



Instituto Tecnológico Superior
de San Andrés Tuxtla

Titulación Integral

Implementar un web framework para visualización de
modelos tipo CAD en el proyecto Plant4D Web

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Maestro en Ingeniería

P R E S E N T A:

Ing. Darío Cubillos Martínez

A S E S O R:

MTI. Rogelio Enrique Telona Torres

**MAESTRÍA EN
INGENIERÍA**



San Andrés Tuxtla, Veracruz 2023



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



MAESTRÍA EN INGENIERÍA



TECNOLÓGICO NACIONAL DE MÉXICO

San Andrés Tuxtla, Veracruz, **08/DICIEMBRE/2023**
OFICIO No **Oficio No. DG/DA/SPI/81/2023**
Asunto: Autorización de impresión de Tesis

ING. DARIO CUBILLOS MARTINEZ
CANDIDATO AL GRADO DE MAESTRO EN INGENIERIA
PRESENTE

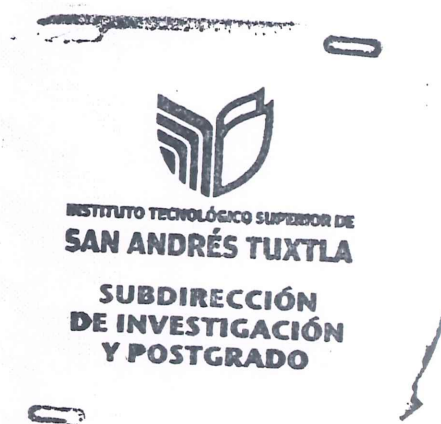
De acuerdo con los Lineamientos para la Operación de los Estudios de Posgrado en el Tecnológico Nacional de México y las disposiciones en este Instituto, habiendo cumplido con todas las indicaciones que la Comisión Revisora realizó con respecto a su Trabajo de Tesis titulado **"IMPLEMENTAR UN WEB FRAMEWORK PARA VISUALIZACIÓN DE MODELOS TIPO CAD EN EL PROYECTO PLANT4D WEB"**, la Subdirección de Posgrado e Investigación de este Instituto, concede la Autorización para que proceda a la impresión del mismo.

Sin otro particular, quedo de Usted.

ATENTAMENTE

"Transformarse para trascender"

ING. VICTOR PALMA CRUZ.
SUBDIRECTOR DE POSGRADO E INVESTIGACIÓN



C.p.p. *Dra. Verónica Guerrero Hernández-Presidenta del NAB Presente*
C.p. *Archivo*

ÍNDICE GENERAL

LISTA DE FIGURAS	3
LISTA DE TABLAS	4
AGRADECIMIENTOS	5
DEDICATORIAS.....	6
RESUMEN	7
ABSTRACT	8
INTRODUCCIÓN.....	9
CAPITULO 1.....	10
1.1. Planteamiento del problema	10
1.2. Pregunta de Investigación	11
1.3. Objetivo.....	11
1.1. Justificación.....	12
CAPITULO 2.....	13
2.1 Estado del arte.....	13
2.2 Marco teórico.....	18
CAPITULO 3.....	27
3.1 Scrum como metodología ágil.....	29
3.2 Determinación del tipo de estudio.....	32
3.3 Representación gráfica de los avances	36
CAPITULO 4.....	37
4.1 Descripción de los resultados del trabajo	37
4.2 Página de inicio de sesión del proyecto	38
4.3 Visor 3D del proyecto vista simple del modelo	38
4.4 Visor 3D del proyecto con propiedades del modelo	39
4.5 Visor 3D del proyecto barra de herramientas	39
4.6 Conclusiones	40
4.7 Recomendaciones	40
BIBLIOGRAFÍA	41
ANEXOS	43

LISTA DE FIGURAS

Figura 1 Diagrama que explica el controlador DirectDraw.....	13
Figura 2. Modelo 3D de la ciudad de Rotterdam representación del Proyecto CityGML.....	14
Figura 3. Mapa que muestra una nube de puntos para ayudar en la planificación urbana.....	14
Figura 4. Evolución de la metodología BIM.....	15
Figura 5. Estructura del framework desarrollado por IBM para el manejo de modelos en 3D.....	16
Figura 6. Configuración experimental para la construcción de hologramas de fibra PR.....	17
Figura 7. Ciclo de trabajo de scrum.....	29
Figura 8. Recolección de información.....	32
Figura 9. Azure ® Isologo.....	33
Figura 10. Inicio de sesión del visor.....	38
Figura 11. Vista por defecto de un modelo 3D en el web framework.....	38
Figura 12. Selección y panel informativo de una parte específica.....	39
Figura 13 Barra de herramientas del web framework.....	39
Figura 14 Antes y Después de los problemas de compatibilidad.....	43

LISTA DE TABLAS

Tabla 1. Plan de procesamiento y análisis de información.....	35
Tabla 2. Cronograma de actividades.....	37
Tabla 3. Plan de procesamiento y análisis de información.....	37

AGRADECIMIENTOS

“Primeramente a Dios que sin él nunca hubiera podido llegar a este punto en mi carrera académica y profesional, a los docentes que me guiaron en el proceso para obtener mi título universitario, en especial a mi asesor. A toda mi familia, en especial a mis padres y hermanos por creer en mí y mis amigos por apoyarme y darme fuerzas siempre que surgía un problema.”

DEDICATORIAS

“Dedicado a todos mis maestros a cada uno de los que he tenido el gusto aprender algo y en especial a mi padre por ser el pilar fundamental en mis logros académicos.”

RESUMEN

Actualmente las tecnologías web y de representación gráfica por computadora está muy avanzada en comparación a los primeros años de la informática por esa razón muchas empresas utilizan distintos tipos de tecnologías de visualización de modelos de tipo CAD, con la llegada de la web y sus enormes ventajas se ha comenzado a plantear el uso las tecnologías de representación gráfica en este medio, por lo que en esta presentación planteamos la planificación e implementación de estas nuevas tecnologías web para la visualización de modelos y gráficos del tipo CAD, en el contexto de una aplicación moderna utilizando un web framework desarrollado con una metodología ágil para ser más exactos la metodología scrum, que se utilizó para todo el desarrollo e implementación del proyecto.

ABSTRACT

Currently, web technologies and computer graphics are very advanced compared to the early years of computing, for that reason, many companies use different types of visualization technologies CAD models to do this many are aided by commercial 3D rendering engines that allow them to integrate this functionality to their existing systems, this has been very common a few decades ago, but with the advent of the web and its enormous advantages and even great challenges in this area so this publication raises the investigation of an implementation of these new web technologies for the visualization of CAD type models and graphics.

INTRODUCCIÓN

La presente investigación describe la planificación e implementación de un web framework para la visualización de modelos de tipo CAD tanto la planificación como y la tecnología descrita aquí están destinadas a la industria, debido a que las representaciones CAD (computer-aided design) o de diseño asistido por computadora generalmente tienen por objetivo un contexto profesional.

Este estudio y la información resultante de la investigación proporciona una lectura de valor para hacer una implementación exitosa en un contexto profesional y principalmente en las áreas donde la graficación es el eje central de la tecnología web.

Capitulo I. Aquí se describen los conocimientos más sintetizados relativos al proyecto, donde se plantea el problema, se hacen las preguntas específicas y fundamentales de la investigación, además de contener el objetivo general y los específicos que dan sentido a la justificación del proyecto.

Capitulo II. En este capítulo se despliegan todos los puntos teóricos concernientes al proyecto presentando primero el estado del arte donde a la fecha de este documento están las implementaciones más importantes en tecnología y teoría con relación a este proyecto y el marco teórico donde está toda la información concerniente a los conocimientos y tecnologías que sustentan este proyecto.

Capitulo III. Con ayuda de la metodología seleccionada, se establece un marco de trabajo para poder enfocar los esfuerzos del proyecto en terminar, entregar y evaluar cada componente que sea necesario para la realización de este software, asignando roles, tiempos y recursos a cada una de las fases, desde la primera etapa del proyecto que se denomina planeación hasta la retrospectiva donde se evalúa la ejecución de las fases anteriores, esto con el objetivo de que en cada fase del desarrollo del proyecto exista una mejora y resultados tangibles y medibles.

Capitulo IV. Se presentan los resultados del proyecto, como capturas de pantalla con el funcionamiento del software, además de interpretar y describir la solución de los principales retos.

CAPITULO 1

GENERALIDADES

1.1. Planteamiento del problema

Implementar un web framework en una plataforma web que tenga como función principal la visualización de modelos tipo CAD enfocado a una plataforma web de nombre Plant4D que vende la empresa CEA Systems, cuya función principal de este software es el manejo y administración de assets (activos) relacionados con la ingeniería, la visualización de varios tipos de gráficos y modelos por computadora.

Actualmente este visor funciona por medio de un software licenciado por un tercero, llamado *Hoops Communicator*, el cual posee ciertos errores relacionados con la visualización de los modelos 3D.

Uno de los problemas más importantes que presenta es la compatibilidad con otro software de *Hoops* llamado *Hoops Visualize* y se refleja en el escalado de los modelos entre estas dos plataformas, además, el sistema de coordenadas que maneja es muy diferente entre estos dos sistemas, es por estos detalles que la experiencia del usuario se ve afectada considerablemente, aunado a que es un software de terceros y con código fuente cerrado, la idea de implementar nuevas funcionalidades es imposible, quedando limitados a los cambios que la empresa Hoops desarrolle en sus nuevas versiones. Esto limita el equipo de desarrollo de la empresa CEA Systems, reflejándose en diversos retrasos. Por lo anterior se plantea el desarrollo del software Plant4DWeb que dependa de un web framework para el manejo de sus modelos en 3D.

