



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA SUPERIOR
DE INGENIERÍA GEODÉSICA
CARTOGRÁFICA Y TOPOGRÁFICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA
CARTOGRÁFICA Y TOPOGRÁFICA

MODELADO Y BASE DE DATOS TRIDIMENSIONAL PARA EL CONTROL DE ESTOCAJE EN EL SECTOR ALIMENTARIO.

GRADO EN INGENIERÍA GEOMÁTICA Y TOPOGRAFÍA



"El presente documento ha sido realizado completamente por el firmante; no ha sido entregado como otro trabajo académico previo y todo el material tomado de otras fuentes ha sido convenientemente entrecomillado y citado su origen en el texto, así como referenciado en la bibliografía".

Firmado: "Javier Ignacio Chiva Juanes"

AUTOR: JAVIER IGNACIO CHIVA JUANES

TUTORES: RAMÓN PONS CRESPO Y ÁNGEL MARQUÉS MATEU

CURSO ACADÉMICO: 2020-2021



Agradecimientos.

A mis tutores Ramón Pons Crespo y Ángel Marqués Mateu, por tener esa paciencia casi infinita conmigo y haberme apoyado, aceptado y ayudado en todas mis ideas estrafalarias.

A los amigos de siempre, a los nuevos hechos en la universidad, gracias por todas las risas y las historias que hemos tenido y tendremos.

A mi nueva familia, Eusebio, Diego, Arturo y Natalia, gracias por apoyarme y animarme para hacer todo esto.

A mi familia de siempre, “jefa”, “tete”, “yaya” y al hermano que elegí, gracias por estar allí en las buenas y en las malas, habéis hecho que este cabezota crezca como persona y gracias por no permitir que jamás se rindiera.

Por último, a los que por desgracia ya no están aquí, espero que estéis orgullosos de mí desde allá donde os encontréis.

A todos los anteriores y a los que me dejo en el tintero, gracias por estar ahí y formar parte de mi corazón. Este trabajo os lo dedico a todos vosotros.

RESUMEN

El proyecto está dividido en dos partes bien diferenciadas entre sí.

La primera parte es la generación de un levantamiento de la nave industrial de una empresa situada en Paiporta, la cual se dedica al sector de la alimentación, utilizando el láser escáner para obtener los datos y la posterior creación de un modelo tridimensional de la nave para que se pueda utilizar como herramienta de marketing digital en su página web.

La segunda parte es la implementación de una base de datos más óptima para llevar un estocaje mucho más preciso y controlado desde cualquier lugar.

Palabras claves:

Base de datos – Modelo 3D – Estocaje – Láser escáner – Publicidad – Dashboard

RESUM

El projecte està dividit en dues parts ben diferenciades entre sí.

La primera part és la generació d'un alçament de la nau industrial d'una empresa situada a Paiporta, la qual es dedica al sector de l'alimentació, utilitzant el làser escàner per a obtindre les dades i la posterior creació d'un model tridimensional de la nau perquè es puga utilitzar com a ferrament de màrqueting digital en la seua pàgina web.

La segona part és la implementació d'una base de dades més òptima per a portar un estocatge molt més precís i controlat des de qualsevol lloc.

Paraules clau:

Base de dades – Model 3D – Estocatge – Làser escàner – Publicitat – Dashboard

ABSTRACT

The project is divided into two well differentiated parts.

The first one is the generation of a survey about the industrial warehouse for a company located in Paiporta, which belongs to the food industry, using the laser scanner to obtain the data and the subsequent creation of a three-dimensional model of the warehouse which can be used as a digital marketing tool on their website.

The second one is an implementation of an optimal database to carry on much more precise and controlled stock from anywhere.

Keywords:

Database – 3D Model - Stockage – Laser Scan – Publicising – Dashboard

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1: Fachada de la empresa a modelar y situación en Google maps..... | 10 |
| Figura 2: Imagen del láser escáner Trimble TX6..... | 11 |
| Figura 3: Fotografías del láser escáner Trimble TX6 realizando el levantamiento en campo.. | 12 |
| Figura 4: Croquis realizado en campo para la situación de los estacionamientos..... | 12 |
| Figura 5: Software utilizado. | 14 |
| Figura 6: Registro de las nubes de puntos. | 14 |
| Figura 7: Explicación de cómo realizar el registro con el Autodesk ReCap mediante puntos homólogos. | 15 |
| Figura 8: Informe de la nube de puntos resultado y croquis de las uniones..... | 15 |
| Figura 9: Registro de varias nubes de puntos. | 16 |
| Figura 10: Interior de la nube de puntos. | 16 |
| Figura 11: Nube de puntos sin limpiar. | 17 |
| Figura 12: Nube de puntos con limpieza realizada..... | 17 |
| Figura 13: Creación del proyecto arquitectónico. | 18 |
| Figura 14: Nube de puntos cargada en Revit. | 18 |
| Figura 15: Creación de los muros exteriores..... | 19 |
| Figura 16: Creación de todos los muros exteriores. | 19 |
| Figura 17: Creación de los suelos..... | 20 |
| Figura 18: Creación de puertas y ventanas..... | 20 |
| Figura 19: Creación de la planta cero de las oficinas..... | 21 |
| Figura 20: Creación de la primera y segunda planta de las oficinas. | 22 |
| Figura 21: Proceso de creación de las estanterías..... | 22 |
| Figura 22: Modelo final desde el exterior. | 23 |
| Figura 23: Plano de planta a altura 1 (altura suelo). | 23 |
| Figura 24: Vista del interior sin tejado. | 23 |
| Figura 25: Publicación en ACCA Software del modelo resultado. | 24 |
| Figura 26: Comparación de base de datos SQL (izquierda) con bases de datos NoSQL (Derecha). | 25 |
| Figura 27: Script de carga de datos. | 27 |
| Figura 28: Script de carga de datos. | 28 |
| Figura 29: Comparación entre JSON y YAML. | 30 |
| Figura 30: Visual Studio Code. | 30 |
| Figura 31: Ejemplo de código final en HTML..... | 31 |

| | |
|--|----|
| Figura 32: Código creado para envío de formulario en PHP. | 31 |
| Figura 33: Código creado para ocultar el menú en JavaScript. | 32 |
| Figura 34: Ejemplo de JSON..... | 32 |
| Figura 35: Ejemplo de código de diseño CSS..... | 32 |
| Figura 36: HTML final. Portada. | 33 |
| Figura 37: HTML final. Dónde encontrarnos..... | 33 |
| Figura 38: HTML final. Conócenos..... | 34 |
| Figura 39: HTML final. Ejemplo de producto de la base de datos. | 34 |
| Figura 40: Carga de datos desde el servidor a Power BI. | 35 |
| Figura 41: Administración de relaciones manual. | 36 |
| Figura 42: Inserción de forma visual los datos..... | 36 |
| Figura 43: Dashboard final. Productos. | 37 |
| Figura 44: Dashboard final. Anual..... | 37 |
| Figura 45: Dashboard final. Clientes. | 38 |
| Figura 46: Dashboard final. Control anual del stock | 38 |
| Figura 47: Raspberry Pi 4..... | 39 |
| Figura 48: Webcam-Cámara IP. | 39 |
| Figura 49: Captura a tiempo real. | 40 |
| Figura 50: Sensor VL53L0X. | 41 |

ÍNDICE

| | |
|---|----|
| Agradecimientos..... | 3 |
| Resumen..... | 4 |
| Resum..... | 4 |
| Abstract..... | 4 |
| Índice de figuras..... | 5 |
| 3. Introducción..... | 9 |
| 3.1. Antecedentes y justificación del proyecto..... | 9 |
| 4. Objetivos..... | 9 |
| 5. Situación..... | 10 |
| 6. Instrumentación y materiales..... | 10 |
| 6.1. Láser escáner..... | 10 |
| 6.2. Ordenador..... | 11 |
| 6.2.1. Hardware:..... | 11 |
| 7. Toma de datos en campo..... | 11 |
| 8. Software utilizado para realizar el modelo tridimensional..... | 13 |
| 8.1. Trimble RealWorks 11.0..... | 13 |
| 8.2. Autodesk ReCap..... | 13 |
| 8.3. Autodesk Revit 2018..... | 13 |
| 9. Descripción del proceso del modelaje..... | 14 |
| 9.1. Registro de la nube de puntos..... | 14 |
| 9.2. Modelización tridimensional..... | 17 |
| 10. Publicación del modelo tridimensional..... | 24 |
| 11. Base de datos NoSQL..... | 24 |
| 11.1. Diferencias fundamentales entre bases de datos SQL..... | 25 |
| 11.2. Sistema operativo en base Linux: Ubuntu..... | 25 |
| 11.2.1. Docker..... | 25 |
| 11.2.2. Fiware..... | 26 |
| 11.2.4. Solicitudes GET/POST..... | 27 |
| 11.3. Carga de los datos de la base de datos mediante script..... | 28 |
| 12. Realización de página web..... | 29 |
| 12.1. Lenguajes utilizados..... | 29 |
| 12.1.1. HTML..... | 29 |



| | | |
|---------|--|----|
| 12.1.2. | PHP..... | 29 |
| 12.1.3. | JAVASCRIPT..... | 29 |
| 12.1.4. | YAML/JSON..... | 30 |
| 12.2. | Software utilizado..... | 30 |
| 12.2.1. | Visual Studio Code..... | 30 |
| 13. | Dashboard..... | 34 |
| 13.1. | Software utilizado..... | 35 |
| 13.1.1. | Microsoft Power Bi..... | 35 |
| 13.2. | Raspberry Pi 4..... | 39 |
| 13.2.1. | Sistema operativo en base Linux: Raspbian..... | 39 |
| 14. | Acciones de mejora u optimización..... | 40 |
| 15. | Presupuesto..... | 41 |
| 16. | Conclusiones..... | 42 |
| 17. | Bibliografía..... | 43 |
| 18. | Cartografía..... | 45 |
| 19. | Anejos..... | 46 |

3. Introducción.

3.1. Antecedentes y justificación del proyecto.

Viendo la pandemia sufrida y todas las consecuencias que ha generado durante el año pasado y que siguen presentes hoy, decidí intentar aportar algo positivo dentro de lo posible a una empresa con la cual tengo una cierta amistad. Es obvio que un gran número de empresas ha perdido clientela y por ende los beneficios que han generado han disminuido. Es por ello que en este trabajo final de grado (T.F.G.) se ha intentado dos cosas: la primera y más importante tocar todos los temas posibles impartidos en la carrera, toma de medidas en campo, postproducción, bases de datos... y la segunda parte hacer que la empresa en cuestión tenga productos “novedosos” para que los publique y llame la atención a nuevos clientes.

4. Objetivos.

Como he comentado en el anterior punto, en este T.F.G., el objetivo son dos, el primero intentar hacer un proyecto con el mayor número de campos relacionados y el segundo hacer que la empresa disponga de material nuevo para llegar a más clientes a nivel publicitario.

Los objetivos específicos son:

- Creación de un modelo tridimensional y su publicación para poder visualizarlo.
- Implantar una base de datos NoSQL, para hacer la empresa “Smart-warehouse”.
- Implantación de un dashboard para llevar un control del stock de los productos de una manera más precisa.

5. Situación.

El edificio a tratar se sitúa en el polígono industrial del municipio de Paiporta. La empresa en cuestión es José Rodrigo e hijos S.A.



Figura 1: Fachada de la empresa a modelar y situación en Google maps.

6. Instrumentación y materiales.

En este punto explicaremos el material utilizado y la metodología posterior en el tratamiento de los datos tomados en campo.

6.1. Láser escáner

Es un instrumento que se utiliza para capturar formas geométricas y colores de cualquier objeto o entorno circundante a partir de un almacenamiento masivo de datos, guardándolo en puntos, con información de coordenadas tridimensiones (X, Y, Z). Estos datos existen en forma de nubes de puntos y se obtienen midiendo distancias y ángulos utilizando un rayo láser que, en combinación con una cámara, también registra información sobre el rango visible. Estos datos se utilizarán más tarde para formar un modelo tridimensional del objeto. En otras palabras, con la ayuda de la tecnología de escáner láser, podemos convertir objetos reales en objetos virtuales y operarlos a través de software especializado.

El potencial del escáner es muy grande, puede obtener las coordenadas geográficas de todas las superficies alrededor del instrumento (dentro de un radio limitado) en unos pocos minutos, sin ningún contacto con el elemento medido.

En el caso particular nuestro, hemos utilizado para la toma de datos en campo, el láser escáner TX6 de la marca Trimble. De todos los niveles de escaneo, que dispone el láser para la adquisición de datos, utilizamos el nivel 1 (segundo nivel del escáner, con un escaneo de 34 millones de puntos y una duración de 3 minutos por escaneo) y adquisición de imágenes (toma de fotografías).



Figura 2: Imagen del láser escáner Trimble TX6.

6.2. Ordenador.

Para el tratamiento y edición de los archivos es importante disponer de un buen ordenador para poder trabajar más ágilmente, debido a la cantidad de información que disponemos.

6.2.1. Hardware:

El ordenador utilizado tiene un i5-9600k como procesador, 16 gb de memoria RAM y una NVIDIA GeForce GTX 1650 SUPER.

7. Toma de datos en campo.

Realizamos un total de 11 escaneos. Los dos primeros escaneos que realizamos fueron de la fachada y el lateral de la empresa. Los otros 9 se utilizaron para realizar el interior de la empresa. Por motivos de seguridad las oficinas y las neveras y congeladores no se realizaron por dentro.



Figura 3: Fotografías del láser escáner Trimble TX6 realizando el levantamiento en campo.

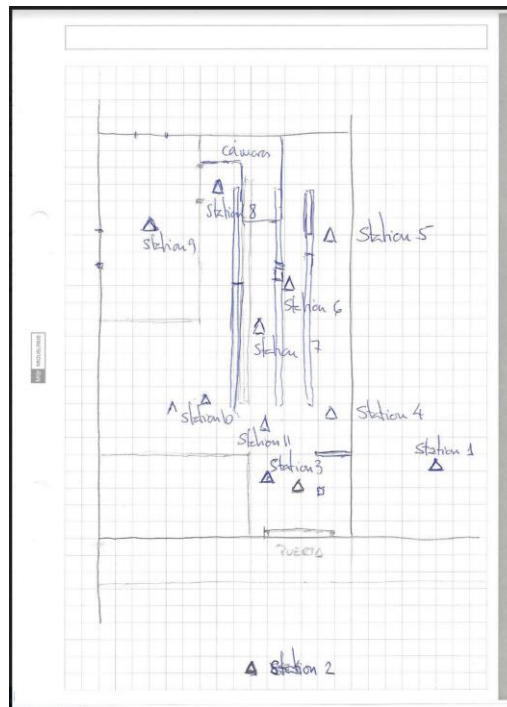


Figura 4: Croquis realizado en campo para la situación de los estacionamientos.

8. Software utilizado para realizar el modelo tridimensional.

8.1. Trimble RealWorks 11.0.

Trimble RealWorks está diseñado para procesar y analizar nubes de puntos. Este software permite registrar, analizar, modelar y crear el producto final.

