentos existentes.
- c) El selector indica que está listo.
- d) Este otro selector indica que no está listo.

Actividad	Departamento Responsable	OK	NOK
* A) ▼	B) ▼	<input type="checkbox"/>	<input type="checkbox"/>
		C)	D)

Ilustración 30. Tabla para registrar las actividades.

La segunda tabla hace referencia al producto que estará marcado como meta semanal para entregar en la línea de producción, únicamente se puede seleccionar el nombre del elemento y si está listo o no (ver ilustración 31).

Id Producto	OK	NOK
* ▼	<input type="checkbox"/>	<input type="checkbox"/>

Ilustración 31. Tabla de meta semanal

Por último, en la pestaña de documentos relacionados (ver ilustración 35) se encuentra una única tabla que contiene dos tipos de datos a almacenar, el primero de ellos es el control de selección que define si está o no seleccionado el documento, el otro tipo de dato a guardar es el relacionado con los documentos los cuales tienen como origen un archivo de configuración.

Check	Documentos
* <input type="checkbox"/>	▼

Ilustración 32. Tabla de documentos relacionados.

Si el usuario ya ha cargado todos los elementos pertinentes dentro de la ventana para editar datos, puede hacer uso de los botones inferiores (ver ilustración 33) el primer botón tiene la función de salir y el segundo botón puede desplegar un menú flotante (ver ilustración 34).



Ilustración 33. Botones inferiores de la ventana editar datos.



Ilustración 34. Menú flotante.

Dicho menú flotante tiene la función de capturar una imagen, mostrar ayuda, guardar o editar datos y eliminarlos en ese orden.

El apartado siguiente hace referencia a la ventana dinámica (ver ilustración 35), para que esta ventana funcione correctamente es necesario dejar que cargue los componentes por algunos segundos, pues debido a que se cargan los recursos de forma asíncrona, la ventana está repartida en cuatro secciones.

ACC.MEX jueves, 26 de noviembre de 2020 10:44:31

Proceso actual:

Montaje y ensamblaje 1

Proceso siguiente:

Montaje y ensamblaje 2

Proceso actual:

En Proceso

Temas de Líneas	Responsable	Foto	Estatus	Prioridad	Fecha
Duafu	Noemi Cruz Vazquez		Detenido	Alta	05/11/2020

Tiempo Total de trabajo 00:00:00

Tiempo Descanso 00:00:00

Lider de proyecto

Roberto Juarez Angeles

Proyecto: TABLERO CELDAS SOLARES

Cliente: COOPERATIVA LA CRUZ AZUL, S.C.L.

O.P.: 15

L.A.B.: Cliente recoge

Fecha de inicio: martes, 21 de julio de 2020

Fecha de entrega: jueves, 20 de agosto de 2020

Piezas totales del proyecto: 5

Piezas terminadas del proyecto: 1

200

20%

Avance del proceso

Avance del proyecto

Meta Semanal

Id Producto	OK	NOK
DISYUNTOR DE ...	<input checked="" type="checkbox"/>	<input type="checkbox"/>
INTERRUPTOR T...	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>

Ilustración 35. Pantalla dinámica.

La primera sección es el menú lateral izquierdo (ver ilustración 36) en la cual se puede capturar la imagen y guardarse, enviarse por correo o cancelar dicha captura.



Ilustración 36. Menú lateral izquierdo.

La siguiente parte es un espacio estático en donde únicamente se podrá cambiar la pantalla cambiando el valor en el combobox indicado con color rojo (ver ilustración 37).

Proceso actual:	
Montaje y ensamblaje 1	

Proceso siguiente:	
Montaje y ensamblaje 2	

Proceso actual:	
En Proceso	

Lider de proyecto	
	1
Roberto Juarez Angeles	

Proyecto	
	TABLERO CELDAS SOLARES

Cliente	
	COOPERATIVA LA CRUZ AZUL, S.C.L.

O.P.	
	15

L.A.B.	
	Cliente recoge

Ilustración 37. Sección estática.

La parte siguiente está segmentada entre la o las tablas que van cambiando cada 5 segundos (ver ilustración 38), se puede notar fácilmente que los datos aquí mostrados son aquellos que se guardaron en la ventana de editar datos.