1.2. Pregunta de Investigación

¿Cómo implementar un *web framework* para la visualización de modelos 3D de tipo CAD para la empresa CEA Systems?

¿Cómo mantener la compatibilidad en los modelos 3D de tipo CAD con sus datos, para su correcta representación y congruencia en todo el software de la empresa CEA Systems?

¿Cuánto tiempo se demorará el implementar el *web framework* para la empresa CEA Systems?

¿Cómo medir la calidad del software implementado en el *web framework* implementado para la empresa CEA Systems?

1.3. Objetivo

1.3.1. General

Implementar un *web framework* para visualización de modelos tipo CAD en el proyecto *Plant4DWeb*.

1.3.2. Específicos

- Investigar en diversas fuentes de información las características que debe contener el *web framework* para *Plant4DWeb*.
- Planear las actividades a realizar para la implementación del *web framework*.
- Desarrollar los módulos necesarios para la implementación del *web framework*.
- Implementar un conjunto de pruebas para revisar la calidad del código usado para el proyecto.
- Someter a pruebas el desarrollo hacer para revisar la compatibilidad con todas las aplicaciones relacionadas con el proyecto *Plant4DWeb*.

1.1. Justificación

La empresa Cea Systems cuenta con un amplio catálogo de clientes que dependen de la visualización del modelo generado en el Framework Hoops Communicator que actualmente es empleado por la empresa.

La compañía depende del software denominado *Hoops Communicator* que es proveído por un externo y al cual debe adaptarse Cea Systems. Las recientes actualizaciones aportadas al software 3D usado no cumplen con los requisitos necesarios para que el software Plant4DWeb funcione de manera adecuada en cuanto a la visualización de modelos 3D desarrollados.

Derivado de lo anterior se considera necesario la implementación de un framework que permita visualizar correctamente los modelos entre las aplicaciones de *Cea Systems*, además esta nueva implementación para la visualización de modelos 3D permitirá la incorporación de características adicionales, que brindará la posibilidad de adaptarse a las nuevas circunstancias permitiendo que este software otorgue una mejor experiencia al usuario final y la posibilidad de que los desarrolladores de *Cea Systems* puedan implementar o adaptar futuros cambios a este nuevo módulo. Estos son los principales inconvenientes que se busca resolver con este proyecto.

Considerando lo anterior, es notable que el impacto será relevante para la empresa Cea Systems, propietaria de este producto, ya que gracias a esto se ahorrará horas hombre con el uso de este *web framework* que será enriquecido con la visualización de distintos tipos de modelos y datos relacionados a los activos industriales con un enfoque web. Solventando en lo general todas las necesidades sus clientes.

Con este enfoque el proyecto tendrá una visión actual de las necesidades que requiere la industria para la administración de recursos e información según los más altos estándares de calidad y respuesta como menciona Ryo Yoshida de IBM en en su publicación *3D web environment for knowledge management* citó: *“Hoy, el objetivo general de la gestión del conocimiento es poner activos intelectuales en un ciclo de conocimiento, es decir, almacenar, recopilar, reutilizar, compartir y actualizar valiosa experiencia, sabiduría y conocimiento de las personas que pertenecen a un equipo de trabajo de manera eficiente y de forma visible, con el objetivo de mejorar la capacidad de respuesta, el tiempo del ciclo de desarrollo, la productividad y competencia.”* [2] lo que señala qué se espera del software en la actualidad relacionado con la web, este es el enfoque planteado por IBM.

CAPITULO 2

FUNDAMENTO TEÓRICO

2.1 Estado del arte

En los últimos años ha existido una enorme demanda en el software y hardware relacionado a la graficación en computadoras, esto dio origen a que diversas empresas dedicadas a la creación de software para el procesamiento de gráficos 3D buscaran estandarizar la tecnología de creación de gráficos mediante la creación de APIs, (acrónimo del inglés interfaz de programación de aplicaciones, conocida también por la sigla API, en inglés, application programming interface), un caso particular fue Microsoft que forzó su API (DirectX) a usarse en sus sistemas operativos, siendo al final la empresa gran ganadora de la estandarización del software. Otras empresas de la industria informática (Sun Microsystems, Hewlett-Packard e IBM, entre otros) desarrollaron sus APIs para gráficos para competir, pero al final de la década el resultado de estas tecnologías fue la creación de OpenGL. [2]

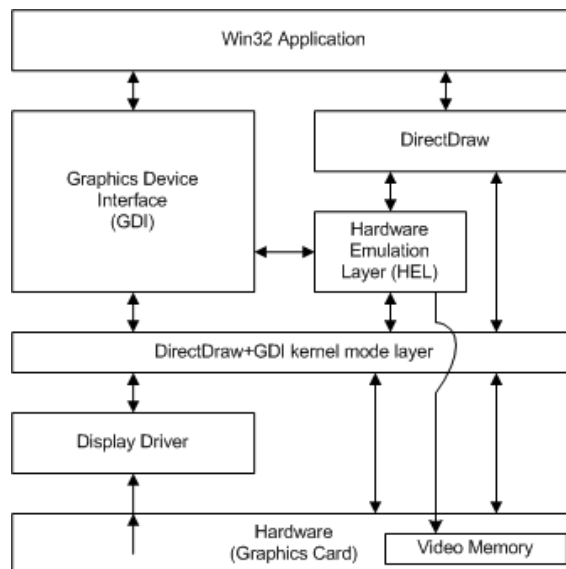


Figura 1 Diagrama que explica el controlador DirectDraw base de la API DirectX base de la comunicación tecnológica entre el software y hardware

Podemos encontrar muchas áreas de aplicación especialmente en la ingeniería, donde gobiernos o la industria privada han tenido excelentes resultados en emplear tecnología de este tipo como, en dos ciudades muy grandes como Berlín Alemania y Rotterdam, Holanda que lograron

representar ambas ciudades desde una web donde el reto está en la gran cantidad de información y modelos a manipular.[3]



Figura 2. Modelo 3D de la ciudad de Rotterdam representación del Proyecto CityGML

Dentro de la ingeniería civil otro proyecto muy importante es el de la universidad de Las Palmas de Gran Canaria en Las Palmas, España. donde gracias a la gran cantidad de datos abiertos se plantea la generación de modelos para simulaciones y predicciones que busca un mejor enfoque de la planificación urbana y espacios adaptados a las necesidades de los ciudadanos.[4]



Figura 3. Mapa que muestra una nube de puntos para ayudar en la planificación urbana en el Proyecto Generating 3D City Models from Open LiDAR

Muy cerca en Portugal en el departamento de Ingeniería Civil, Universidad de Aveiro, en ese departamento tienen como prioridad desarrollar los conocimientos prácticos que permiten la aceleración de la innovación en industrias como la construcción, ayudando a futuras investigaciones y su acción para madurar el modelo de esa universidad y su investigación de los datos relacionados a la construcción.

El modelado de información de edificios (BIM), como metodología ha venido siendo esencial; apoyando el desarrollo de la industria de la construcción, incluyendo el concepto/diseño y planificación/programación de los trabajos de construcción o el mantenimiento/gestión de edificios. En el sector de la construcción, BIM ha automatizado los procesos ofreciendo una gran ventaja al disminuir las discrepancias entre el modelos planificados y construidos, ahorrando dinero y tiempo y contribuyendo a evitar errores, limitar los riesgos y reducir las divergencias de costes.[5]

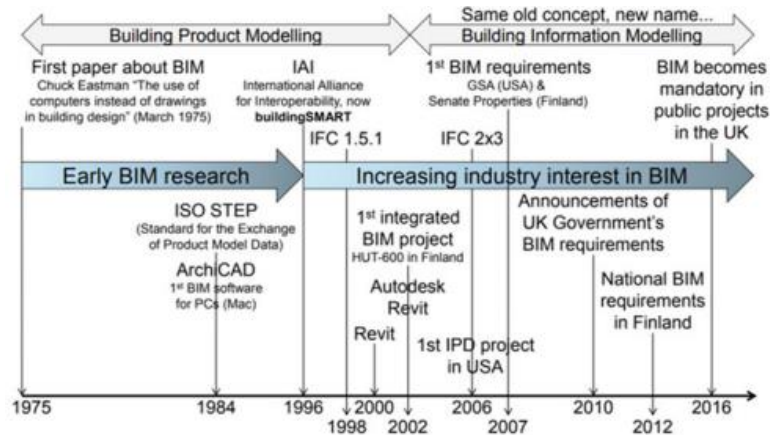


Figura 4. Evolución de la metodología BIM

En los usos muy particulares relacionados a la representación de modelos en 3d en el mundo como, por ejemplo, en el *Tokyo Research Laboratory*, *IBM Japan* donde ya poseen un *framework* para el procesamiento de información y visualización de 3D para la industria. En diversas áreas de la industria, la acumulación y la comunicación del conocimiento se está volviendo más importante que nunca.

Especialmente en las áreas de fabricación, este ciclo de conocimiento, que se puede aplicar a través de la discusión, la toma de decisiones, la planificación y la producción procesos, se ha acertado recientemente, y por lo tanto su eficiencia es más importante que nunca.