Este es el primer y único de los softwares que he utilizado por obligación, ya que es necesario para realizar la conversión de los datos tomados en campo, ya que el formato con el que se guarda no es legible. El cambio de datos ha sido de .tcf a .e57.

8.2. Autodesk ReCap.

Autodesk ReCap significa "Reality Capture" y es un programa para procesar nubes de puntos reales a partir del escaneo realizado por láser escáner. Se crean miles de puntos (es decir, nubes de puntos) para cada escaneo, que puede considerarse como un modelo simplificado del proyecto escaneado.

Decidí realizar el registro, orientación y la unión de las nubes de puntos, con este programa, debido a que no es el habitual para realizarlo y quería saber hasta que nivel era funcional y preciso como otros como el Cyclone o el propio Trimble RealWorks, ya que el ReCap tiene un coste muchísimo menor a los nombrados.

8.3. Autodesk Revit 2018.

Autodesk Revit es un software de diseño inteligente de modelado BIM para arquitectura e ingeniería, que puede facilitar las tareas de diseño de proyectos y los flujos de trabajo. La mayor característica de este software es que todo el contenido del modelado se obtiene en 3D de forma instantánea cuando desarrollamos proyectos de abajo hacia arriba a través de objetos inteligentes (familias de parámetros). Revit se basa en BIM: métodos de trabajo colaborativos y utiliza modelado paramétrico de objetos de construcción y elementos estructurales.

Con este software generaremos el modelo tridimensional, una vez realizado la unión de las nubes de puntos. Una de las características por las que elegí el Autodesk Revit 2018, es la compatibilidad entre los demás programas "hermanos" de Revit, como es el propio AutoCad.

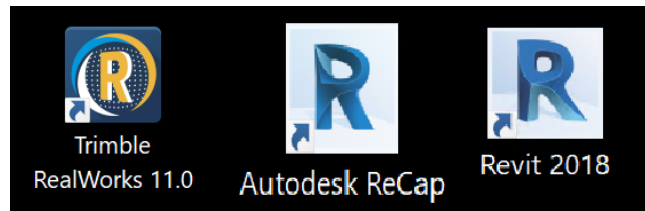


Figura 5: Software utilizado.

9. Descripción del proceso del modelaje.

El primer paso que hay que realizar en el procesado de los datos es la conversión de los datos que “salen” del láser escáner a un formato que sea editable. Para esto iremos al Trimble RealWorks 11.0, y cargaremos los puntos. A continuación los exportaremos a formato .e57.

9.1. Registro de la nube de puntos.

Una vez que tengamos los archivos en .e57 empezaremos con el registro de las nubes de puntos. Abriremos el Autodesk ReCap y crearemos un proyecto e importaremos las nubes de puntos.

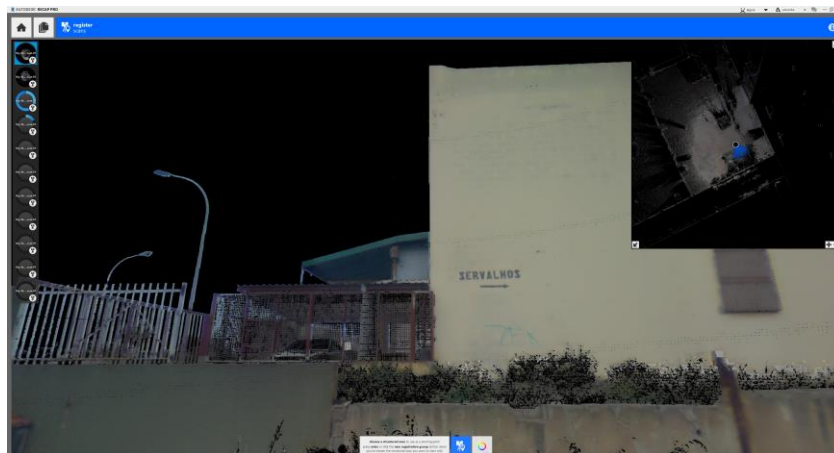


Figura 6: Registro de las nubes de puntos.

Una vez que todas las nubes de datos estén cargadas comenzamos con el registro de las nubes. Al no utilizar esferas en campo, utilizaré puntos homólogos (puntos que son los mismos entre dos imágenes o nubes) entre las nubes.

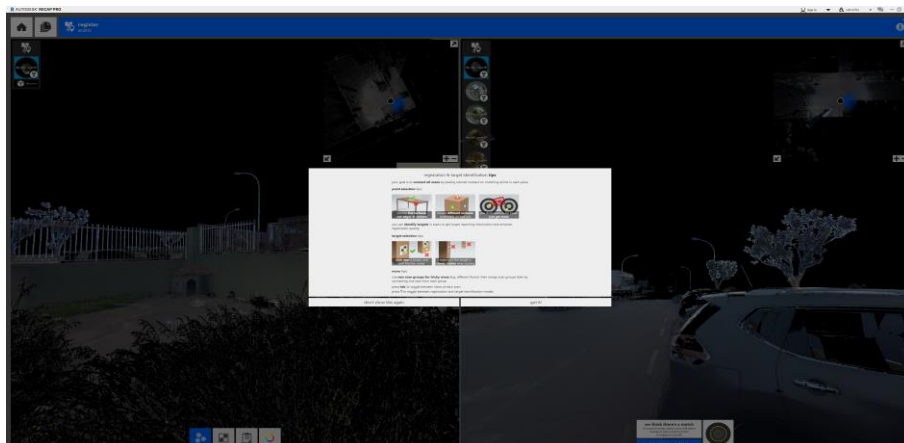


Figura 7: Explicación de cómo realizar el registro con el Autodesk ReCap mediante puntos homólogos.

Este proceso de selección de puntos homólogos entre nubes, se realizará igual entre todos los estacionamientos.

Cada vez que finalizamos el registro podemos ir viendo el resultado del “solape” generado y de la unión final tanto analítica como visualmente como se muestra en la imagen de a continuación, donde vemos en la parte de la derecha un croquis de las uniones realizadas (los puntos negros son las estaciones) y en la parte inferior vemos el “informe” donde nos informa del solape entre nubes y la precisión.

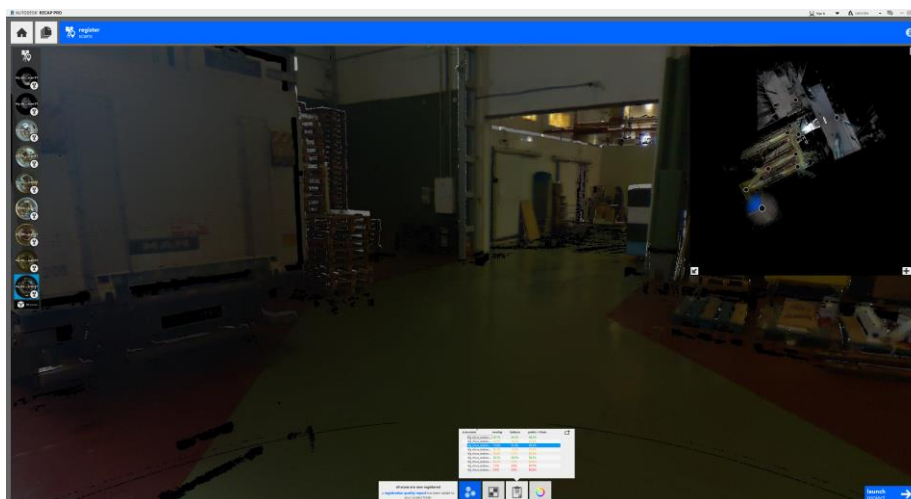


Figura 8: Informe de la nube de puntos resultado y croquis de las uniones.

En cualquier momento podemos visualizar el resultado que llevamos.



Figura 9: Registro de varias nubes de puntos.

Una vez realizado todo el registro, tendremos la nube de puntos, pero sin “limpiar”. Para acabar la parte del registro borraremos los puntos que son o bien sobrantes (edificios colindantes, carretera...) o generados por error (cruce de vehículos, cruce de personas...). Este proceso es sencillo: seleccionaremos los puntos que queremos eliminar y los suprimimos.

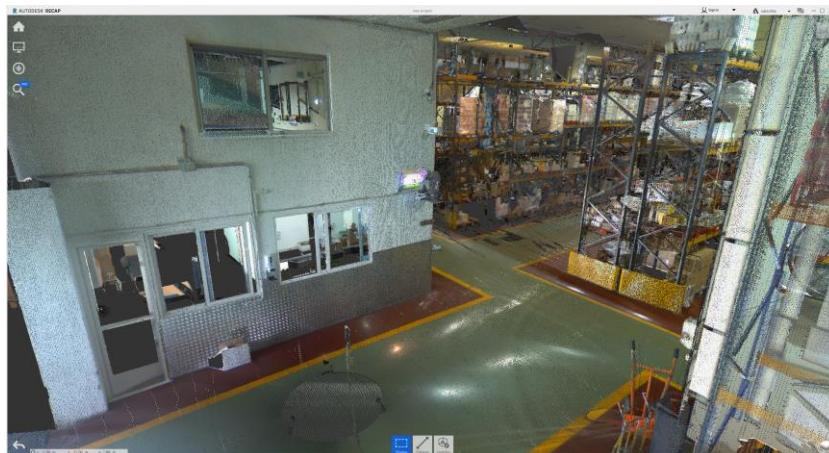


Figura 10: Interior de la nube de puntos.

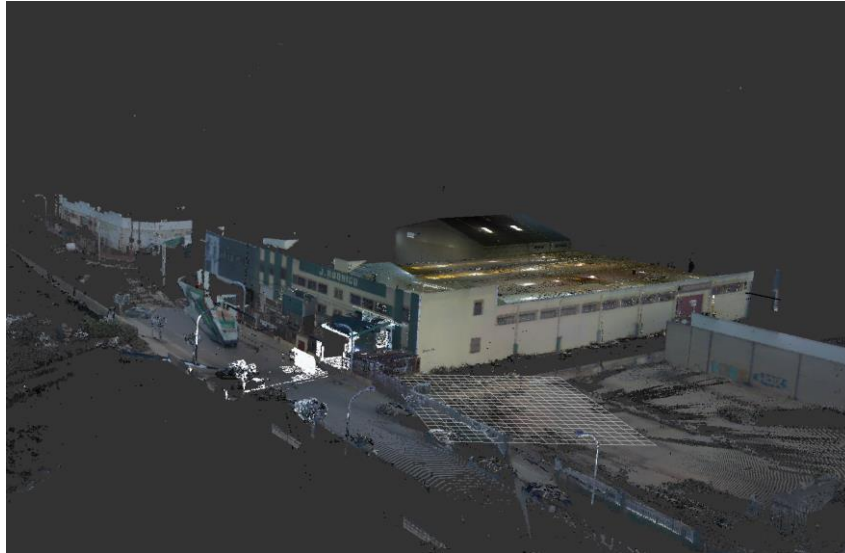


Figura 11: Nube de puntos sin limpiar.

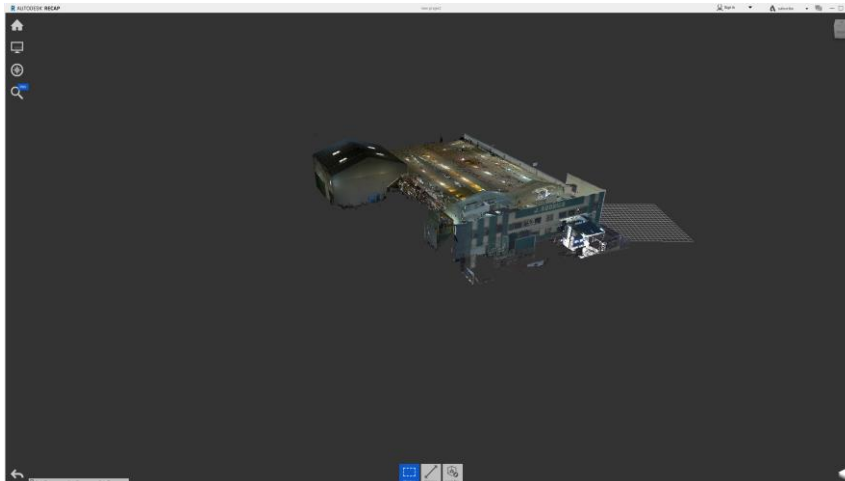


Figura 12: Nube de puntos con limpieza realizada.

Solo nos queda exportar la nube en formato .e57 para que pueda ser cargada en el Autodesk Revit.

9.2. Modelización tridimensional.

Con la nube de puntos ya limpiada y exportada, abrimos un proyecto en Revit como plantilla arquitectónica.

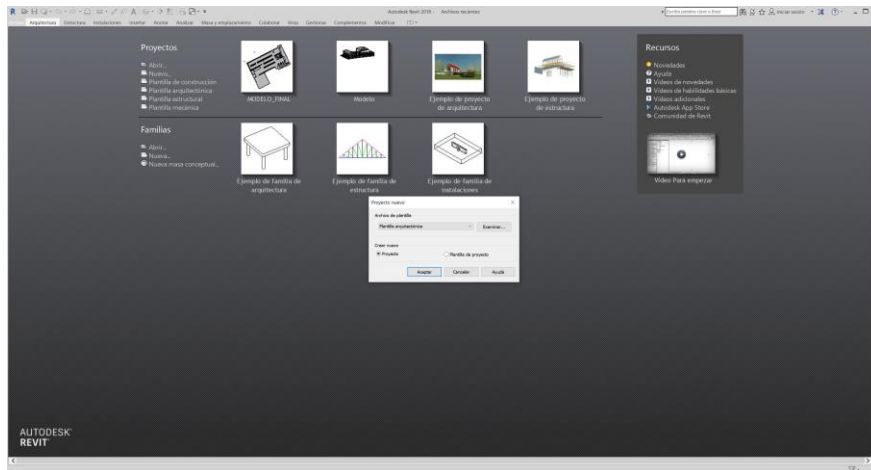


Figura 13: Creación del proyecto arquitectónico.

Ahora tenemos que insertar la nube de puntos para poder modelarla. Vamos a la pestaña insertar y ahí Nube de puntos.