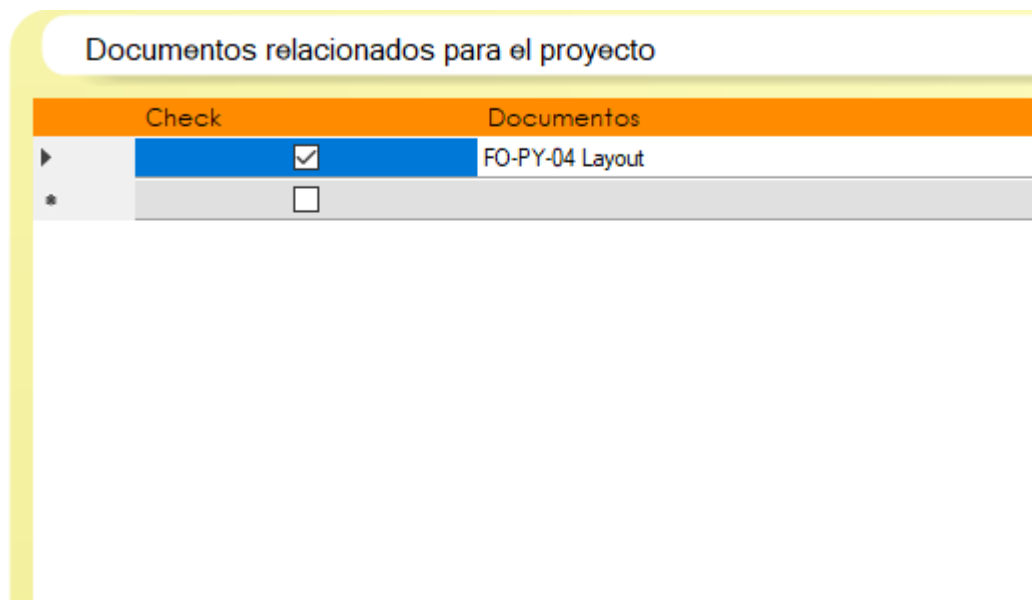


Ilustración 38. Parte dinámica 1.

Y la otra parte es la tabla que se obtiene de la pestaña datos del menú editar datos, no puede ser editado (ver ilustración 39).

Tareas de Líneas	Responsable	Foto	Estatus	Prioridad	Fecha
Diseño	Noemi Cruz Vazquez		Detenido	Alta	05/11/2020

Ilustración 39. Parte dinámica 2.

Finalmente se pueden observar controles de tiempo y algunos cuadros de texto, esta sección se realizó para conocer cuál es tiempo total de trabajo en la línea de producción (ver ilustración 40), a continuación, se explica qué función tiene cada botón y componente:

- Inicio del tiempo de trabajo.
- Pausa del tiempo de trabajo, entra en acción el tiempo de descanso.
- Pausa del tiempo de descanso, crea un paro en producción.
- Termina la producción y muestra el total de tiempo invertido en el componente E) y F)

- e) Muestra únicamente el tiempo que se ha registrado para el trabajo.
- f) Muestra únicamente el tiempo que se ha registrado para el descanso.

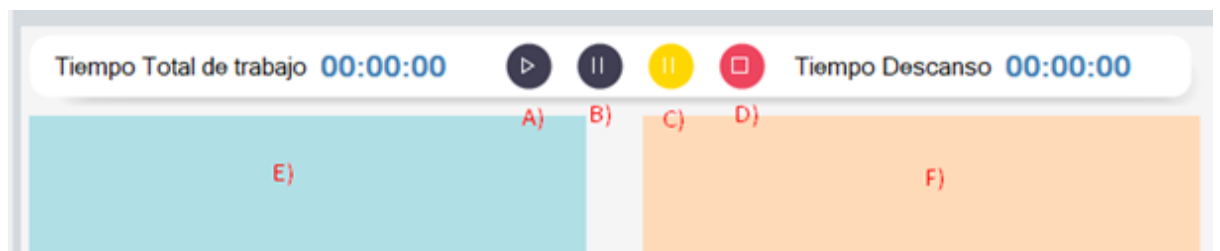


Ilustración 40. Controles de tiempo.