El autor de esta investigación señala que los contenidos se recogen a través de muchas discusiones y los participantes en otra discusión pueden recuperar varios objetos del contenido recopilado especificando un objeto clave. Además, considera que el primero de estos dos procesos cae en la categoría de colaboración web para múltiples usuarios comunicación, mientras que el segundo cae en el de búsqueda de geometría para la sistematización de objetos 3D. [6]

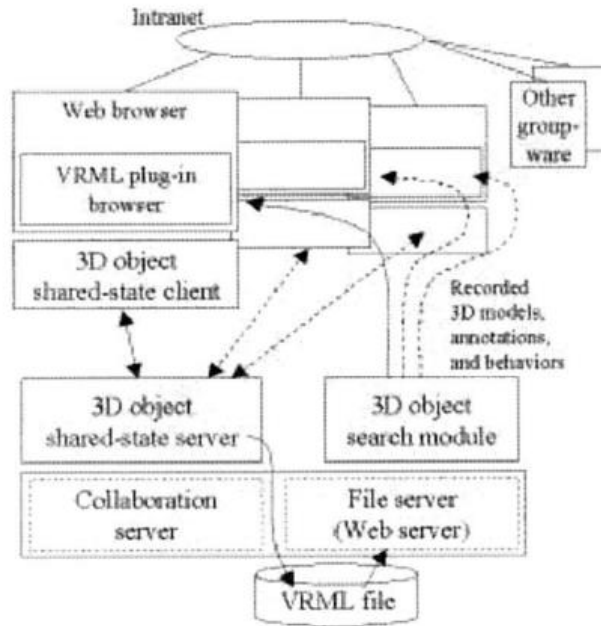


Figura 5. Estructura del framework desarrollado por IBM para el manejo de modelos en 3D.

Otro Proyecto muy significativo que aporta la flexibilidad de visualización en estos modelos podemos encontrarlo en Korea del sur, donde la escuela de ciencias computacionales en *Kyungpook, Daegu* que con sus investigaciones buscan emular el motor *OpenGL* en plataformas no diseñadas para ese trabajo lo que permitiría la flexibilidad de la representación de la información en general relacionada a los modelos 3D. [7]

Un uso muy novedoso e interesante es tomado por la sociedad internacional de cirugía plástica donde gracias al modelado en 3D se puede auxiliar mucho la práctica de la reconstrucción de glándulas mamarias en pacientes que hayan sufrido algún tipo de cáncer.[8]

Estos avances en el mundo han llevado a reconsiderar otros aspectos no directamente relacionados al modelado pero si al transporte y representación de estos datos como en la conferencia de *Holography: Origin, Rediscovery, Development* en Shangai China, donde el envío de hologramas por medio de fibra óptica aprovechando fenómenos físicos en el comportamiento de la transmisión permite la transmisión de estos datos de una forma sorprendentemente rápida y finalmente en *Technische Universitat Darmstadt* en Alemania, estudian la creación y el software disponible que dependiendo de las necesidades del proyecto en la universidad estos pueden ser una buena opción para el uso y desarrollo de la tecnología en aquel país. [9]

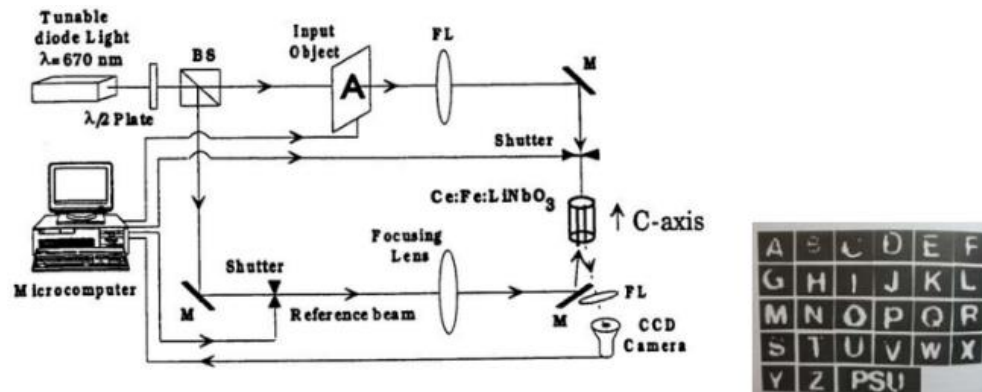


Figura 6. Configuración experimental para la construcción de hologramas de fibra PR.

Esta gran cantidad de innovaciones y proyectos interesantes determinan que existe un gran futuro en la representación y procesamiento de datos en el manejo de modelos en 3D, para planificación urbana de universidades o grandes ciudades incluso auxiliares en sectores muy diferentes, como la cirugía plástica, la aeronáutica, química y geografía se puede deducir que actualmente esta disciplina es muy socorrida para la toma de decisiones en todas estas áreas por tal motivo el manejo y representación de estos datos se ha convertido en prioridad para muchas organizaciones, gobiernos y universidades del mundo.

Hoy en día, el rápido desarrollo tecnológico para adquirir información; unido a las capacidades para procesar estos datos en un período relativamente corto de tiempo, permite la generación de modelos detallados que se convertirán en una parte esencial de la infraestructura de información moderna y, se puede utilizar para integrar diversos datos de diferentes fuentes para visualización accesible al público y muchas otras aplicaciones.

Estos modelos son representados en su mayoría gracias a el uso de motores gráficos, en la actualidad existen muchos para diferentes necesidades que tengan los proyectos a desarrollar estos motores ya que su función principal es realizar el proceso de creación de dichos gráficos, mediante cálculos y técnicas propias del motor.

Pero para crear esas aplicaciones se necesita una base robusta que pueda, gestionar los recursos informáticos y la presentación de esta información, por lo que recientemente es pertinente el uso de un *framework* web. Un *framework* para aplicaciones web es un marco de trabajo diseñado para apoyar el desarrollo de sitios web dinámicos, aplicaciones y servicios web. Este tipo de *frameworks* intenta aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web. Por ejemplo, muchos *framework* proporcionan bibliotecas para acceder a bases de datos, estructuras para plantillas y gestión de sesiones, y con frecuencia facilitan la reutilización de código.[10]

2.2 Marco teórico

El software es un conjunto de programas informáticos y de documentación y datos asociados [18], en contraste con el hardware, a partir del cual se construye el sistema y que realmente realiza el trabajo.

En el nivel más bajo de la programación, el código ejecutable consiste en instrucciones de lenguaje de máquina soportadas por un procesador individual, normalmente una unidad central de procesamiento (CPU) o una unidad de procesamiento gráfico (GPU). El lenguaje de máquina consiste en grupos de valores binarios que significan instrucciones del procesador que cambian el estado del ordenador respecto al estado anterior. Por ejemplo, una instrucción puede cambiar el valor almacenado en una ubicación concreta del ordenador, un efecto que no es directamente observable por el usuario. Una instrucción también puede invocar una de las muchas operaciones de entrada o salida, por ejemplo, mostrar algún texto en la pantalla del ordenador, provocando cambios de estado que deberían ser visibles para el usuario. El procesador ejecuta las instrucciones en el orden en que se proporcionan, a menos que se le indique que "salte" a una instrucción diferente, o sea interrumpido por el sistema operativo.

Existen distintos tipos de software, pero lo mejor es clasificarlos por su naturaleza de ejecución, siendo los dos principales de Escritorio y Web.

Aplicaciones de escritorio como procesadores de texto y navegadores web y son muy similares a las aplicaciones para smartphones y tabletas (llamadas popularmente "apps").

Aplicaciones web, que normalmente se ejecutan en el servidor web y dan salida a páginas web generadas dinámicamente para los navegadores web, utilizando, por ejemplo, PHP, Java, ASP.NET, o incluso JavaScript que se ejecuta en el servidor. En los tiempos modernos, estas aplicaciones suelen incluir algo de JavaScript para que se ejecute también en el navegador web, en cuyo caso suelen ejecutarse en parte en el servidor y en parte en el navegador web.

En el año 1961 William Fetter uso el término “gráficos por computadora” en 1961 para describir su trabajo en Boeing , fue una de las primeras interpretaciones pero la historia de los gráficos por computadora empieza en 1950 cuando un matemático de Iowa llamado Ben Laposky creo las primeras imágenes generadas por computadora estas imágenes fueron figuras de Lissajous que mostro en un osciloscopio, este acontecimiento fue muy temprano si consideramos que la historia de la electrónica digital moderna comenzó en 1940.

Hasta antes de 1961 el termino *modelo* solo se definía como una representación informativa de un objeto, persona o sistema y el término designaba originalmente los planos de un edificio en el inglés de finales del siglo XVI, y derivó a través del francés y el italiano en última instancia del latín *modulus*, una medida.

Los modelos pueden dividirse en modelos físicos (por ejemplo, un plano modelo) y modelos abstractos (por ejemplo, expresiones matemáticas que describen patrones de comportamiento). Los modelos abstractos o conceptuales son fundamentales para la filosofía de la ciencia.[19] [20]

En este último apartado podríamos encontrar a la abstracción que existe con los modelos ya sean 2D o 3D visualizados por computadora estas formas de visualizar los modelos se pueden comprender por saber el tipo de espacio en el que están representados 2D en una definición formal es un espacio euclidiano bidimensional y el 3D es un espacio tridimensional donde tomando como base el espacio euclidiano bidimensional se agrega un tercer parámetro para la representación gráfica.

El espacio euclidiano bidimensional o simplemente el espacio bidimensional (también conocido como espacio 2D, espacio 2 o espacio bidimensional) es un entorno geométrico en el que se requieren dos valores (llamados parámetros) para determinar la posición de un elemento (es decir, un punto) en el plano. El conjunto de pares de números reales (espacio de coordenadas reales) con una estructura adecuada suele ser el ejemplo canónico de un plano euclidiano, el espacio euclidiano bidimensional; para una generalización del concepto. El espacio bidimensional puede verse como una proyección del universo físico sobre un plano. Normalmente, se considera un espacio euclidiano y las dos dimensiones se denominan longitud y anchura.