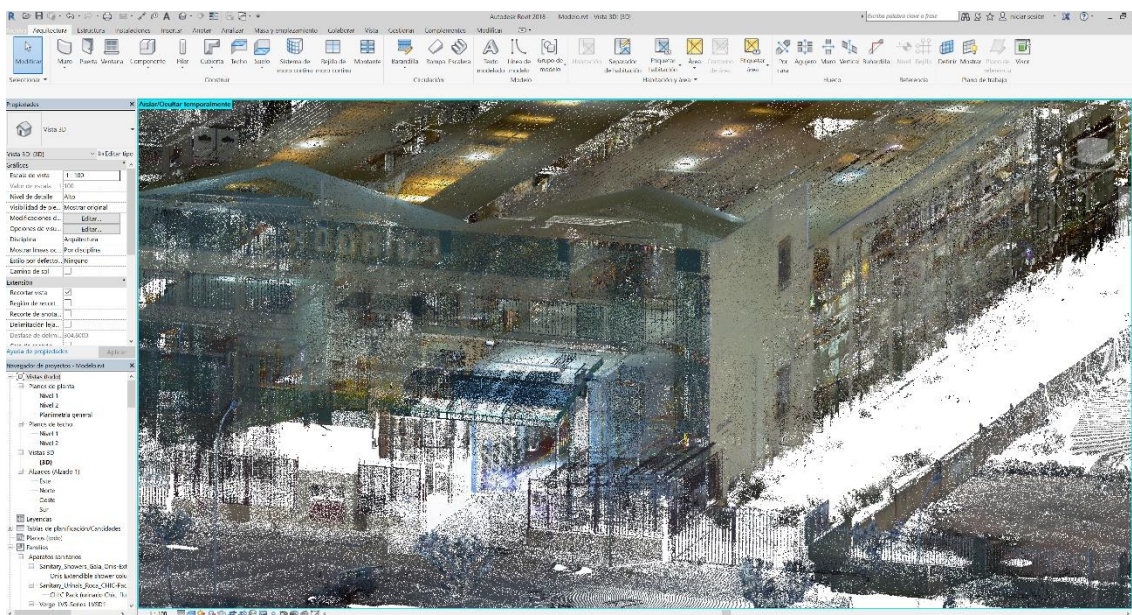


Figura 14: Nube de puntos cargada en Revit.

Ya podemos empezar a modelar la nube de puntos. Empezaremos con los muros exteriores, debido a que la nube de puntos está continuamente activada, por lo que se hace un poco tedioso modelar con la nube de puntos activada todo el rato.

Para crear los muros iremos a la pestaña arquitectura y dibujaremos los muros con la herramienta, muro.

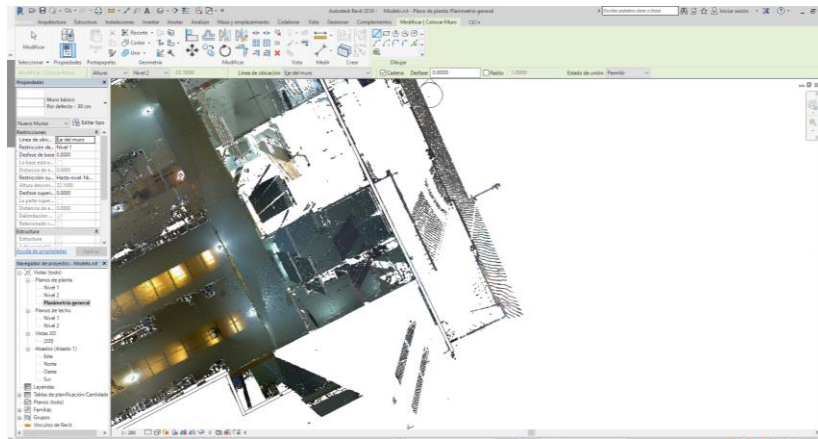


Figura 15: Creación de los muros exteriores.

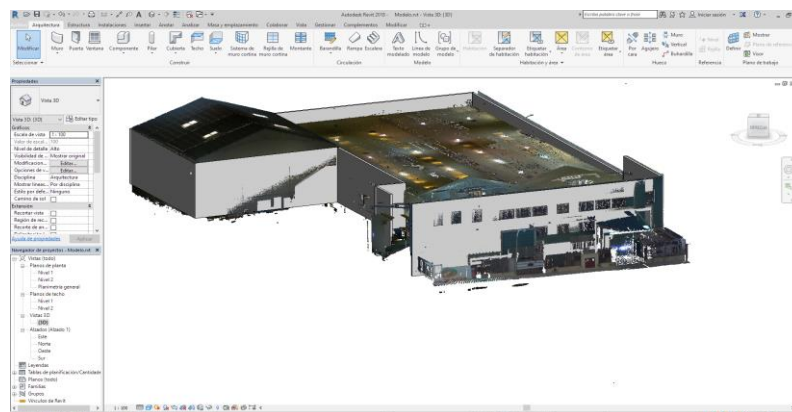


Figura 16: Creación de todos los muros exteriores.

Creación de los suelos, se realiza de la misma forma los muros, iremos a la pestaña arquitectura y dibujaremos los suelos por el centro de los muros. Para este paso es recomendable que se desactive la nube de puntos (seleccionamos la nube de punto y hacemos clic en las gafas en la parte inferior izquierda). Como podemos ver en la figura 17, hay suelo de color verde. Eso es debido a que esa estructura es el congelador al igual que la “L” que lo forma, y ese suelo es de un material diferente al suelo base.

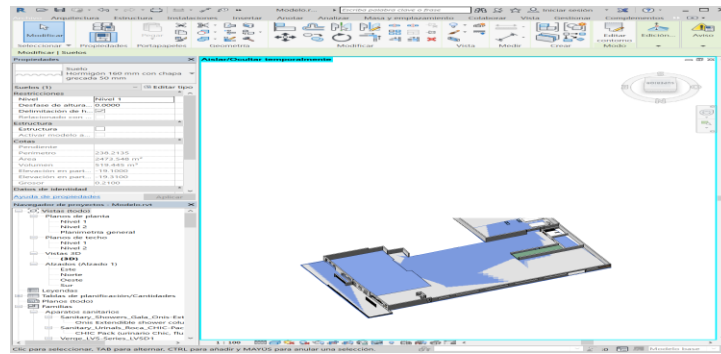


Figura 17: Creación de los suelos.

Creación de ventanas y puertas, elegimos en arquitectura puertas o ventanas y en los muros que hemos digitalizado hacemos clic en la posición que deseamos. Para este paso es recomendable activar la nube de puntos. Como en los muros y el suelo es editable en cualquier momento, siempre buscaremos lo más parejo a la realidad.

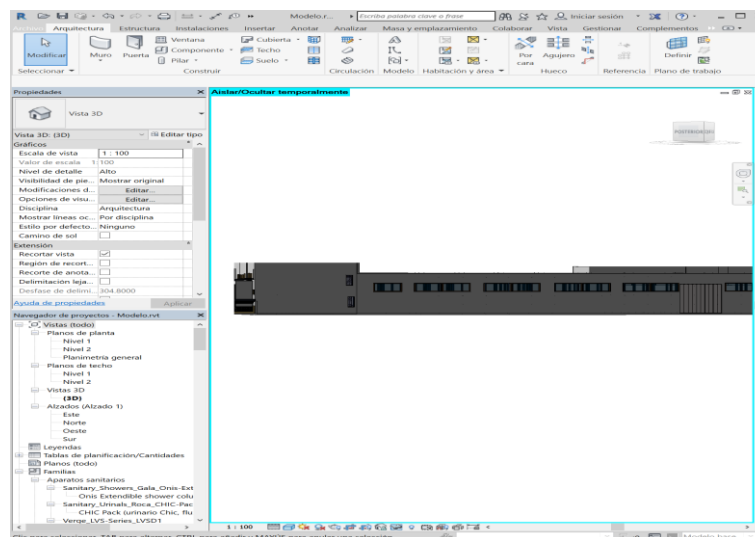


Figura 18: Creación de puertas y ventanas.

Ahora que tenemos las ideas básicas, generaremos los elementos decorativos o detalles. Con la herramienta muros, y la nube de puntos activada, haremos primero el contorno de las zonas de oficina. Recordamos que por tema de seguridad no se accedió al interior de las oficinas al igual que los congeladores, por lo que la distribución será aproximada.

Con los muros exteriores creados, situamos ventanas y puertas con la nube de puntos activada para situarlo en la zona correcta. Esto nos servirá para estructurar de mejor manera la zona interior. El siguiente paso para realizar será hacer la separación, mediante muros, de la parte inferior de las oficinas. La altura de los muros se ha podido

sacar gracias a que en el estacionamiento 2, a través de una ventana, se puede calcular la altura con la herramienta medir de ReCap.

Ya hemos separado las distintas habitaciones, así que empezamos a decorar nuestra parte inferior de las oficinas, hay que recordar que el suelo ya ha sido realizado para todo el modelo entero. Como Revit 2018 dispone de pocos elementos decorativos (llamados familias), por lo que deberemos buscar en Google elementos libre de uso diseñados por otros usuarios o realizarlo por nuestra cuenta. Una de las páginas utilizadas ha sido: <https://www.bimobject.com/es/product?sort=trending> que solo solicita hacer una cuenta y “definir” el uso que se le va a dar a los modelos que ya han generado otros usuarios, en nuestro caso ámbito estudiantil/universitario. Para añadir las familias, iremos a insertar y cargar familias. En el panel inferior nos permitirá añadirlos a nuestro modelo.

Deberemos poner techo entre las plantas, esto se realiza de la misma manera que el suelo. La única diferencia entre el suelo es que debemos añadir la altura a la que se encuentra el techo, desde el suelo.

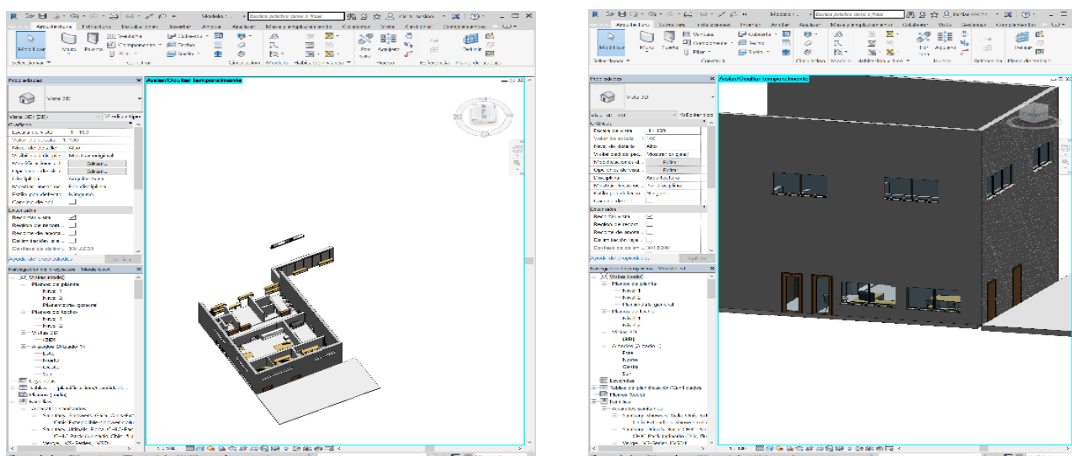


Figura 19: Creación de la planta cero de las oficinas.

El primer piso, segunda altura, se realiza de la misma manera que la planta cero. Y como en la planta cero deberemos dejar el agujero del ascensor.

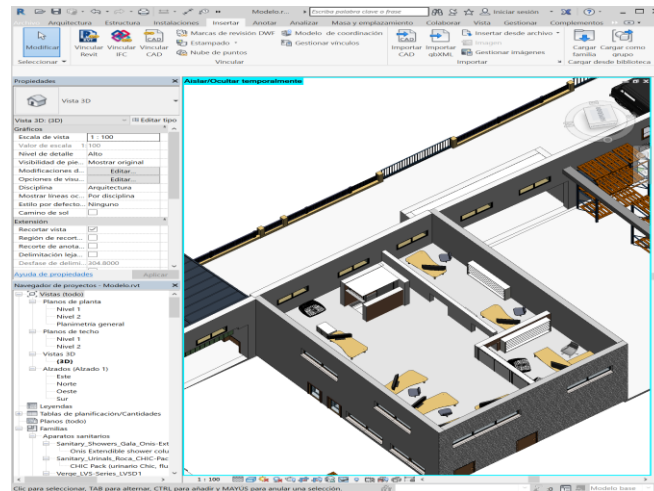


Figura 20: Creación de la primera y segunda planta de las oficinas.

Solo faltaría realizar el elemento más característico de la empresa, las estanterías. Estas estanterías han sido diseñadas a partir de dos modelos libre que fueron modificados para adaptarse a la realidad.

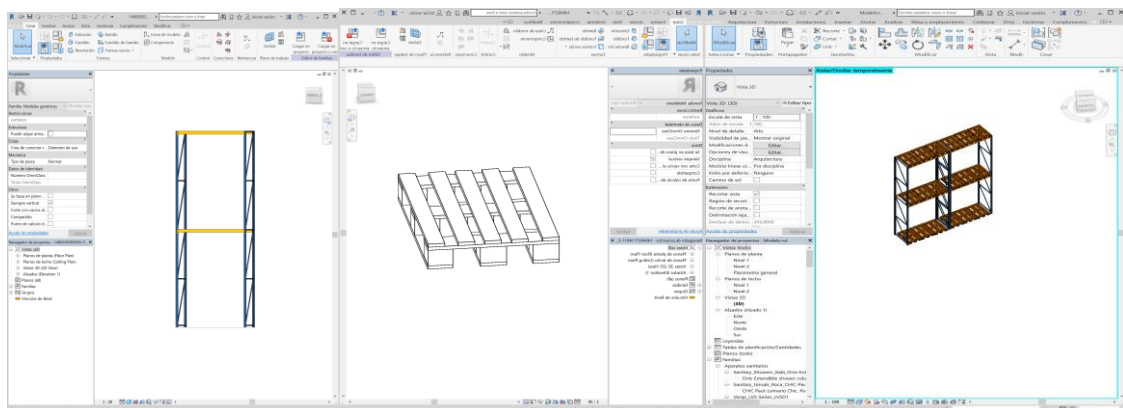


Figura 21: Proceso de creación de las estanterías.

Lo que queda para finalizar en nuestro modelo es insertar todas las puertas, ventanas, elementos decorativos (luminarias, estanterías, vehículos...), algún que otro muro o pilares, el vallado que tiene fuera de la nave (se utiliza una familia para el muro con reja y otra familia para las persianas) y los tres tipos diferentes de techados que tiene el edificio.

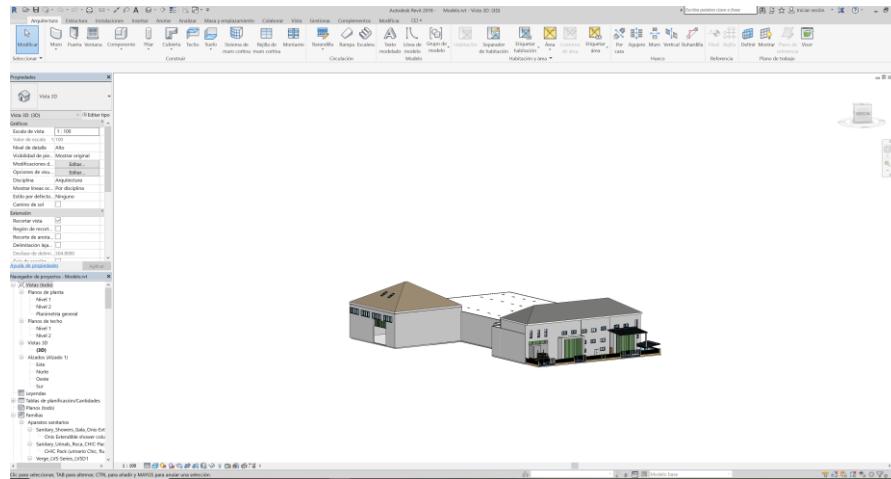


Figura 22: Modelo final desde el exterior.