El espacio tridimensional (también:3D,) es un entorno geométrico en el que se requieren tres valores (llamados parámetros) para determinar la posición de un elemento (es decir, un punto). Este es el significado informal del término dimensión.

En 1959, con trabajo conjunto *General Motors* e *IBM* produjeron la primera

pieza de software de dibujo, el DAC-1 (Diseño aumentado por computadoras). Los usuarios pudieron ver el automóvil en perspectiva y rotarlo.

Con este primer acercamiento al diseño asistido por computadoras se empezó a designar el término *Computer-aided design* o en su traducción al español *diseño asistido por computador* simplemente a simplemente *CAD* y esto es el uso de ordenadores (o estaciones de trabajo) para ayudar en la creación, modificación, análisis u optimización de un diseño. Este software se utiliza para aumentar la productividad del diseñador, mejorar la calidad del diseño, mejorar las comunicaciones a través de la documentación y crear una base de datos para la fabricación. Los diseños realizados a través del software CAD son útiles para proteger los productos y las invenciones cuando se utilizan en las solicitudes de patentes. El resultado del CAD suele ser en forma de archivos electrónicos para la impresión, el mecanizado u otras operaciones de fabricación. También se utilizan los términos dibujo asistido por ordenador (CAD) y diseño y dibujo asistido por ordenador (CADD).

Los inicios de los gráficos por computadora para el público fueron en la década de 1960 pero aún estaban fuera del alcance de la mayoría de los usuarios porque el hardware gráfico especial era caro.

No había computadoras personales ni estaciones de trabajo. Los usuarios tenían que pagar por el tiempo del mainframe por segundo o comprar caros miniordenadores. El resultado fue que pocos profesionales de la computación pudieron desarrollar técnicas de gráficos por computadora.

Gracias a estos avances en el terreno de los gráficos por computadora comenzaron a desarrollarse diferentes tipos de software enfocados a facilitar al usuario la creación y visualización de gráficos en los computadores de aquellos años.

Entre los años 1970 a 1980 los gráficos por computadora evolucionaron hasta llegar al mundo del entretenimiento, gracias a consolas de videojuegos como el *Magabox Odisey* o los primeros arcades de *Atari*, además de los primeros micro ordenadores en comercializarse por esos años de las marcas *Comodore* y la *Sinclair*, fue como se las capacidades gráficas de las computadoras se definieron por representaciones planas de Píxeles (acrónimo del inglés *picture element*, ‘elemento de imagen’, es la menor unidad homogénea en color que forma parte de una imagen digital) y de acuerdo a las capacidades de procesamiento de la época pudieron programarse para tener comportamientos muy variados en planos bidimensionales estos recursos fueron muy accesibles para quien tuviera en su poder algunas de estas *máquinas* ya sea por motivos profesionales o recreación.

A comienzos de 1980 las capacidades de procesamiento de datos de los computadores crecieron enormemente gracias al interés que existía por la naciente industria de la computación, esto permitió ir un paso más y trabajar con modelos en tres dimensiones de profundidad popularmente conocidos como gráficos en 3D, debido a que se programaban en planos tridimensionales, durante los comienzos de esta tecnología el público en general, comenzó a tomar conciencia de estas disciplinas como la simulación de mundos, fantasía y efectos especiales, con introducción de películas como TRON de esta forma las personas no especializadas pudieron observar Gráficos por computadora y 3D gracias a el aspecto técnico de la industria informática de esos años.

Durante la década de 1990 a medida que ingresó nuevo hardware en el mercado, fueron apareciendo gran variedad de aplicaciones para los sistemas operativo existentes y la compatibilidad para este tipo de gráficos se masifico tanto para aplicaciones profesionales, aficionados y la industria militar.

En este nuevo milenio, Microsoft lanzó una nueva gama de dispositivos con un enfoque aún vigente en la actualidad y acuñó el término Microsoft Tablet PC para las tabletas construidas según la especificación de Microsoft. Los Tablet PC, basados en la arquitectura de CPU x86, son PC completamente funcionales que admiten pantalla táctil, en lugar de una pantalla, un mouse y un teclado tradicionales, este sumado a los nuevos dispositivos de inicios del milenio permitió que los gráficos en 3D fueran ahora portables para dispositivos tanto profesionales como de entretenimiento, durante este último siglo lo que ha significado un gran avance tanto en la cantidad de polígonos como en la capacidad de verse en dispositivos cada vez más pequeño ha sido un significativo avance en una industria como poco más de 50 años. [2]

Con estos avances en la informática en las décadas pasadas la informática estaba por dar un nuevo salto y un enfoque en donde todo estaría más conectado, con la aparición del internet y por consecuente las aplicaciones web.

Una aplicación web es un programa informático que por lo general para usarse exclusivamente dentro de un navegador web tal como Google Chrome, Mozilla Firefox, entre otros, accediendo a un servidor web a través de internet o de una intranet. Las aplicaciones web son convenientes y ampliamente utilizadas en la industria del desarrollo de software debido a la facilidad para actualizar y mantenerlas, sin tener que distribuir e instalar software a miles de usuarios.

En la actualidad la tecnología en la cual se sustenta toda la web está muy estandarizada, básicamente los 3 pilares de cualquier cosa que funcione en

internet es HTML, JavaScript y CSS y la organización que estudia y publica los estándares es el World Wide Web Consortium o w3.org.

La ruta por defecto y los componentes que sigue posee cualquier aplicación web que permiten la comunicación con el usuario y/o otras aplicaciones generalmente sigue el mismo patrón y están sostenidas por los mismos principios en todo el planeta con muy pocas variaciones [16]:

- **Conexión a Internet:** Permite enviar y recibir datos en la red. Es básicamente como las calles que conectan las ciudades.
- **TCP/IP:** El Protocolo de Control de Transmisión y el Protocolo de Internet son protocolos de comunicación que definen cómo deben viajar los datos a través de Internet. Es como los mecanismos de transporte que te permiten hacer un pedido, ir a la tienda y comprar tus productos.
- **DNS:** El Sistema de Nombres de Dominio es como una libreta de direcciones para los sitios web. Cuando escribes una dirección web en tu navegador, éste mira el DNS para encontrar la dirección IP del sitio web antes de poder recuperarlo. El navegador necesita saber en qué servidor se encuentra el sitio web para poder enviar mensajes HTTP al lugar correcto. Es como buscar la dirección de la tienda para poder acceder a ella.
- **HTTP:** El Protocolo de Transferencia de Hipertexto es un protocolo de aplicación que define un lenguaje para que los clientes y los servidores hablen entre sí.
- **Archivos de componentes:** Un sitio web se compone de muchos archivos diferentes, que son como las diferentes partes de los productos que compras en la tienda. Estos archivos son de dos tipos principales:
 - **Archivos de código:** Los sitios web se construyen principalmente a partir de HTML, CSS y JavaScript.
 - **Activos:** Se trata de un nombre colectivo para todas las demás cosas que componen un sitio web, como imágenes, música, vídeo, documentos de Word y PDF.

Esta clase de aplicaciones permiten realizar múltiples tareas como crear documentos, editar fotos y escuchar música, sin tener que instalar complejos programas; además de poder implementar y ejecutar módulos que permitan administrar contenido de muy variado tipo lo que permite la interacción entre

un usuario de la aplicación con un sistema en línea y logrando observar en tiempo real cualquier tipo de transacción que se tenga entre el sistema, y si es el caso, una base de datos.

Estas son las principales ventajas y características [14]:

- **Ahorrar tiempo:** Se pueden realizar tareas, trabajar vía online, sin necesidad de descargar programas ni instalar aplicaciones adicionales en la computadora personal.
- **Compatibilidad multiplataforma:** Sólo es necesario tener un navegador web para el uso de estas aplicaciones. No existe problema alguno con el sistema operativo utilizado como Windows, Linux, IOS, entre otros.
- **No ocupa espacio en el disco duro:** Para esta clase de aplicaciones no existe la necesidad de instalación.
- **Actualizaciones:** El software es gestionado por el propio desarrollador, por lo tanto, al momento de uso y conexión a la aplicación, se asegura que se está usando la última versión sin necesidad de llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza que va a iniciar nuevas descargas y procedimientos de instalación (algunas veces imposible cuando se está trabajando dentro de grandes organizaciones).
- **Bajo consumo de recursos** debido a que la aplicación no se ejecuta directamente en el ordenador del usuario.
- **Portabilidad:** Es una gran característica y ventaja debido a que puede ser usada desde cualquier ordenador y desde cualquier lugar, siempre y cuando se cuente con un navegador web y acceso a internet. Muchas aplicaciones de hoy en día están diseñadas y desarrolladas incluso para ser usadas desde dispositivos móviles.
- **Precio:** Las aplicaciones web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional.
- **Información y datos online:** Esto permite que el usuario acceda a la aplicación sin importar su ubicación dado que cuenta con una información centralizada almacenada en bases de datos remotas.
- **Múltiples usuarios concurrentes:** Puede ser usada por varios usuarios al mismo tiempo, pueden ver e incluso editar el mismo documento de manera conjunta.

Usos comunes de las aplicaciones web [15]. existen numerosos usos tanto para los visitantes de las aplicaciones como para el mismo desarrollador y

creador tales como:

- Permitir a los usuarios localizar información de forma rápida y sencilla en un sitio Web en el que se almacena gran cantidad de contenido.

Este tipo de aplicación web ofrece a los visitantes la posibilidad de buscar contenido, organizarlo y navegar por él de la manera que estimen oportuna. Entre los ejemplos se incluyen intranets de las empresas, Microsoft MSDN y Amazon.

- Recoger, guardar y analizar datos suministrados por los visitantes de los sitios.

En el pasado, los datos introducidos en los formularios HTML se enviaban como mensajes de correo electrónico a los empleados o a aplicaciones CGI para su procesamiento. Una aplicación Web permite guardar datos de formularios directamente en una base de datos, además de extraer datos y crear informes basados en la Web para su análisis. Ejemplos de ello son las páginas de los bancos en línea, las páginas de tiendas en línea, las encuestas y los formularios con datos suministrados por el usuario.

- Actualizar sitios Web cuyo contenido cambia constantemente.

Una aplicación Web evita al diseñador Web tener que actualizar continuamente el código HTML del sitio. Los proveedores de contenido, como los editores de noticias, proporcionan el contenido a la aplicación Web y ésta actualiza el sitio automáticamente. Entre los ejemplos, se incluyen The Economist y la CNN.

Con la popularización de la web recientemente en las últimas dos décadas fue necesario dotar a los desarrolladores de herramientas que permitieran realizar su trabajo de una forma más organizada y rápida. La respuesta a esta necesidad fueron los marcos de trabajo para web o *frameworks* web como popularmente se conocen [16].

Un marco de trabajo para aplicaciones web es un tipo de marco de trabajo, o base, diseñado específicamente para ayudar a los desarrolladores a crear aplicaciones web. Estos marcos suelen proporcionar funcionalidades básicas comunes a la mayoría de las aplicaciones web, como la gestión de sesiones de usuario, la persistencia de datos y los sistemas de plantillas. Utilizando un marco de trabajo adecuado, un desarrollador puede ahorrar una cantidad significativa de tiempo en la construcción de un sitio web.

Por ejemplo, un marco de aplicación web puede proporcionar la gestión de la

sesión del usuario, el almacenamiento de datos y un sistema de plantillas. Un marco de trabajo para aplicaciones de escritorio puede proporcionar la funcionalidad de la interfaz de usuario y los widgets (elementos comunes de la interfaz gráfica de usuario). Casi todos los frameworks controlan al menos algún aspecto del flujo de ejecución de una aplicación.

Obviamente, cada aplicación web tiene sus propias necesidades en cuanto a las estructuras de datos exactas que deben persistir. Un marco de trabajo puede ayudar a la persistencia de datos con una variedad de características como, por ejemplo:

- Una API consistente para acceder a múltiples sistemas de almacenamiento de datos.
- Almacenamiento y recuperación automáticos o simplificados de los objetos de datos, como con ORM.
- Estructura central de objetos de negocio, de modo que todos los datos, incluso en varios sistemas, tengan una interfaz básica coherente.
- Mejoras en el rendimiento, como el almacenamiento en caché por encima de la capa de la base de datos.
- Comprobación de la integridad de los datos, como la validación de las relaciones o la confirmación de los campos obligatorios.
- Generador de consultas SQL.

Con este enfoque surgieron una multitud enorme de *frameworks* que se enfocaban a solucionar distintos escenarios, pero para este proyecto el *framework* seleccionado fue angular el cual por sus características es el idóneo para resolver este problema.

Angular es una plataforma y un marco de trabajo para construir aplicaciones de cliente de una sola página utilizando HTML y TypeScript. Angular está escrito en TypeScript. Implementa la funcionalidad principal y opcional como un conjunto de bibliotecas de TypeScript que importas en tus aplicaciones.

La arquitectura de una aplicación Angular se basa en ciertos conceptos fundamentales. Los bloques de construcción básicos del marco de trabajo de Angular son componentes de Angular que se organizan en NgModules. Los NgModules reúnen código relacionado en conjuntos funcionales; una aplicación Angular se define por un conjunto de NgModules. Una aplicación siempre tiene al menos un módulo raíz que permite el arranque, y normalmente tiene muchos más módulos funcionales.

- Los componentes definen las vistas, que son conjuntos de elementos de pantalla entre los que Angular puede elegir y modificar según la lógica y los datos de su programa
- Los componentes utilizan servicios, que proporcionan una funcionalidad específica no relacionada directamente con las vistas. Los proveedores de servicios pueden ser inyectados en los componentes como dependencias, haciendo que tu código sea modular, reutilizable y eficiente.
- Los componentes definen las vistas, que son conjuntos de elementos de pantalla entre los que Angular puede elegir y modificar según la lógica y los datos de su programa.
- Los componentes utilizan servicios, que proporcionan una funcionalidad específica no relacionada directamente con las vistas. Los proveedores de servicios pueden ser inyectados en los componentes como dependencias, haciendo que tu código sea modular, reutilizable y eficiente.

Los módulos, componentes y servicios son clases que utilizan decoradores. Estos decoradores marcan su tipo y proporcionan metadatos que indican a Angular cómo utilizarlos.

- Los metadatos de una clase de componente la asocian con una plantilla que define una vista. Una plantilla combina HTML ordinario con directivas de Angular y marcas de enlace que permiten a Angular modificar el HTML antes de renderizarlo para su visualización.
- Los metadatos de una clase de servicio proporcionan la información que Angular necesita para ponerla a disposición de los componentes a través de la inyección de dependencias (DI).

Los componentes de una aplicación suelen definir muchas vistas, ordenadas jerárquicamente. Angular proporciona el servicio Router para ayudarte a definir las rutas de navegación entre las vistas. El router proporciona sofisticadas capacidades de navegación en el navegador.

CAPITULO 3

La metodología empleada fue *Scrum* debido a que permite probar los productos sin tener que pasar por todo el ciclo de producción; Aporta un enfoque democrático al trabajo

En este marco de trabajo pueden emplearse varios procesos, técnicas y métodos. Scrum envuelve las prácticas existentes o las hace innecesarias. Scrum hace visible la eficacia relativa de las técnicas actuales de gestión, entorno y trabajo, de modo que se puedan realizar mejoras

Fases de Scrum: los eventos que componen la metodología Scrum

Anteriormente, te conté sobre los eventos de Scrum. En esta sección vamos a profundizar en cada una de las fases para que tengas claro cómo funcionan las etapas de Scrum.

1. Planeación del Sprint/Sprint Planning

Todos los involucrados en el equipo se reúnen para planificar el Sprint. Durante este evento se decide qué requerimientos o tareas se le asignará a cada uno de los elementos del equipo. Cada integrante deberá asignar el tiempo que crea prudente para llevar a cabo sus requerimientos. De esta manera se define el tiempo de duración del Sprint.

2. Reunión de equipo de Scrum/Scrum team meeting

A estas reuniones se les deberían dedicar máximo 15 minutos diarios, y deberían ser siempre en el mismo horario y lugar. En ellas, cada miembro del equipo deberá responder tres simples preguntas:

- ¿Qué hiciste ayer?
- ¿Qué tienes planeado hacer hoy?
- ¿Qué obstáculos encontraste en el camino?

Estas reuniones sirven para que todos los miembros del equipo se apoyen entre ellos. Si alguno de ellos tiene algún inconveniente que obligue a extender el encuentro, este debe tratarse más a fondo en una reunión enfocada en buscar la mejor solución para ello.

3. Refinamiento del Backlog/Backlog Refinement

El Product Owner revisa cada uno de los elementos dentro del Product Backlog con el fin de esclarecer cualquier duda que pueda surgir por parte del equipo de desarrolladores. También sirve para volver a estimar el tiempo y esfuerzo dedicado a cada uno de los requerimientos.

4. Revisión del Sprint/Sprint Review

Los miembros del equipo y los clientes se reúnen para mostrar el trabajo de desarrollo de software que se ha completado. Se hace una demostración de todos los requerimientos finalizados dentro del Sprint. En este punto no es necesario que todos los miembros del equipo hablen, pueden simplemente estar presentes, pero la presentación está a cargo del Scrum Master y el Product Owner.

5. Retrospectiva del Sprint/Retrospective

En este evento el Product Owner se reúne con todo su equipo de trabajo y su Scrum Master para hablar sobre lo ocurrido durante el Sprint. Los puntos principales a tratar en esta reunión son:

- Qué se hizo mal durante el Sprint para poder mejorar el próximo.
- Qué se hizo bien para seguir en la misma senda del éxito.
- Qué inconvenientes se encontraron y no permitieron poder avanzar como se tenía planificado.

3.1 Scrum como metodología ágil

Scrum es una Metodología Ágil de Gestión de Proyectos que se basa en la adaptación continua a las circunstancias evolutivas del Proyecto apoyándose en iteraciones cortas conocidas como *Sprints* [11] a través del siguiente ciclo:

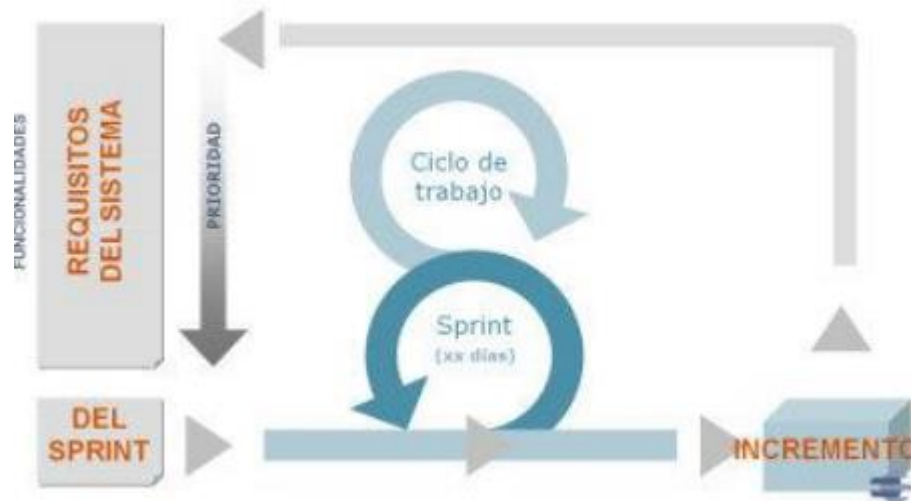


Figura 7. Ciclo de trabajo de scrum.