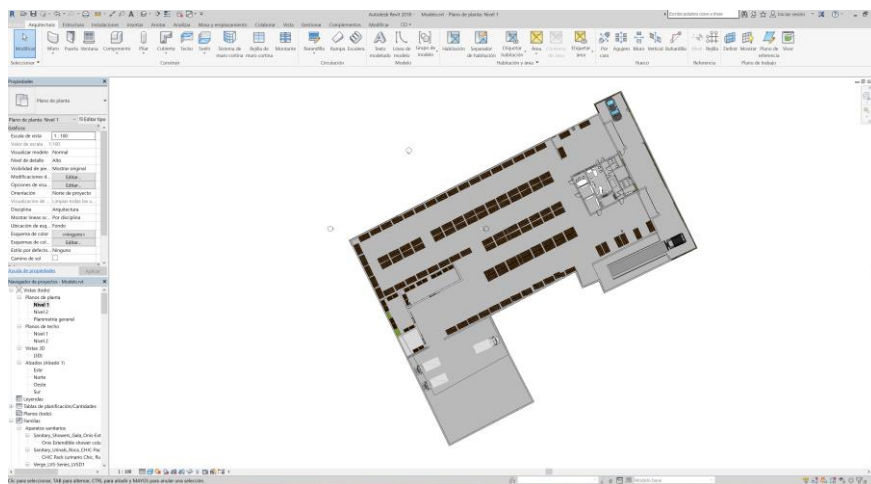


Figura 23: Plano de planta a altura 1 (altura suelo).

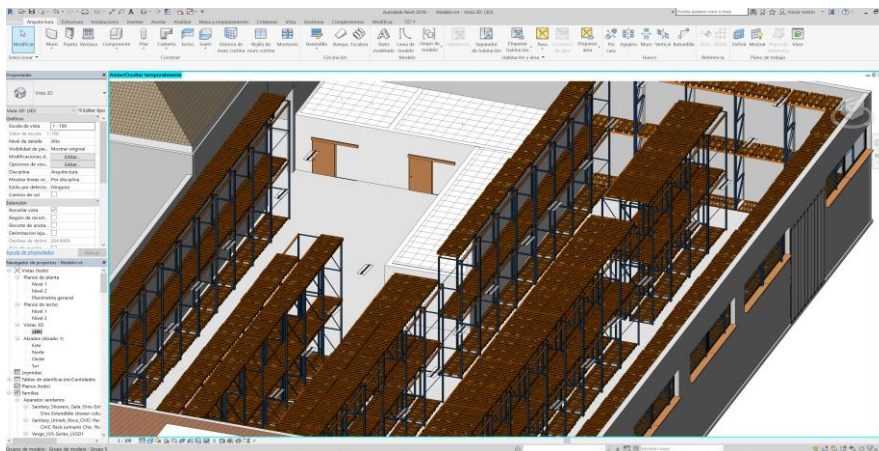


Figura 24: Vista del interior sin tejado.

10. Publicación del modelo tridimensional.

Para finalizar esta parte del T.F.G. queda “difundir” este modelo, para que la empresa lo pueda utilizar como herramienta de marketing digital. Se subirá el modelo a un servidor web llamado ACCA Software, que es un servidor de modelos BIM, para subir el modelo a la página web deberemos convertirlo en formato .ifc. Su único inconveniente es que necesitas un usuario y contraseña y que en ciertos elementos no guardan correctamente su color. Tanto para publicarlo como para subirlo a esta página web es muy sencillo, tan solo hay que seguir los pasos que nos indica la web.

<https://cloud.usbim.com/login>

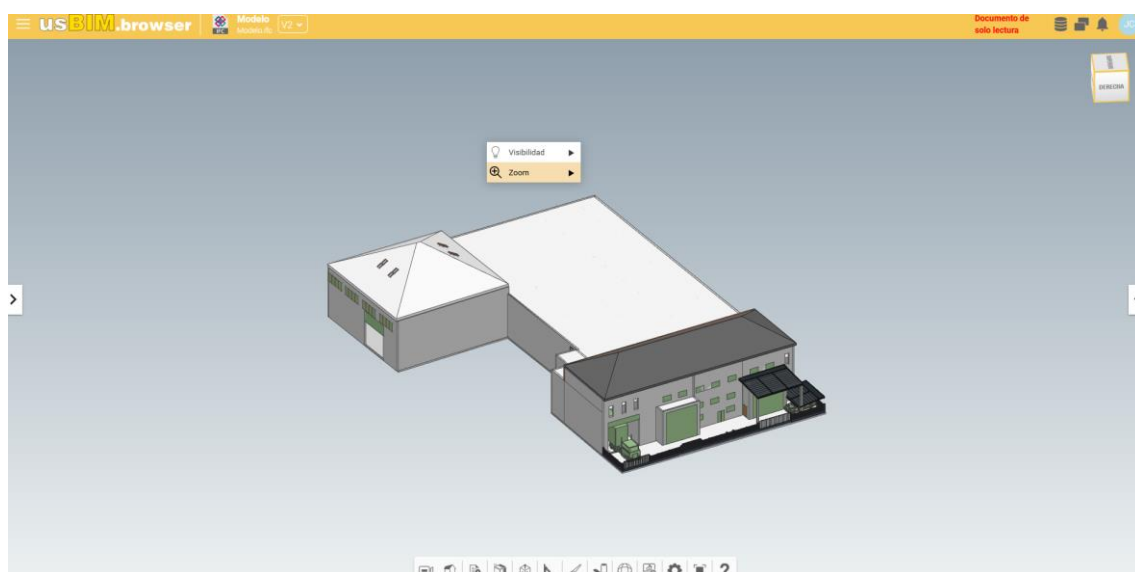


Figura 25: Publicación en ACCA Software del modelo resultado.

11. Base de datos NoSQL.

Una base de datos NoSQL (No sólo SQL), es una base de datos con diferente “estructura”(ya que tiene de estructura definida) a las bases de datos tradicionales, que nos permite almacenar grandes cantidades de información sin tener los problemas principales como son el rendimiento y escalabilidad de las bases de datos tradicionales. Y su principal diferencia respecto a otras bases de datos es que no deben mantener el esquema entidad-relación, ni tampoco mantener la forma de tabla, sino que almacena los datos en formato clave valor. Su evolución ha sido notoria gracias a las redes sociales como Twitter o Facebook.

11.1. Diferencias fundamentales entre bases de datos SQL.

- A nivel de implantación es mucho más rentable una base de datos NoSQL, ya que no necesita de tarifas de licencias de software y con un hardware modesto puede ser operativa.
- No está basada en el esquema entidad-relación (es decir forma de tabla), sino en clave-valor (su forma es conocida como JSON/YAML). Esta diferencia se ve clara en la Figura 24.
- El intercambio de datos en una base de datos NoSQL es mucho más rápida a la de una base de datos SQL, ya sea para consultar un dato como para actualizar valores.
- A diferencia de las bases de datos SQL, la NoSQL al ser una base abierta los desarrolladores deben introducir su propio código haciendo tedioso el sistema.



Figura 26: Comparación de base de datos SQL (izquierda) con bases de datos NoSQL (Derecha).

11.2. Sistema operativo en base Linux: Ubuntu.

Los programas y plataformas que se han utilizado para la realización de la base de datos NoSQL, están optimizados para Linux y no para Windows, por lo que en un pen drive instalé Ubuntu, que es un sistema operativo libre.

11.2.1. Docker.

Docker es una plataforma de software que le permite probar e implementar aplicaciones rápidamente.

Docker empaqueta el software en unidades estandarizadas llamadas que incluyen todo lo necesario para hacer el software, incluidas las bibliotecas, las herramientas del sistema y el tiempo de ejecución del código.

Con Docker, puede implementar y escalar aplicaciones rápidamente en cualquier entorno con la confianza de que su código se ejecutará.

Para realizar su instalación el profesor Ángel Marqués Mateu me facilitó un script para que la instalación fue algo más ágil. De igual modo desde la página de Docker, nos indica los comandos para instalarlo desde la terminal.

```
sudo apt-get update
sudo apt-get install \
  apt-transport-https \
  ca-certificates \
  curl \
  gnupg \
  lsb-release
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --
dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

sudo apt-get update

sudo apt-get install docker-ce
```

11.2.2. Fiware.

Fiware es una plataforma de software abierta que hace de intermediario entre MongoDB y Docker, que son lugares donde se almacenan la información.

Para realizar su instalación el profesor Ángel Marqués Mateu me facilitó un script para que la instalación fueran algo más ágil. De igual modo desde GitHub, plataforma donde los usuarios suben sus códigos, podemos acceder a un archivo parejo o instalarlo mediante líneas de comando desde la terminal.

```
docker pull mongo:4.4
docker pull fiware/orion
docker network create fiware_default
docker run -d --name=mongo-db --network=fiware_default \
  --expose=27017 mongo:4.4 --bind_ip_all
docker run -d --name fiware-orion -h orion --network=fiware_default \
  -p 1026:1026 fiware/orion -dbhost mongo-db
```


11.2.3. MongoDB.

Mongo DB es un sistema de base de datos NoSQL, de código libre. Como hemos explicado anteriormente no almacena la información como tablas, sino que la almacena como una estructura BSON, haciendo que las interacciones entre las aplicaciones y el cliente se realicen fácil y rápido.

11.2.4. Solicitudes GET/POST

Solicitudes o peticiones **GET**, son solicitudes que mandas al “servidor” que están incluidas en la URL de la dirección y nos muestran lo pedido. Es decir los parámetros de búsqueda se encuentran la URL. Podemos ver dos ejemplos de esta petición a continuación.

(1) `http://192.168.1.133:1026/v2/entities?type=product&limit=300`

(2) `curl -G -X GET 'http://192.168.1.133:1026/v2/entities/urn:ngsi-Id:Jose_Rodrigo_e_Hijos_SA:product:23092050' -d 'options=keyValues'`

Ambas son peticiones GET, pero como podemos observar hay alguna diferencia y eso es debido a que (1) es una petición realizada al servidor desde el navegador, es como si buscáramos algo en Google. En esta petición se solicita que muestre de ese listado los primeros 300 productos. En la (2) es una petición GET realizada desde la terminal y en este caso esa petición nos mostrará sólo el producto con esa ID solicitada es decir un solo valor. Y en las 2 lo que nos devuelve la petición es lo que se conoce como JSON o YAML.

```
{
  "id": "urn:ngsi-Id:Jose_Rodrigo_e_Hijos_SA:product:23092050",
  "type": "product",
  "brand": {
    "type": "Text",
    "value": "ACME",
    "metadata": {}
  },
  "description": {
    "type": "Text",
    "value": "Lentéjas ecologicas",
    "metadata": {}
  },
  "position": {
    "type": "Text",
    "value": "D",
    "metadata": {}
  },
  "shelf": {
    "type": "Text",
    "value": "A",
    "metadata": {}
  },
  "weight": {
    "type": "Float",
    "value": 1000,
    "metadata": {}
  }
}
```

Figura 27: Script de carga de datos.

Solicitudes o peticiones **POST**, son solicitudes que mandas al “servidor” para actualizar información. La carga de los datos mediante peticiones POST, las cuales se realizarían de la manera que tenemos a continuación (código de ejemplo). Para insertar todos los valores podríamos utilizar la misma petición, sólo cambiarían los valores de los atributos.

```
$ curl -iX POST 'http://localhost:1026/v2/entities' -H 'Content-Type: application/json' \  
-d '{"type": "Dummy", "id": "urn:ngsi-ld:Dummy:X", "name": {"type": "Text", "value": "X"}}'
```

Como podemos observar la estructura es muy pareja entre ambas, pero debemos tener claro que GET sólo nos permite ver los datos si necesitamos actualizar algún dato deberemos usar POST.

11.3. Carga de los datos de la base de datos mediante script.

Para hacer la carga de los datos al servidor que tenía en formato .csv, utilicé un script que me facilitó el profesor Ángel Marqués Mateu y yo posteriormente lo adapté, debido al gran número de datos.

```
import json  
import requests  
  
with open('../listado_ID_completo.csv', 'rt') as stock:  
    records = stock.readlines()  
  
fiware_type = 'product'  
fiware_company = 'Jose_Rodrigo_e_Hijos_SA'  
url = 'http://localhost:1026'  
end_point = '/v2/entities'  
headers = {'Content-Type': 'application/json'}  
  
for rec in records[1:]:  
    print(rec)  
    product_id, brand, product_desc, weight, shelf, position, price, units = rec.strip().split(',')  
    weight = float(weight)  
    price = float(price)  
    units = float(units)  
    fiware_record = {  
        'id': 'urn:ngsi-ld:' + fiware_company + ':' + fiware_type + ':' + product_id,  
        'type': fiware_type,  
        'brand': {  
            'type': 'Text',  
            'value': brand  
        },  
        'description': {  
            'type': 'Text',  
            'value': product_desc  
        },  
        'weight': {  
            'type': 'Float',  
            'value': weight  
        },  
        'shelf': {  
            'type': 'Text',  
            'value': shelf  
        },  
        'position': {  
            'type': 'Text',  
            'value': position  
        },  
        'price': {  
            'type': 'float',  
            'value': price  
        },  
        'units': {  
            'type': 'Float',  
            'value': units  
        }  
    }  
    #print(fiware_record)  
    response = requests.post(  
        url = end_point,  
        headers = headers,  
        data = json.dumps(fiware_record)  
    )  
    print(response)
```

Figura 28: Script de carga de datos.

12. Realización de página web.

Con la base de datos ya hecha lo óptimo para que el público pueda acceder a ella y verla es mediante una página web, por lo que creé una web funcional programándola con el software Visual Studio Code.

12.1. Lenguajes utilizados.

Como no solo iba a crear una web que mostrará las peticiones decidí hacer una web funcional por lo que tuve que usar diversos lenguajes de programación, aunque la base está obviamente realizada con HTML.

12.1.1. HTML.

HTML es el lenguaje básico en cualquier Web. Estructura y define el contenido web. En conjunto a CSS (*Cascading Style Sheets*) permite modificar la apariencia de la propia web.

Este lenguaje depende de ciertas etiquetas especiales como son <head>,<body>,<div>...

12.1.2. PHP.

Es un lenguaje de programación apto para HTML, que nos permite realizar acciones entre un servidor y el cliente. Para el diseño de la página web lo utilizaremos como intermediario, para “contacto”.

12.1.3. JAVASCRIPT.

JavaScript es un lenguaje de programación con una idea similar a CSS, es un complemento para “diseñar” nuestro HTML. En este caso particular el código creado se utiliza para modificar el menú, para optimizarlo al tamaño de pantalla.

12.1.4. YAML/JSON.

JSON es un formato de texto sencillo para el intercambio de datos. YAML es un tipo JSON el cual simplifica el texto. En la figura siguiente se puede ver la comparación entre ambos la diferencia entre ambos es claramente el diseño.

| JSON | YAML |
|---|--|
| <pre>{ Servers: [{ name: Server1, owner: John, created: 123456, status: active }] }</pre> | <pre>Servers: - name: Server1 owner: John created: 123456 status: active</pre> |

Figura 29: Comparación entre JSON y YAML.

12.2. Software utilizado.

12.2.1. Visual Studio Code.

Como he nombrado anteriormente el software que he utilizado para programar el Visual Studio Code, que es libre y está disponible tanto para Windows como Linux. Este programa es muy funcional debido a que te permite programar en un gran número de lenguajes.

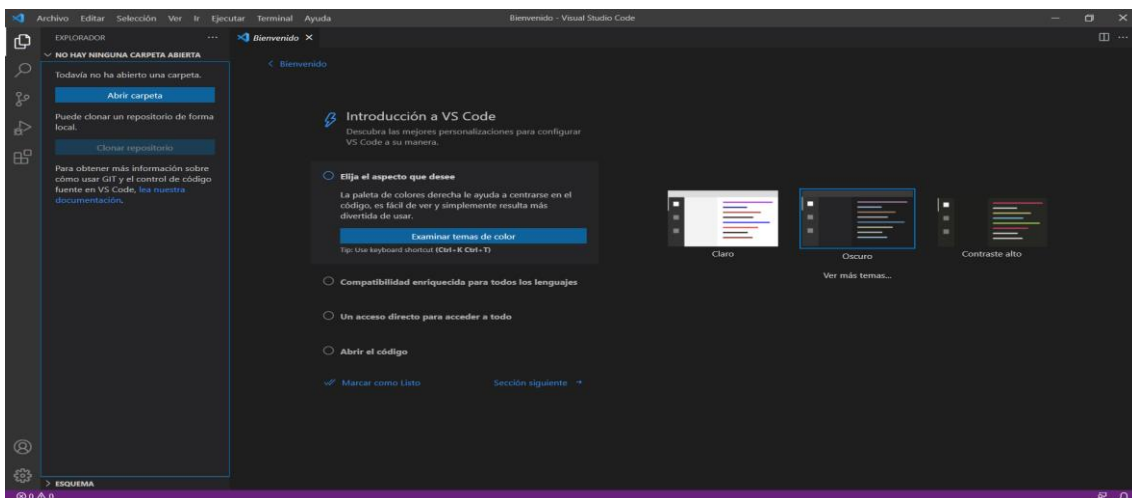
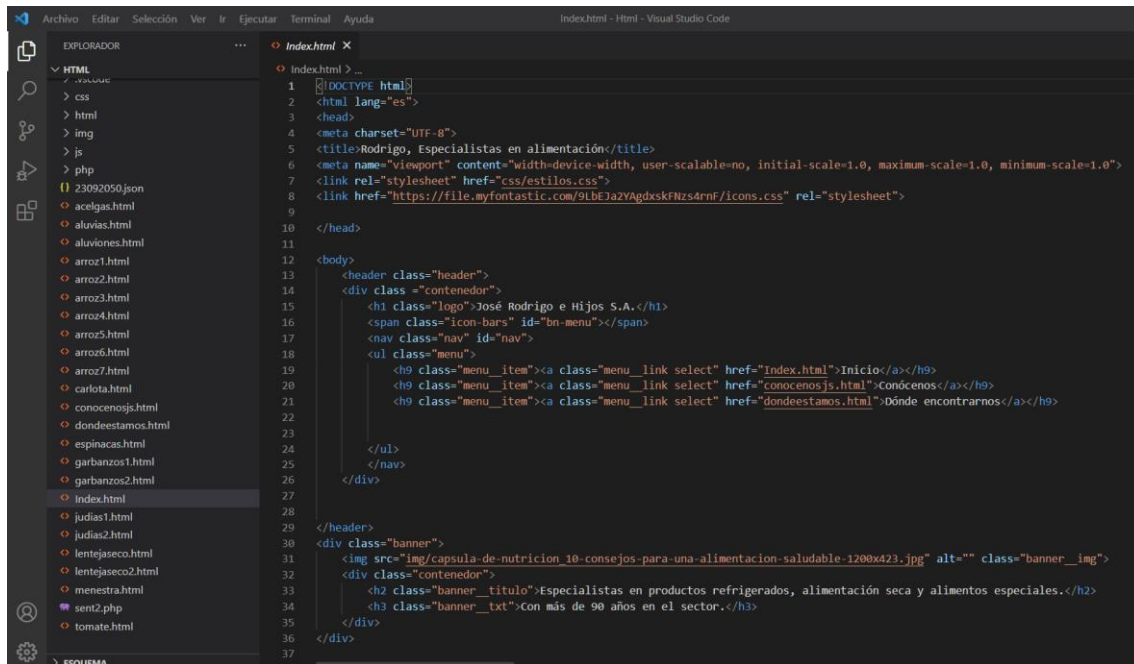


Figura 30: Visual Studio Code.



```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <title>Rodrigo, Especialistas en alimentación</title>
6 <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
7 <link rel="stylesheet" href="css/estilos.css">
8 <link href="https://file.myfontastic.com/9LbE3a2YAgdxskFNz4rnF/icons.css" rel="stylesheet">
9
10 </head>
11
12 <body>
13 <header class="header">
14 <div class="contenedor">
15 <h1 class="logo">José Rodrigo e Hijos S.A.</h1>
16 <span class="icon-bars" id="bn-menu"></span>
17 <nav class="nav" id="nav">
18 <ul class="menu">
19 <li class="menu_item"><a class="menu_link select" href="Index.html">Inicio</a></li>
20 <li class="menu_item"><a class="menu_link select" href="conocenos.js.html">Conócenos</a></li>
21 <li class="menu_item"><a class="menu_link select" href="dondeestamos.html">Dónde encontrarnos</a></li>
22
23 </ul>
24 </nav>
25 </div>
26 </div>
27
28 </header>
29
30 <div class="banner">
31 
32 <div class="contenedor">
33 <h2 class="banner_titulo">Especialistas en productos refrigerados, alimentación seca y alimentos especiales.</h2>
34 <h3 class="banner_text">Con más de 90 años en el sector.</h3>
35 </div>
36 </div>
37

```

Figura 31: Ejemplo de código final en HTML.



```

1 <?php
2 header("Content-type: text/html;charset=utf-8");
3 $nombre = $_POST['nombre'];
4 $mail = $_POST['email'];
5 $telefono = $_POST['telefono'];
6 $asunto = $_POST['asunto'];
7 $empresa = $_POST['mensaje'];
8
9 $header = "From: " . $mail . "\r\n";
10 $header .= "X-Mailer: PHP/" . phpversion() . "\r\n";
11 $header .= "Mime-Version: 1.0 \r\n";
12 $header .= "Content-Type: text/plain";
13
14 $mensaje = "Este mensaje fue enviado por " . $nombre . ",\r\n";
15 $mensaje .= "Su e-mail es: " . $mail . "\r\n";
16 $mensaje .= "Asunto: " . $asunto . "\r\n";
17 $mensaje .= "teléfono: " . $telefono . "\r\n";
18 $mensaje .= "Mensaje: " . $empresa . "\r\n";
19 $mensaje .= "Enviado el " . date('d/m/Y', time());
20
21 $para = 'chi20ros20@gmail.com';
22 $asunto = 'Mensaje de mi sitio web';
23
24 if (mail($para, $asunto, utf8_decode($mensaje), $header))
25 echo "<script type='text/javascript'>alert('Tu mensaje ha sido enviado exitosamente!');</script>";
26 echo "<script type='text/javascript'>window.location.href='Index.html';</script>";
27
28

```

Figura 32: Código creado para envío de formulario en PHP.

```
JS menu.js X
js > JS menu.js > ...
1 var bnmenu= document.getElementById("bn-menu");
2 var nav= document.getElementById("nav");
3 bnmenu.addEventListener('click', function(){nav.classList.toggle("mostrar");})
4 function resetHref() {
5     location.hash = '';
6 }
```

Figura 33: Código creado para ocultar el menú en JavaScript.

```
{} 23092050.json X 22092025.html
{} 23092050.json > ...
1 [{"id":"urn:ngsi-ld:Jose_Rodrigo_e_Hijos_SA:product:23092050","type":"product","brand":{"type":"Text","value":"ACME","metadata":{}},"desc
```

Figura 34: Ejemplo de JSON.

```
# estilos.css X 22092025.html
css > # estilos.css > ...
1 @import url('https://fonts.googleapis.com/css2?family=Acme&display=swap');
2 @import url('https://fonts.googleapis.com/css2?family=Acme&family=Nanum+Myeongjo&display=swap');
3 *{box-sizing: border-box;}
4 img{display:block;max-width: 100%;}
5
6 body{ font-family: 'Nanum Myeongjo', serif; margin: 0%;background: #rgb(202, 201, 201);}
7
8 h1,h2,h3,h4,h5,h6{font-family: 'Acme', sans-serif; letter-spacing: 1.5px;}
9 h2,h3{ text-align:center;
10 color: #fff;
11 font-size: 40px;
12 letter-spacing: 0;
13 text-shadow: -3px -3px 3px #00000F, 3px -3px 3px #00000F, -3px 3px 3px #00000F, 3px 3px 3px #00000F;}
14 h4{font-family: 'Acme', sans-serif; font-size: 30px;color: #rgb(2, 2, 2);}
15 h9{font-family: 'Acme', sans-serif; letter-spacing: 3px;font-weight: bold;}
16 .contenedor{
17 margin:auto;
18 width: 99%;
19 }
20 /*Cabecera*/
21 .header{ height: 60px;}
22 .header {display: flex; justify-content: space-between;}
23 .contenedor1 {display: flex; justify-content: space-between;}
24 .logo, .icon-bars{margin: 10px;color: #rgb(255, 0, 0);}
25 font-size: 40px;
26 letter-spacing: 0;
27 text-shadow: -3px -3px 3px #ffffff, 3px -3px 3px #ffffff, -3px 3px 3px #ffffff, 3px 3px 3px #fdfdfd;}
28 .icon-bars{cursor:pointer;margin-left: auto;display: block;width: 40px;height: 40px; font-size: 30px;background: #cef011; color: #fff}
29
30
31 /*Menu desplegable*/
32 .nav{
33 position: absolute;
34 top: 151px;
35 left: -100%;
36 width: 100%;
37 transition: all 0.5s;
38 }
```

Figura 35: Ejemplo de código de diseño CSS.

Cómo el formato JSON/YAML no permite modificaciones ni de letra, ni de color, transformé las peticiones a formato HTML. Para realizar esta tarea utilicé un conversor online para que me lo transformara a HTML.

<https://www.textcompare.org/yaml/yaml-to-html/>

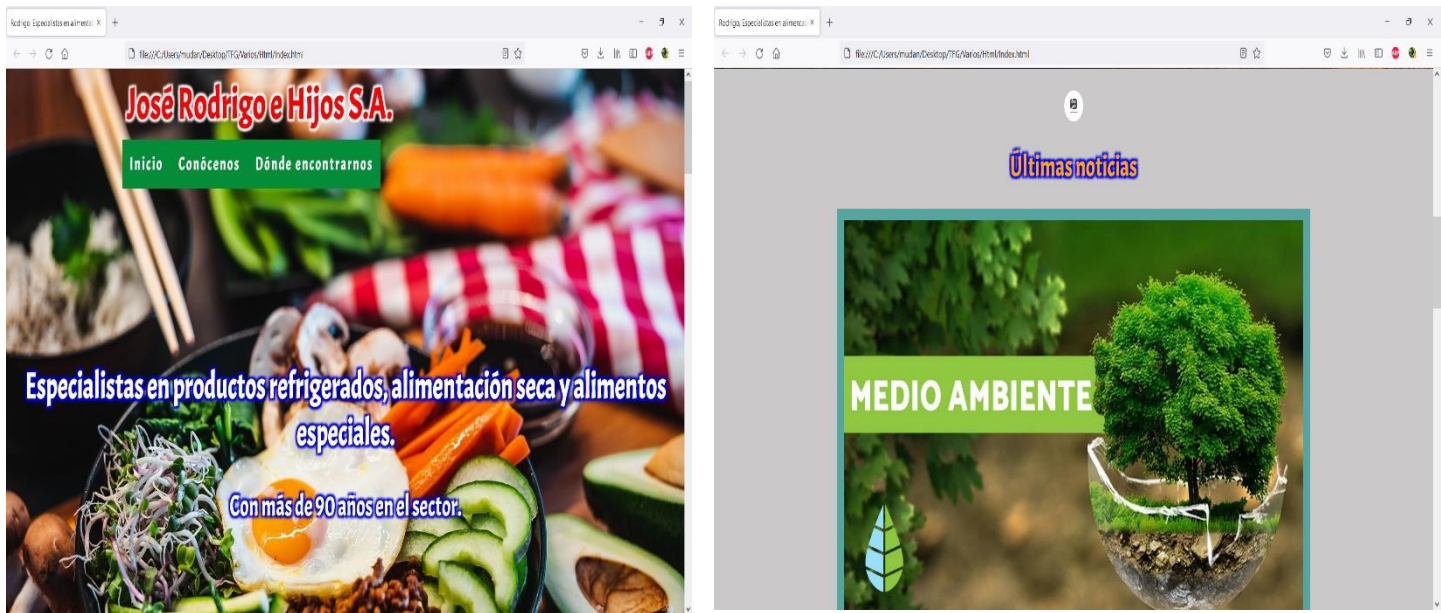


Figura 36: HTML final. Portada.

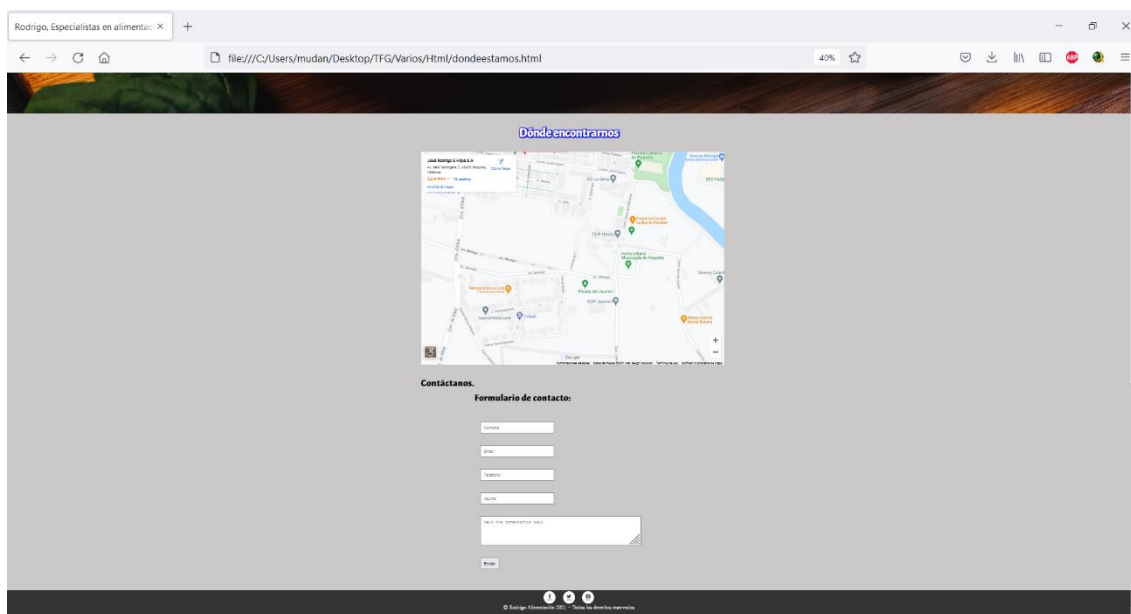


Figura 37: HTML final. Dónde encontrarnos.