Su funcionamiento está marcado por sus roles/responsabilidades y sus reuniones:

Roles/Responsabilidades:

Comprometidos:

- Propietario del Producto (*Product Owner*): responsable de lograr el mayor valor del producto desarrollado.
- Equipo de Desarrollo (*Team*): responsable de desarrollar el producto.

Implicados:

- Otros interesados (*Stakeholders*): intervienen de manera indirecta o tienen intereses en el proyecto.
- *Scrum Master*: responsable del funcionamiento de *Scrum*

Reuniones:

- Planificación del *Sprint*: reunión donde el Equipo define la Pila del Sprint a partir de la explicación de la Pila del Producto por parte del Propietario del Producto.
- Seguimiento del *Sprint*: reunión rápida diaria donde el Equipo revisa las tareas de la pila del sprint que ha realizado el día anterior, las que hará en el día y las posibles necesidades e impedimentos que tenga para continuar el trabajo.
- Revisión del *Sprint*: reunión informativa donde el Equipo presenta al Propietario del Producto el Incremento.
- Retrospectiva: reunión de, “mejora continua” donde el Equipo analizará los diferentes problemas encontrados y los aspectos mejorables de la aplicación de *Scrum* en el Proyecto

Equipo de Desarrollo

Para *Scrum* el equipo es el grupo multidisciplinar y autoorganizado cuyo trabajo se realiza de forma cohesionada en aras de completar las tareas de los diferentes *sprints* que se generan a partir de la pila del producto, en otras metodologías más clásicas podríamos determinar al equipo de desarrollo como población. Frente al concepto de grupo de trabajo, el equipo, respondiendo en su conjunto y sin la necesidad de un gestor que marque su ritmo, trabaja de manera colaborativa en la búsqueda de proporcionar el mayor valor posible al cliente.

En un entorno distribuido los miembros del equipo deben mostrar un mayor compromiso y una mayor disciplina en la gestión de sus quehaceres personales, así como con el uso de las herramientas que facilitan la interacción ya que, en tales ambientes, es mucho más fácil que estos se vean comprometidos.

Esto implica, por tanto, una mayor responsabilidad de interacción por parte del equipo, que no deberá dejar caer sobre el *Scrum Manager* la responsabilidad de perseguir, a través de los diferentes mecanismos y herramientas de comunicación establecidos, a los miembros del equipo en busca de, entre otros, la visibilidad de los avances y de los impedimentos que requiere esta metodología.

En definitiva, al trabajar en un *Scrum* Distribuido los miembros del equipo deberían exigirse una mayor disciplina personal, viendo las herramientas como extensiones o ayudas que, en estos contextos, facilitarán la interacción y la autoorganización del equipo.

3.2 Determinación del tipo de estudio

En las fuentes bibliográficas hay varias definiciones exactas del término investigación y desarrollo, si bien están marcadas por las mismas características. La investigación y desarrollo abarca todas las actividades metódicas y sistemáticas sobre una base de métodos científicos con el cometido de adquirir más conocimientos reales. Este término es usualmente utilizado para la descripción de un departamento específico en las empresas que buscan innovaciones sea en sus servicios o productos. [12]

El tipo de estudio para este proyecto es una investigación aplicada ya que busca la generación de conocimiento con aplicación directa a los problemas de la sociedad o el sector productivo. Esta se basa fundamentalmente en los hallazgos tecnológicos de la investigación básica, ocupándose del proceso de enlace entre la teoría y el producto. El presente ensayo presenta una visión sobre los pasos a seguir en el desarrollo de investigación aplicada, la importancia de la colaboración entre la universidad y la industria en el proceso de transferencia de tecnología, así como los aspectos relacionados a la protección de la propiedad intelectual durante este proceso.[13]

Selección, diseño y prueba del instrumento de recolección de la información

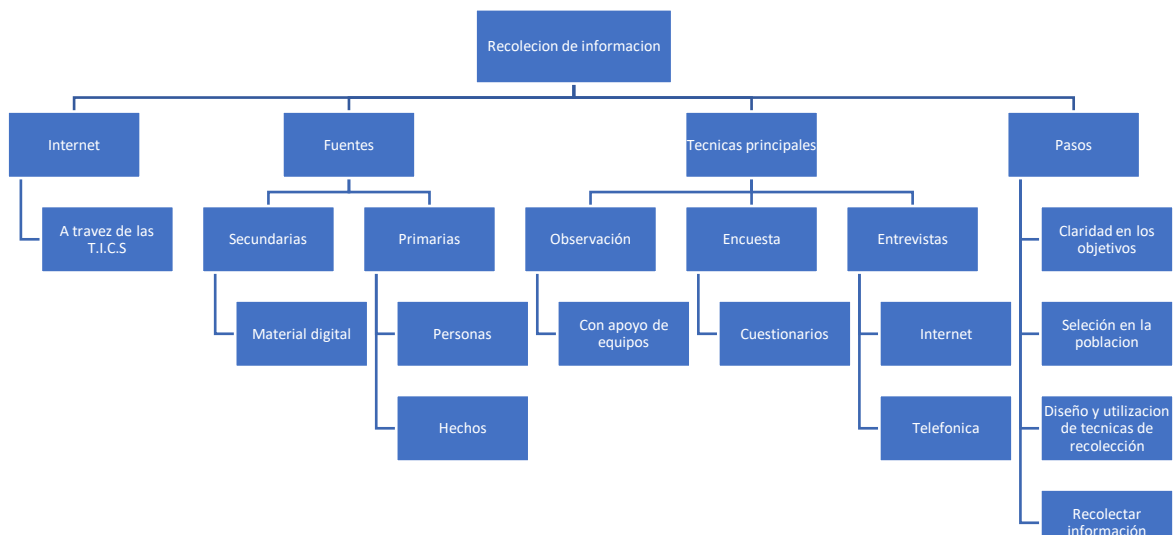


Figura 8. Recolección de información

Instrumento para la recolección de información: Azure DevOps



Figura 9. Azure ® Isologo

Azure DevOps Server es un producto de Microsoft que proporciona control de versiones (TFVC), informes, gestión de requisitos, gestión de proyectos (tanto para el desarrollo de software ágil como para los equipos en cascada), capacidades de gestión de versiones, pruebas y compilaciones automatizadas. Cubre todo el ciclo de vida de la aplicación y habilita las capacidades de *DevOps*. *Azure DevOps* se puede utilizar como *back-end* para numerosos entornos de desarrollo integrados (IDE), pero está diseñado para *Microsoft Visual Studio* y *Eclipse* en todas las plataformas.

Team Foundation Version Control:

TFVC es un sistema de control de versiones centralizado que permite a los equipos almacenar cualquier tipo de artefacto dentro de su repositorio. TFVC admite dos tipos diferentes de espacios de trabajo cuando se trabaja con herramientas cliente: espacios de trabajo del servidor y espacios de trabajo locales. Los espacios de trabajo del servidor permiten a los desarrolladores bloquear archivos para retirarlos y notificar a otros desarrolladores que se están editando archivos. Una queja frecuente de este modelo es que los archivos de la máquina de desarrollo están marcados como de solo lectura. También requiere que los desarrolladores "se desconecten" cuando no se pueda contactar con el servidor. Los espacios de trabajo locales se diseñaron para evitar estos problemas. En un escenario de espacio de trabajo local, los archivos no son de solo lectura y no es necesario desprotegerlos antes de trabajar en ellos. Siempre que los archivos estén en la máquina local del desarrollador, no importa si el servidor está conectado o no. Los conflictos se resuelven en el momento del *check-in*.

Azure DevOps también admite una función de análisis de código que, cuando se usa de forma independiente, se conoce como *FxCop*. La inclusión en *Azure DevOps* significa que el análisis se puede ejecutar con el código registrado en el servidor y durante las compilaciones automatizadas.

Plan de recolección de la información para el trabajo de campo

¿Cuáles son las fuentes de donde se obtendrán los datos?

- Directo con desarrolladores (Entrevistas, cuestionarios)
- *Azure DevOps*
 - Desarrolladores
 - Informes de código
 - Informes de desarrollo
 - Informes de scrum
 - Retrospectivas
 - Código
 - Gráficas de desempeño
 - Gráficas de performance de software
 - Reporte de errores
 - Planes de pruebas

¿En dónde se localizan tales fuentes?

- *Azure DevOps*
 - Desarrolladores
 - Informes de código
 - Informes de desarrollo
 - Informes de scrum
 - Código
 - Gráficas de desempeño
 - Gráficas de performance de software

¿A través de qué medio o método se recolectaron los datos?

Entrevistas directas con desarrolladores y con los informes de *Azure DevOps*.

¿De qué forma se preparó para analizar y responder al planteamiento del problema?

1. Analizar las opiniones directas de los desarrolladores con el viejo *framework* y compararlas con la retroalimentación actual del nuevo *framework*.
2. Revisar todos los informes y gráficas de desempeños tanto de desarrolladores como del código además de las realizadas por los planes de pruebas de *Azure DevOps* con el viejo *framework* y el nuevo realizado por este proyecto para comprobar que los defectos y las opiniones de los usuarios se volvieron favorables con este proyecto.