13.1. Software utilizado.

13.1.1. Microsoft Power Bi.

Power BI es un servicio de análisis de datos de Microsoft orientado a proporcionar visualizaciones interactivas y capacidades de inteligencia empresarial con una interfaz lo suficientemente simple como para que los usuarios finales puedan crear por sí mismos sus propios informes y paneles.

Para comenzar a realizar el dashboard, cargaremos los datos que tenemos en el servidor y aunque Power BI está basado en Excel(base de datos “tradicional”) nos permite cargar los datos sin problemas. Para ello nos iremos a “Obtener datos” y seleccionaremos Web. Allí le insertaremos la petición al servidor.

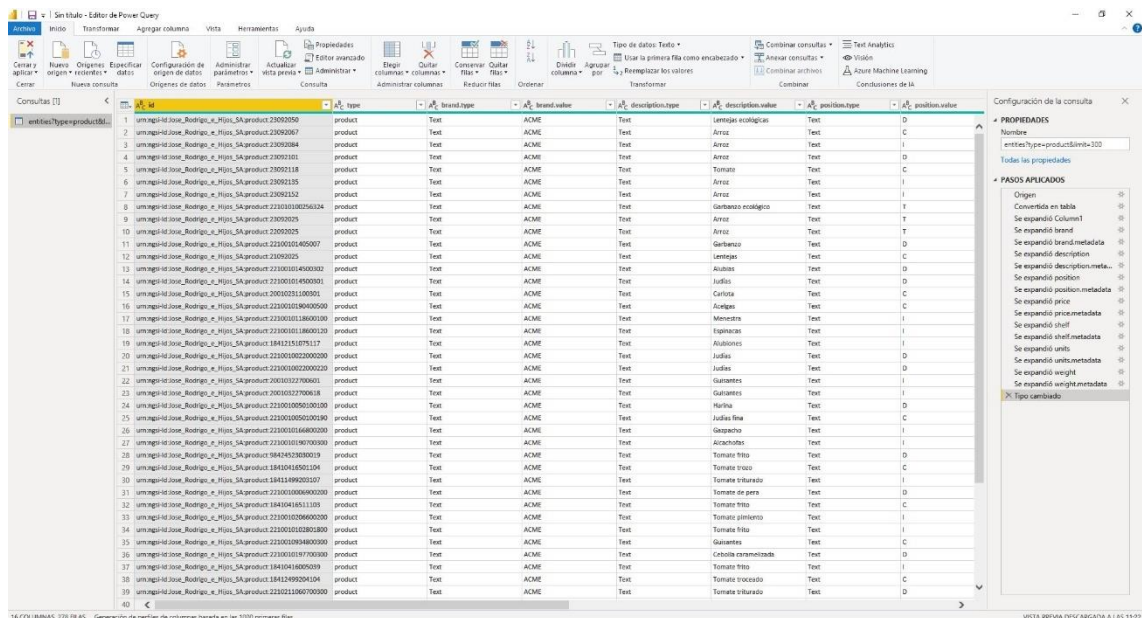


Figura 40: Carga de datos desde el servidor a Power BI.

Como se puede observar en la Figura 38 hay columnas que no son necesarias, haciendo clic sobre ellas las eliminaremos para no tener datos innecesarios.

Como es normal por temas de confidencialidad el valor de los productos, los clientes, y las tarifas han sido inventadas. Por lo que ahora crearemos todas las “tablas” que nos sean necesarias y generaremos las relaciones necesarias entre ellas. Estas relaciones sirven para que al seleccionar un dato el cual es común en varias tablas genere eventos y puedas consultar datos desde otra tabla. Normalmente se generan de manera automática, pero hay casos en los que no. Para solventar esa no relación iremos a modelo y haremos clic sobre la tabla que queremos enlazar.

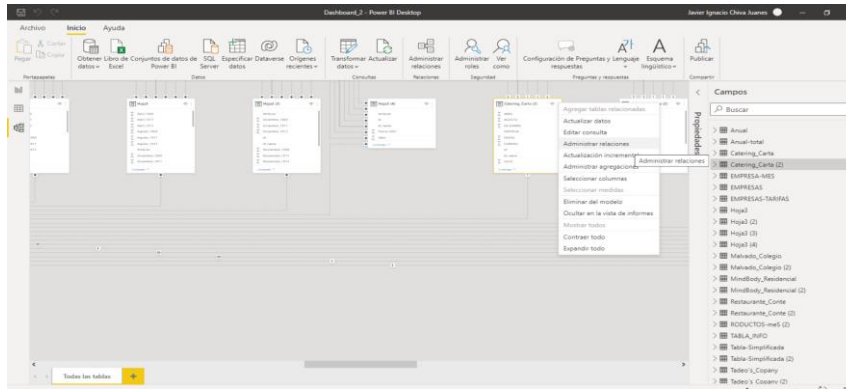


Figura 41: Administración de relaciones manual.

Haremos clic en nuevo crearemos la relación entre la o las tablas que necesitemos.

Una vez finalizado todas las relaciones y haber preparado los datos, se puede empezar el diseño del dashboard. Para insertar cualquier tipo de histograma es muy sencillo: elegimos una forma de representarlo y le añadimos los valores que deseemos. En la ventana de formato podemos cambiar los colores de representación de la leyenda como del histograma así como el título entre otras cosas.

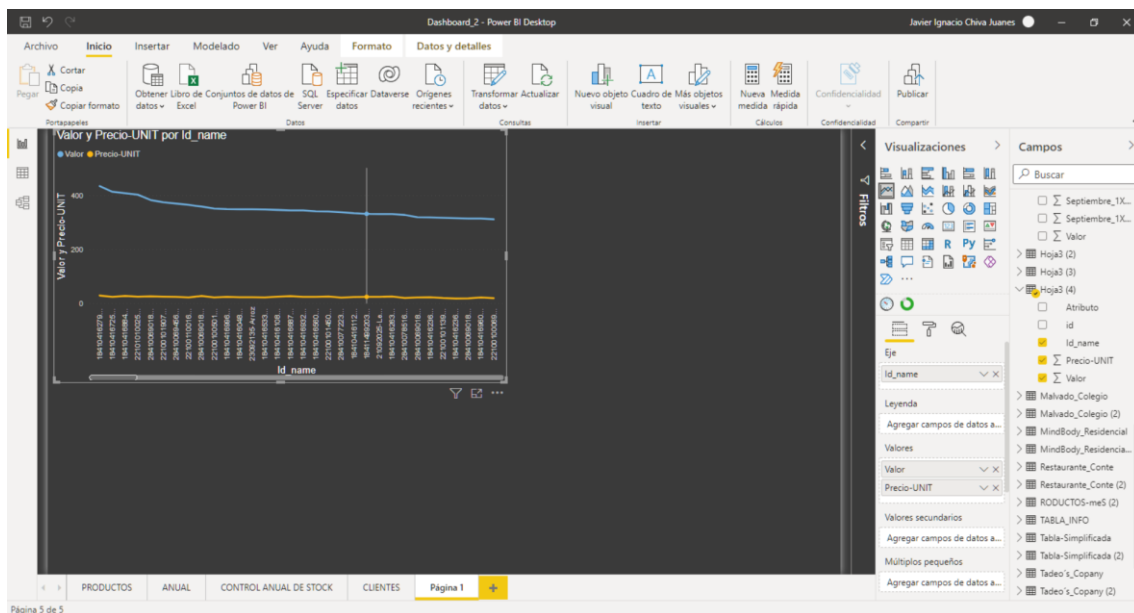


Figura 42: Inserción de forma visual los datos.

Una vez acabado todo el diseño, el cual está todo automatizado para poder elegir verlo de manera independiente como en grupo quedaría de la siguiente manera.

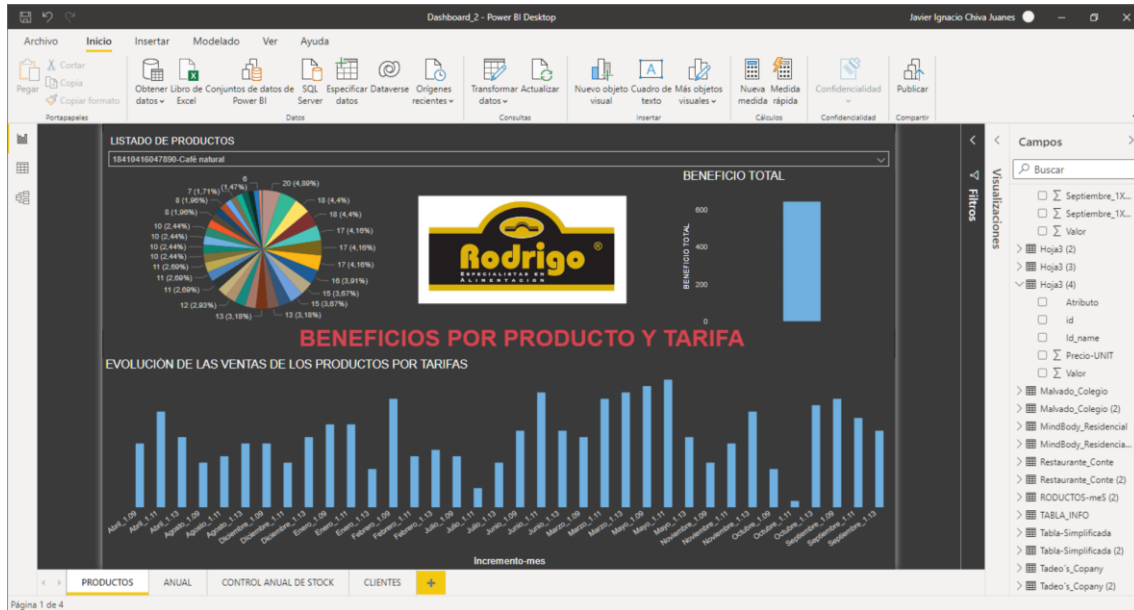


Figura 43: Dashboard final. Productos.

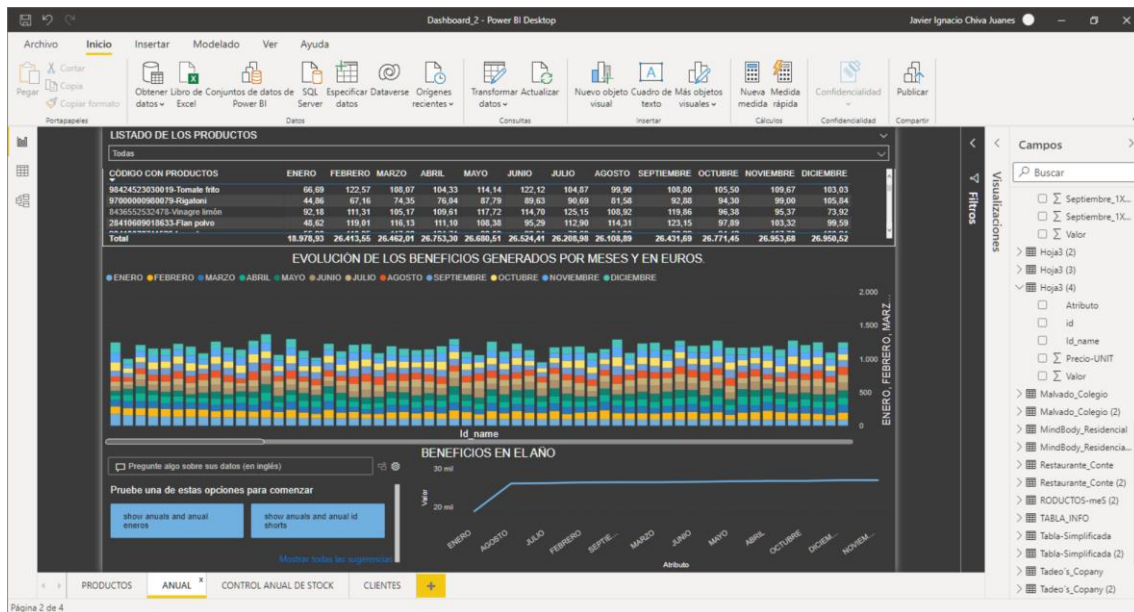


Figura 44: Dashboard final. Anual.

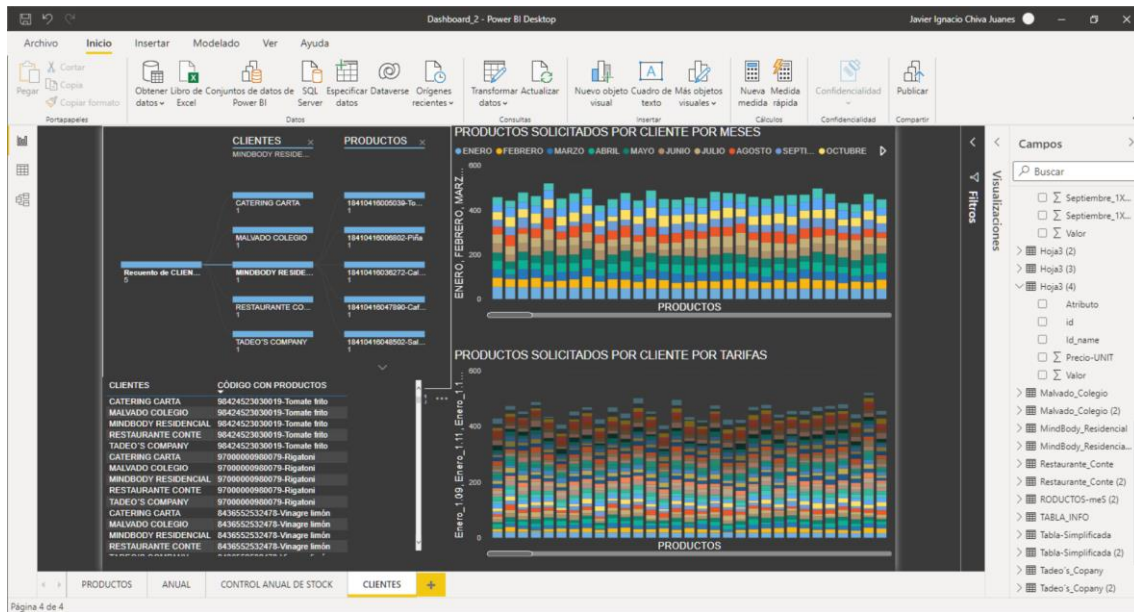


Figura 45: Dashboard final. Clientes.

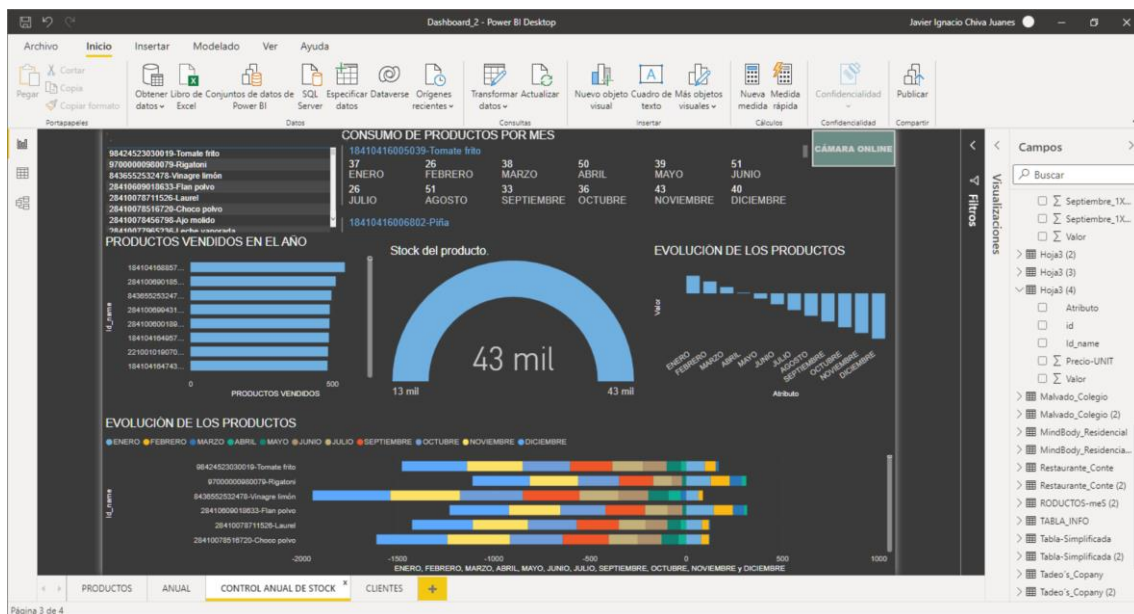


Figura 46: Dashboard final. Control anual del stock .

Como se observa en la figura 41 en la parte superior derecha hay un botón el cual mediante una cámara IP nos permite observar a tiempo real el producto que esté asignado. Esta cámara IP ha sido creada con ayuda de una Raspberry Pi 4 y el software Motion.

13.2. Raspberry Pi 4.

13.2.1. Sistema operativo en base Linux: Raspbian.

Para optimizar el control del stock y que en cualquier momento puedas controlar cuanto producto dispones a tiempo real es mediante cámaras IP y con una Raspberry podemos enlazar esas cámaras. Como una Raspberry está limitado por sus componentes lo más idóneo es instalarle su sistema operativo. En este caso utilizaré Raspbian, basado en Linux.



Figura 47: Raspberry Pi 4.

13.2.1.1. Cámara IP.

Las cámaras IP, en este caso solo será una ya que no se disponía de más, serán webcams ya que son rentables por su precio, por su resistencia y su acople.



Figura 48: Webcam-Cámara IP.

13.2.1.1.1. Motion.

Motion es un software abierto que realiza las funciones de videovigilancia, pero lo adaptaremos a nuestras necesidades, que al fin y al cabo son las mismas: controlar en tiempo real. Este programa permite conectar varias cámaras.

Para su instalación iremos a la terminal de Raspbian y ejecutaremos la siguiente línea de comando:

```
sudo apt-get install motion
```

Y para realizar la configuración de la aplicación ejecutaremos:

```
sudo nano /etc/motion/motion.conf
```

Aquí estableceremos los parámetros que deseemos, en este caso he modificado el tamaño de pantalla, he activado el parámetro Daemon para que se ejecute en segundo plano, he activado la opción de conexiones fuera de la red (el parámetro más importante, ya que gracias a esto desde cualquier punto del mundo con la IP podemos comprobar a tiempo real la mercancía) y desactivaremos la opción de capturas, ya que esto consume mucha memoria.

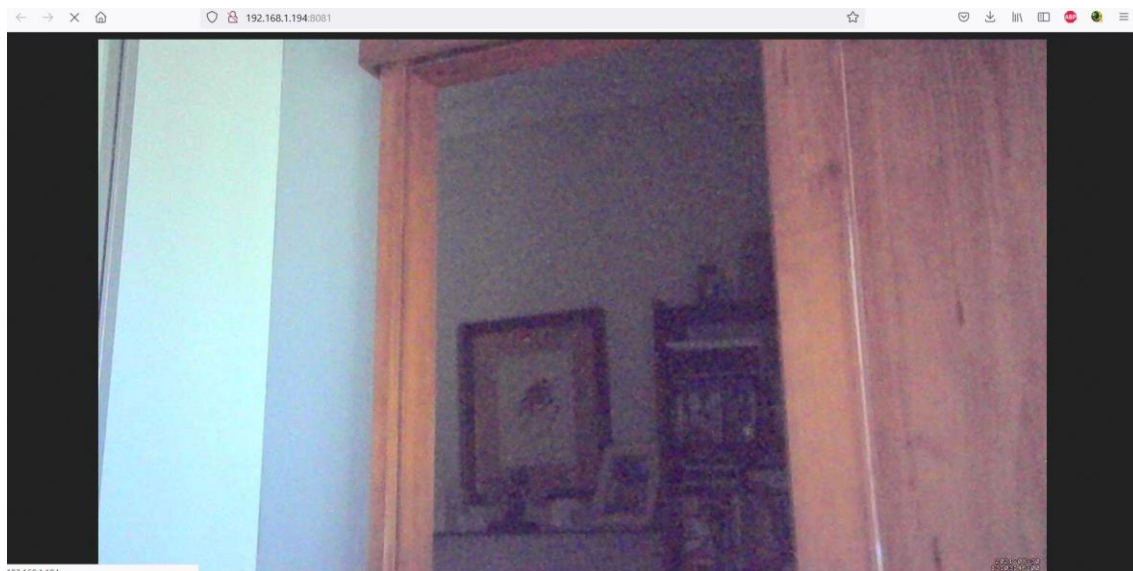


Figura 49: Captura a tiempo real.

No se ha podido realizar la toma en la propia empresa por descuadre de horarios

14. Acciones de mejora u optimización.

Al ser una gran nave industrial aproximadamente 350 estanterías y en cada estantería de media unos 3 productos diferentes necesitaríamos al menos una cámara IP por estante y al menos 3 Raspberry por pasillo, para tener un control.

Pero si lo que deseamos es un valor de volumen para cada producto, podríamos incorporar sensor vl53l0x, junto a las cámaras, que nos mide una distancia y situándolo

en un punto central, podríamos calcular el volumen aproximado de los productos de manera indirecta. Este sensor es asequible ya que se puede conseguir por un precio alrededor de 20€ y si se comprara al por mayor mucho más rentable.



Figura 50: Sensor VL53L0X.

15. Presupuesto.

En el presupuesto se ha cogido como valores de salario el Convenio de Empresas de Ingeniería y Oficinas de Estudios Técnicos del año 2020.

| | PRECIO | Nº DE PAGAS | TOTAL |
|---------------------------|---------|-------------|----------|
| SALARIO BASE | 1291.04 | 14 | 18074.56 |
| PLUS POR CONVENIO | 2349.69 | | 2349.69 |
| SALARIO BRUTO ANUAL | | | 20424.25 |
| Precio por hora realizada | | | 9.12 |

Materiales

| | |
|---|-------|
| Láser escáner- Trimble TX6 (por un día de alquiler) - I.V.A. incluido | 726 |
| Ordenador de sobremesa | 1200 |
| Sensor- I.V.A. incluido | 24.14 |
| Raspberry pi 4 - I.V.A. incluido | 65 |

Licencias

| | |
|---|--------|
| Trimble Real Works (Licencia semanal) - I.V.A. incluido | 344.85 |
| Autodesk ReCap Pro (Licencia mensual) - I.V.A. incluido | 55 |
| Autodesk Revit (Licencia mensual) - I.V.A. incluido | 291 |
| Microsoft Office | 18.89 |
| Fiware (Libre uso) | 0 |

| | |
|---|------|
| Docker | 0 |
| Microsoft Power BI (Incluido en Microsoft 365 E5) - I.V.A. incluido | 8.45 |

| MANO DE OBRA | | | |
|--------------------------|------|-----|-----------|
| 250 HORAS DE TÉCNICO | 9.12 | 250 | 2280 |
| COSTE TOTAL DEL PROYECTO | | | 5004.88 € |

16. Conclusiones.

A nivel personal, la meta que me había propuesto de intentar plasmar todos los conocimientos posibles que he adquirido a lo largo de la titulación creo que la he conseguido. Mirando el punto de vista de la empresa en cuestión, espero que esto le favorezca para que ganen clientes, por la parte de la publicación del modelo y la construcción de la base de datos nueva, y que fidelicen los que tenían gracias al dashboard, viendo en que clientes pueden ajustar más los precios.

Para finalizar, volver a comentar que los puntos a mejorar serían los relacionados al control del stock tanto por un mayor número de webcams y por la implementación de los sensores métricos.

17. Bibliografía.

Trimble. *Trimble RealWorks*. <<https://es-la.geospatial.trimble.com/products-and-solutions/trimble-realworks>>

AECO COMPETENCE CENTER. *¿Qué es Revit de Autodesk y para qué sirve?*. <<https://www.rfaeco.com/que-es-revit-de-autodesk-y-para-que-sirve/>>

CADBLOCKSFREE. *CAD MODELS*. <<https://www.cadblocksfree.com/es/>>

BIMOBJECT. *Objetos BIM*. <<https://www.bimobject.com/es/product?sort=trending>>

YOUTUBE, "Uso de nube de puntos en Revit" en Youtube <<https://www.youtube.com/watch?v=vO89EEw7928>>

YOUTUBE, "CLASE DE MODELADO BIM DE NUBE DE PUNTOS USANDO #REVIT Y #ARCHICAD" en Youtube <<https://www.youtube.com/watch?v=qVpl9VjVHjo>>

ACENS. *Bases de datos NoSQL. Qué son y tipos que nos podemos encontrar*. <<https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>>

Docker. *Install Docker Engine on Ubuntu*. <<https://docs.docker.com/engine/install/ubuntu/>>

Amazon. *¿Qué es Docker?*. <https://aws.amazon.com/es/docker/>

Wikipedia. *Power BI*. <https://es.wikipedia.org/wiki/Power_BI>

Fiware. <<https://www.fiware.org/>>

Ubuntu. *Ubuntu Desktop*. <<https://ubuntu.com/download>>

Raspberry Pi. *Raspberry Pi Desktop*. <<https://www.raspberrypi.org/software/operating-systems/>>

Geekland. *Sistema de videovigilancia con detector de movimiento Motion*. <<https://geekland.eu/sistema-de-videovigilancia-motion/>>

GitHub. <<https://github.com/>>

MDN Web Docs *moz://a*. HTML: Lenguaje de etiquetas de hipertexto. <<https://developer.mozilla.org/es/docs/Web/HTML>>

MDN Web Docs *moz://a*. CSS. <<https://developer.mozilla.org/es/docs/Web/CSS>>

MDN Web Docs *moz://a*. JavaScript. <<https://developer.mozilla.org/es/docs/Web/JavaScript>>

MDN Web Docs *moz://a*. PHP. <<https://developer.mozilla.org/es/docs/Glossary/PHP>>

YOUTUBE, "Como hacer un FORMULARIO con HTML y PHP desde cero" en Youtube <<https://www.youtube.com/watch?v=e2R9uvjKW5g>>

Lázaro, Diego (2018) "Formularios en PHP. Tutorial para crear formularios con los principios básicos, validación y campos requeridos y subida múltiple de archivos" <
<https://diego.com.es/formularios-en-php>>

YOUTUBE, "Curso de HTML5 desde CERO (Completo)". en Youtube <
<https://www.youtube.com/watch?v=kN1XP-Bef7w&t=7627s>>

YOUTUBE, "TUTORIAL HTML desde Cero | COMPLETO y ACTUALIZADO en Visual Studio Code HTML5 Para PRINCIPIANTES" en Youtube.
<https://www.youtube.com/watch?v=vz4z0RLcAyk&t=79s>

Microsoft. Introducción a Power BI Desktop. < <https://docs.microsoft.com/es-es/power-bi/fundamentals/desktop-getting-started>>