Plan de procesamiento y análisis de información

Paso	Fuentes	Ejemplos
Recuperando documentos existentes y datos	<p>Documentos formales como planes de la empresa e implementaciones.</p> <ul style="list-style-type: none"> • Estadísticas oficiales • Datos de seguimiento del programa • Registros del programa 	Revisión de la planificación del programa documentos, reuniones e informes de progreso.
Recopilación de datos de individuos o grupos	<p>Entrevistas a programadores y colaboradores clave. Cuestionarios o encuestas, correo electrónico, web, datos móviles y de Azure DevOps.</p>	<p>Entrevistas con el dueño del producto y los programadores involucrados en el proyecto.</p> <p>Grupos de discusión para analizar los datos de las fuentes mencionadas.</p>
Observación	Notas, Grabaciones imágenes y vídeos de las reuniones o comportamiento.	Observaciones del programa actividades e interacciones con los desarrolladores.

Tabla 1. Plan de procesamiento y análisis de información

3.3 Representación gráfica de los avances

Se tiene considera utilizar estos gráficos para la representación de los tiempos y tareas del desarrollo.

En el desarrollo del cronograma de actividades se consideraron parámetros para el desarrollo del proyecto diferentes variables, tales como:

- Tiempo de disponibilidad del equipo de desarrollo para el proyecto
- Cantidad de desarrolladores
- Experiencia con .Net Core
- Curva de aprendizaje para el Framework 3.1 de Net Core

Dario Cubillos Martinez

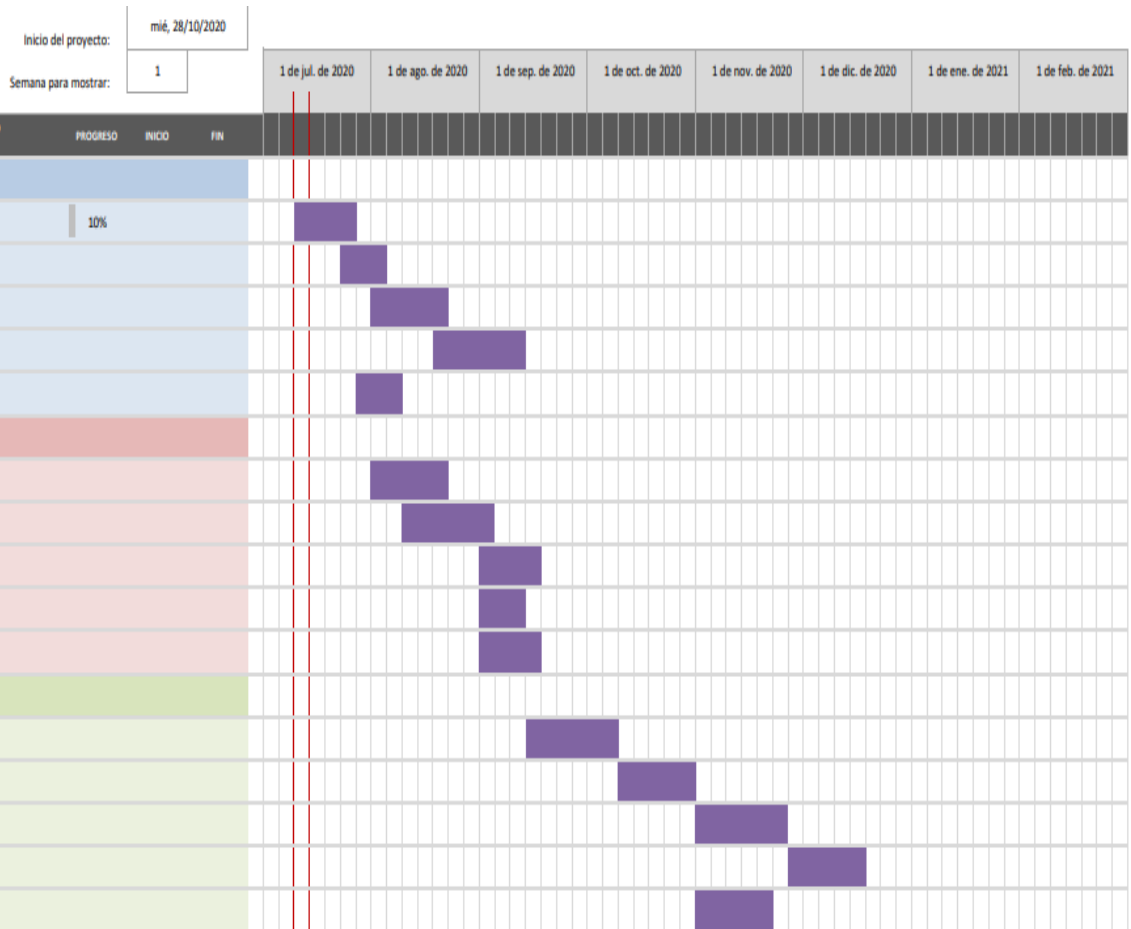


Tabla 2. Cronograma de actividades

CAPITULO 4

4.1 Descripción de los resultados del trabajo

Los resultados fueron positivos permitiendo una excelente compatibilidad, un escalado adecuado en la representación de los dibujos, la posibilidad de implementar nuevas características fácilmente y el tiempo de respuesta fue disminuido significativamente en el software, debido a estos resultados el desarrollo se concluyó satisfactoriamente.

La siguiente tabla muestra las problemáticas, así como el periodo empleado para solventarlas, con una descripción de la solución realizada.

Descripción del problema	Fecha inicio	Fecha solución	Solución
Compatibilidad con software desarrollado internamente y externamente	07/01/2021	17/02/2021	Con las actualizaciones en el código se pudieron arreglar el problema en casi todos los casos, con posibles pequeños cambios en el futuro.
Escalado no proporcional entre aplicaciones	19/01/2021	17/02/2021	Con los cambios recientes en las librerías del proyecto, este problema está completamente solucionado.
Limitaciones para implementar nuevas características	27/01/2021	17/11/2021	Las nuevas características ya son fáciles de integrar gracias a un enfoque en microservicios que hacen fácil tanto el despliegue como la programación de la aplicación.
Tiempo de respuesta largo por parte del equipo de soporte en hoops a CEA Systems	07/01/2021	22/11/2021	Gracias a la experiencia y a los cambios hechos en todo el proyecto el equipo es mucho más preciso en resolver problemas o identificar errores en las librerías de terceros.

Tabla 3. Plan de procesamiento y análisis de información

4.2 Página de inicio de sesión del proyecto

A continuación, se pueden observar pantallas del (de la solución planteada) framework desarrollado.

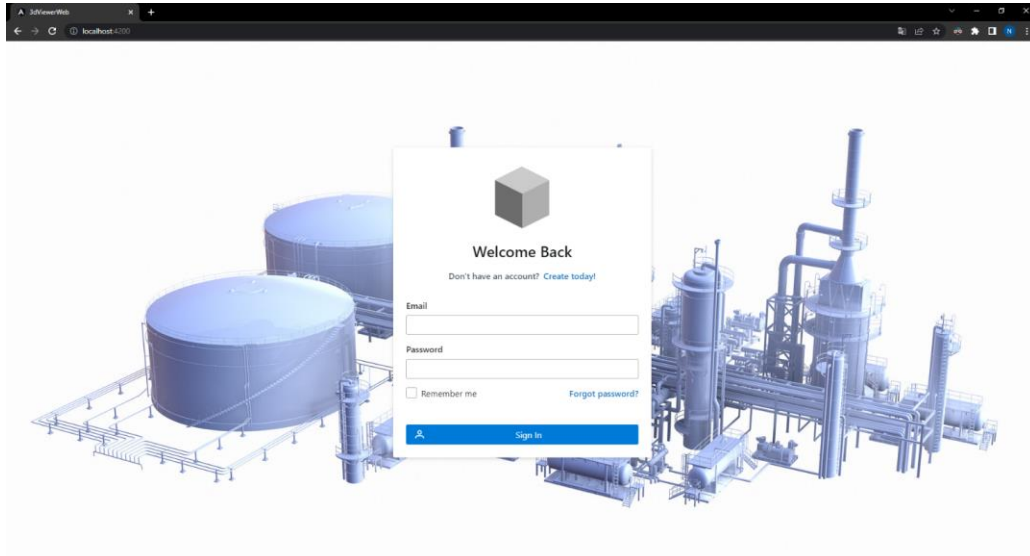


Figura 10. Inicio de sesión del visor

4.3 Visor 3D del proyecto vista simple del modelo



Figura 11. Vista por defecto de un modelo 3D en el web framework

4.4 Visor 3D del proyecto con propiedades del modelo

Se puede observar el panel de información cuando una parte de la pieza es seleccionada lo que nos permite obtener datos de las partes específicas del modelo.

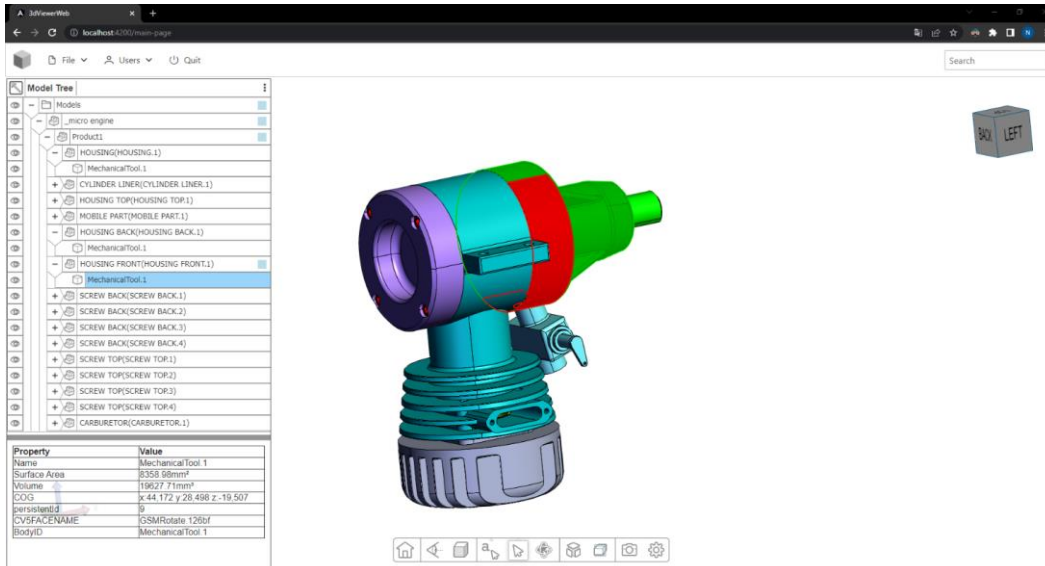


Figura 12. Selección y panel informativo de una parte específica

4.5 Visor 3D del proyecto barra de herramientas

El web framework tiene una barra de herramientas donde el usuario puede cambiar entre los modelos disponibles, observe la siguiente imagen.



Figura 13 Barra de herramientas del web framework

4.6 Conclusiones

- Gracias a una planificación adecuada se implementaron los últimos paradigmas de las actuales tecnologías web y de procesamiento de gráficos además el enfoque fue en gran medida apuntando a las necesidades que requiere la industria para la administración de recursos gráficos e información.
- Este proyecto logró un enfoque en la gestión de los datos de la empresa y puso los activos intelectuales en un ciclo de conocimiento, es decir, almacena, recopila, reutiliza, comparte y actualiza valiosa experiencia, sabiduría y conocimiento de las personas que pertenecen a un equipo de trabajo de manera eficiente y de forma visible lo que con este proyecto potencializa estos aspectos y quita las deficiencias anteriores pudiendo concluirlo en sus objetivos como todo un éxito.

4.7 Recomendaciones

- El proyecto con el paso del tiempo como todo software debe dársele soporte adecuado, encontrando según el contexto, áreas de oportunidad donde mejorar ya que la informática tiende a ser una disciplina con cambios e innovación muy rápidas.
- Las actualizaciones son una parte importante del soporte y debe hacerse un especial énfasis en actualizar aquellas dependencias del proyecto que puedan ser una brecha de seguridad en el futuro, esto como consecuencia de la constante innovación en la disciplina de la informática. La seguridad en estas aplicaciones debe ser una gran prioridad.

BIBLIOGRAFÍA

- [1] R. Yoshida, T. Murao, y T. Miyazawa, «3D web environment for knowledge management», *Future Generation Computer Systems*, vol. 17, n.º 1, pp. 73-78, sep. 2000, doi: 10.1016/S0167-739X(99)00099-0.
- [2] Peddie, *The History of Visual Magic in Computers*. London: Springer London, 2013, pp 199-200.
- [3] F. Prandi, F. Devigili, M. Soave, U. Di Staso, y R. De Amicis, «3D web visualization of huge CityGML models», *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XL-3/W3, pp. 601-605, ago. 2015.
- [4] S. Ortega, J. M. Santana, J. Wendel, A. Trujillo, y S. M. Murshed, «Generating 3D City Models from Open LiDAR Point Clouds: Advancing Towards Smart City Applications», en *Open Source Geospatial Science for Urban Studies*, A. Mobasher, Ed. Cham: Springer International Publishing, 2021, pp. 97-116.
- [5] H. Rodrigues, F. Gaspar, P. Fernandes, y A. Mateus, Eds., *Sustainability and Automation in Smart Constructions: Proceedings of the International Conference on Automation Innovation in Construction (CIAC-2019), Leiria, Portugal*. Cham: Springer International Publishing, 2021.
- [6] R. Yoshida, T. Murao, y T. Miyazawa, «3D web environment for knowledge management», *Future Generation Computer Systems*, vol. 17, n.º 1, pp. 73-78, sep. 2000.
- [7] N. Baek, «An emulation scheme for OpenGL SC 2.0 over OpenGL», *J Supercomput*, vol. 76, n.º 10, pp. 7951-7960, oct. 2020.
- [8] G. S. Hura, A. K. Singh, y L. Siong Hoe, Eds., *Advances in Communication and Computational Technology: Select Proceedings of ICACCT 2019*, vol. 668. Singapore: Springer Singapore, 2021, pp 1153.
- [9] F. T. S. Yu, «Holography: Origin, Rediscovery, Development and Beyond», p. 12, 2015.
- [10] 15 Ene 2010. Web application framework. Disponible: https://web.archive.org/web/20150723163302/http://docforge.com/wiki/Web_application_framework. [Consultado Jun. 22, 2021].
- [11] Scrum.org. 2021. Home. [online] Available at: <<https://www.scrum.org/>> [Accessed 30 November 2021].
- [12] R. Ragno, «www.3d-qsar.com: a web portal that brings 3-D QSAR to all electronic devices—the Py-CoMFA web application as tool to build models from pre-aligned datasets», *J Comput Aided Mol Des*, vol. 33, n.º 9, pp. 855-864, sep. 2019, doi: 10.1007/s10822-019-00231-x.
- [13] José Lozada, “CienciAmérica: Revista de divulgación científica de la Universidad Tecnológica Indoamérica,” *Abbrev. Title of Magazine (in italics)*, vol. 3, no. 1, págs. 47-50, *Abbrev. 01*, 2014.

- [14] Lujan Mora, Sergio. Programación de aplicaciones web: Historia, principios básicos y clientes web, Editorial club universitario, pags 53-60, ISBN: 84-8454-206-8.
- [15] ADOBE COMMUNITY HELP. Aspectos básicos de las aplicaciones web. [En línea]. Disponible en: <https://helpx.adobe.com/es/dreamweaver/using/web-applications.html> Revisado 2022.
- [16] Web framework, Doc forge software development resources [En línea] disponible en: https://web.archive.org/web/20150723163302/http://docforge.com/wiki/Web_application_framework, Revisado 2022.
- [17] Introduction to Angular concepts resources, [En línea] disponible en: <https://angular.io/guide/architecture>, Revisado 2022.
- [18] "ISO/IEC 2382:2015". ISO. 3 September 2020. Retrieved 26 May 2022.
- [19] Merriam-Webster, *Merriam-Webster's Collegiate Dictionary*, Merriam-Webster.
- [20] Tatomir, A.; et al. (2018). "Conceptual model development using a generic Features, Events, and Processes (FEP) database for assessing the potential impact of hydraulic fracturing on groundwater aquifers". *Advances in Geosciences*. 45: 185–192. Bibcode:2018AdG....45..185T. doi:10.5194/adgeo-45-185-2018.

ANEXOS

En la imagen izquierda podemos ver el problema de compatibilidad en el dibujo debido a que hay algunos elementos que no se pueden visualizar, esto si lo comparamos con la imagen de la derecha que tiene todos los elementos del dibujo correctamente cargados.

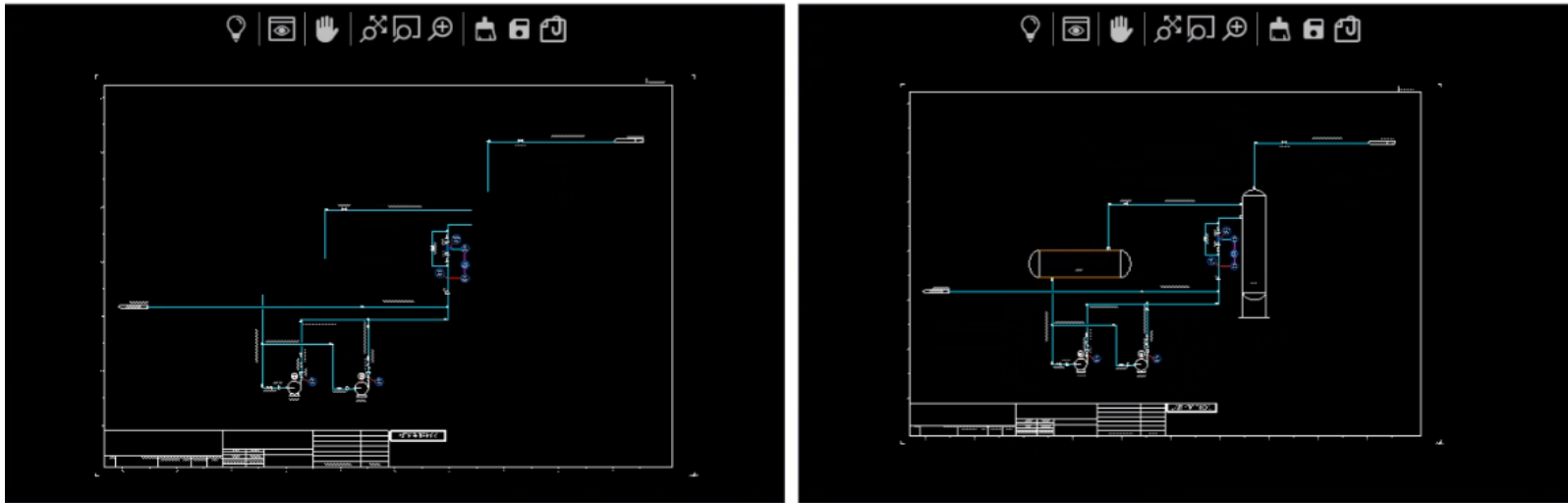


Figura 14 Antes y Después de los problemas de compatibilidad (Izquierda y derecha respectivamente)