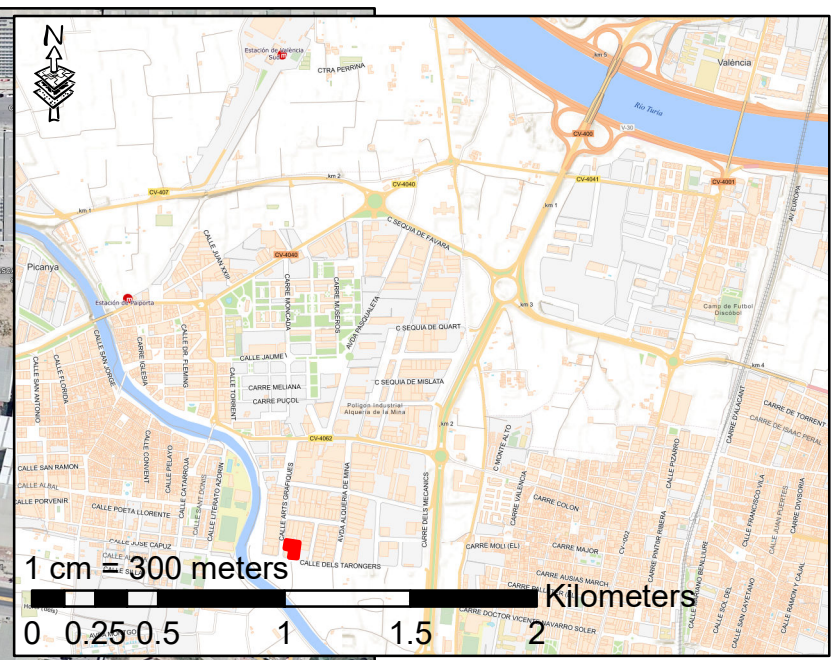
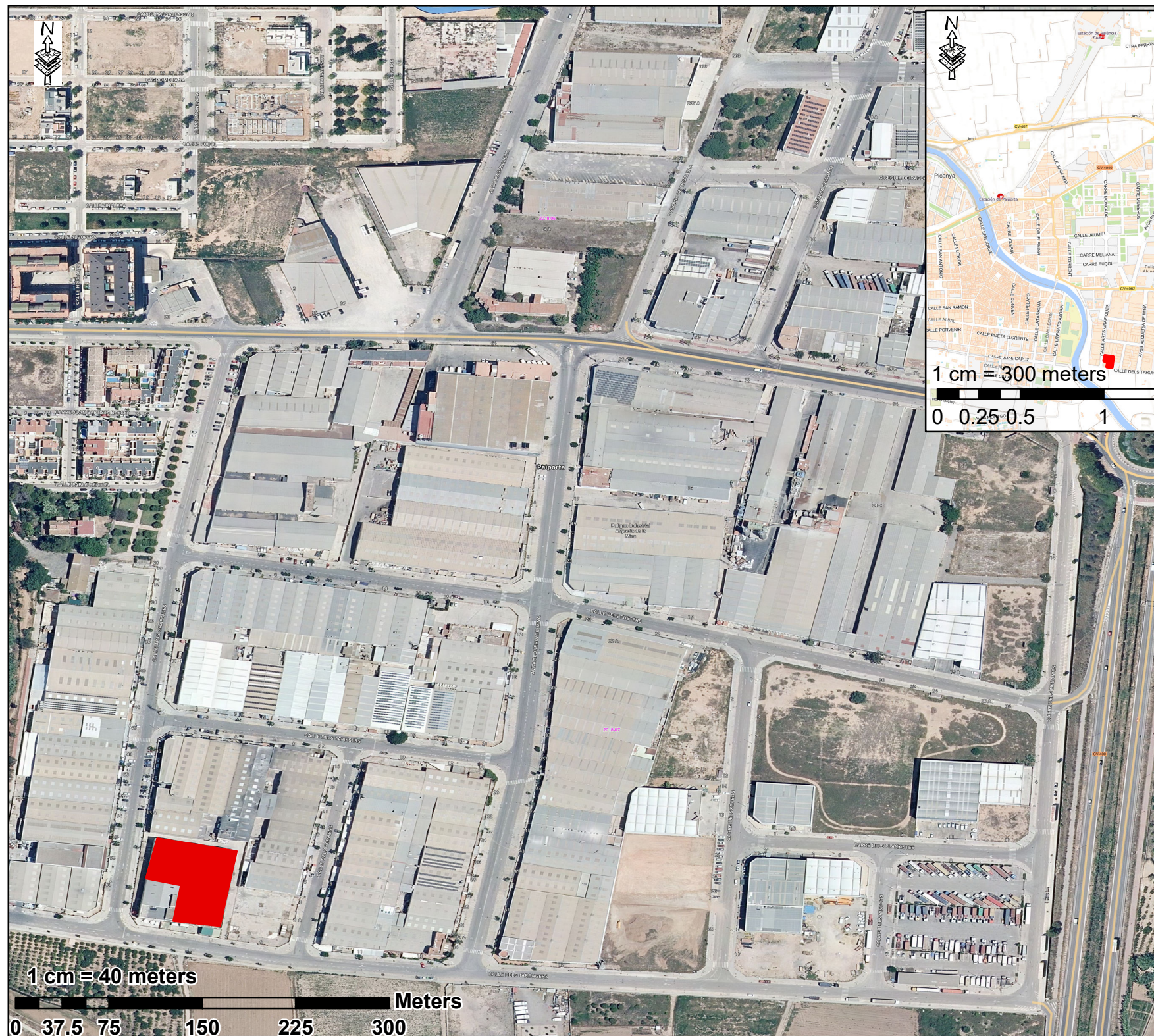
YOUTUBE, "Tutorial Power BI (2020) - Creación de Dashboard en 3 horas" en Youtube. <
<https://www.youtube.com/watch?v=hKe7bHPiSPU>>

Ivaylova Ilieva. T. (2020). MODELIZACIÓN 3D DEL YACIMIENTO ARQUEOLÓGICO DEL TOS PELAT. Trabajo Final de Máster. Valencia: Universitat Politècnica de València,<
https://riunet.upv.es/bitstream/handle/10251/150897/MemoriaFinal_TFM_TerezalvaylovaIlieva.pdf?sequence=1&isAllowed=y>

Valero Fernández, J.I. (2018). Levantamiento industrial y modelización 3D de un helicóptero. Trabajo Final de Grado. Valencia: Universitat Politècnica de València,<
<https://riunet.upv.es/bitstream/handle/10251/106644/VALERO%20-%20Levantamiento%20industrial%20y%20modelizaci%c3%b3n%203D%20de%20un%20helic%c3%b3ptero.pdf?sequence=1&isAllowed=y>>

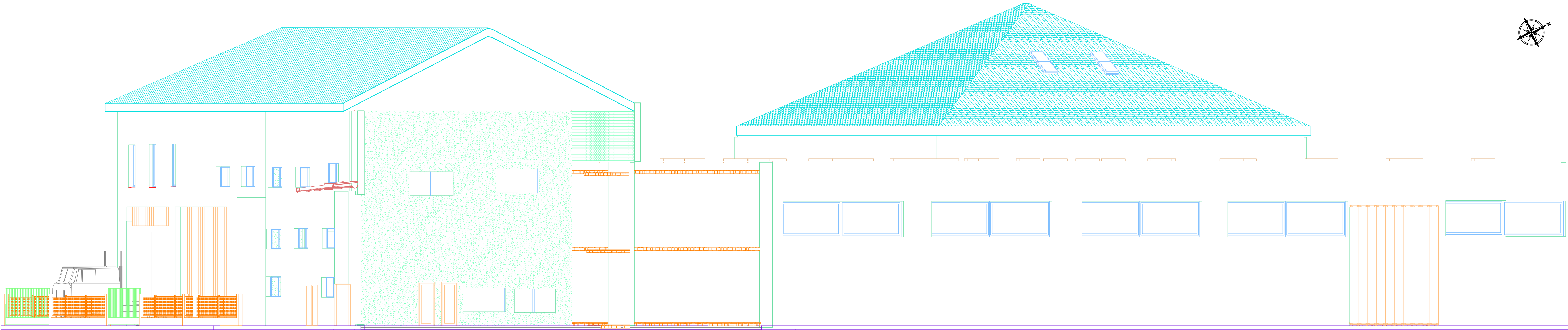


18. Cartografía.

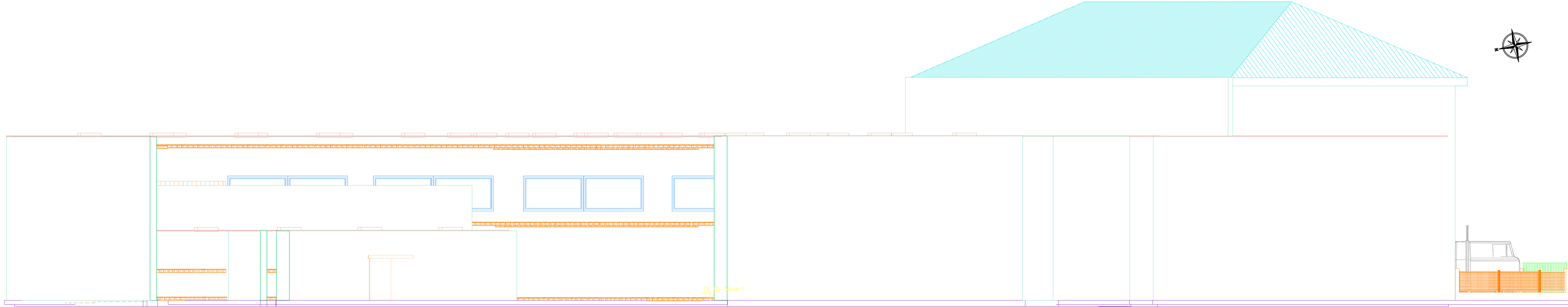


| | | | |
|---|-------------------------------------|---|---|
| | UNIVERSITAT POLITÈCNICA DE VALÈNCIA | | ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA EN GEOGRÁFICA CARTOGRÁFICA Y TOPOGRÁFICA |
| ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA EN GEOGRÁFICA CARTOGRÁFICA Y TOPOGRÁFICA | | TÍTULO DEL PLANO: PLANO DE SITUACIÓN | FECHA: 05/08/2021 |
| AUTOR: JAVIER IGNACIO CHIVA JUANES | | ESCALA: | NÚMERO DE PLANO: 01 |

PLANOS EN ALZADOS



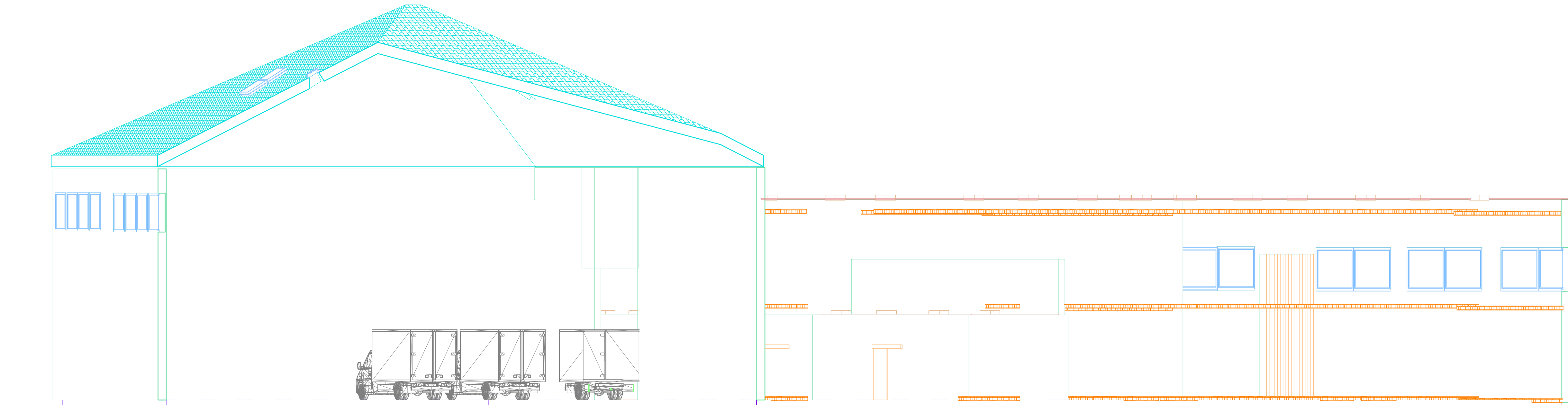
VISTA ALZADO: NORTE



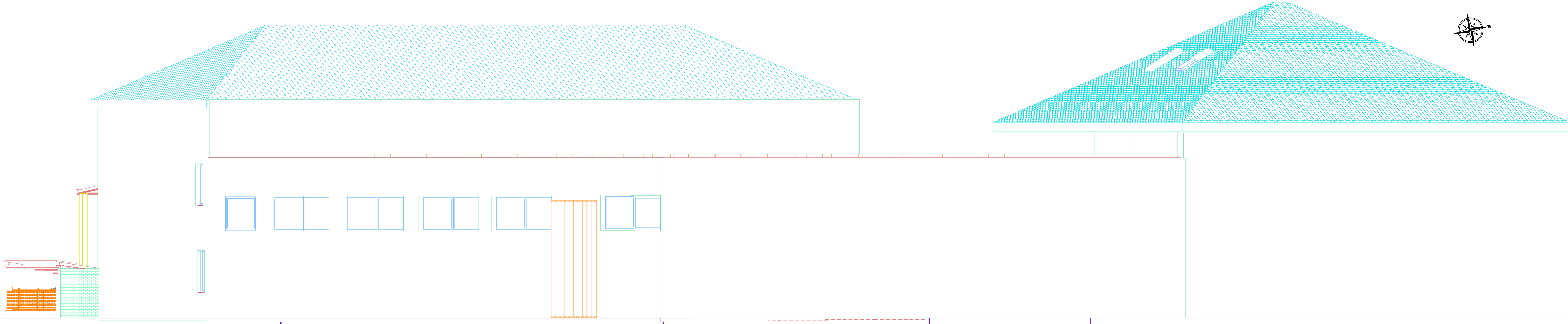
VISTA ALZADO: SUR

| | |
|---|---|
|  UNIVERSITAT POLITÈCNICA DE VALÈNCIA |  ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEODÉSICA CARTOGRÁFICA Y TOPOGRÁFICA |
| ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA EN GEOGRÁFICA CARTOGRÁFICA Y TOPOGRÁFICA | |
| TÍTULO DEL PLANO: PLANO EN ALZADOS. VISTAS: NORTE Y SUR. | FECHA: 05/08/2021 |
| AUTOR: JAVIER IGNACIO CHIVA JUANES | ESCALA: 1/100 |
| | NÚMERO DE PLANO: 03 |

PLANOS DE ALZADOS



VISTA ALZADO: ESTE



VISTA ALZADO: OESTE

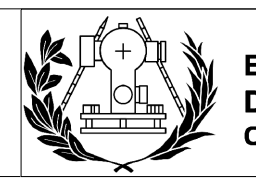


| | | | |
|---|--|--|--|
| UNIVERSITAT POLITÈCNICA DE VALÈNCIA | | ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA GEOGRÁFICA CARTOGRÁFICA Y TOPOGRÁFICA | |
| ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA EN GEOGRÁFICA CARTOGRÁFICA Y TOPOGRÁFICA | | | |
| TÍTULO DEL PLANO: PLANO EN ALZADOS. VISTAS: ESTE Y OESTE. | | FECHA: 05/08/2021 | |
| AUTOR: JAVIER IGNACIO CHIVA JUANES | | ESCALA: 1/100 | |
| | | NÚMERO DE PLANO: 02 | |

PLANO DE PLANTA 0



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA SUPERIOR
DE INGENIERÍA GEODÉSICA
CARTOGRÁFICA Y TOPOGRÁFICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA EN
GEOGRÁFICA CARTOGRÁFICA Y TOPOGRÁFICA

TÍTULO DEL PLANO: PLANO DE PLANTA 0.

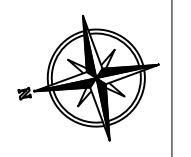
FECHA: 05/08/2021

ESCALA: 1/100

AUTOR: JAVIER IGNACIO CHIVA JUANES

NÚMERO DE PLANO: 04

PLANO DE PLANTA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA EN
GEOGRÁFICA CARTOGRÁFICA Y TOPOGRÁFICA

TÍTULO DEL PLANO: PLANO DE PLANTA.

FECHA: 05/08/2021

ESCALA: 1/100

AUTOR: JAVIER IGNACIO CHIVA JUANES

NÚMERO DE PLANO: 05

19. Anexos.

ESCÁNER LÁSER TRIMBLE TX6

RENDIMIENTO

Visión general
Principio de escaneado Espejo rotativo vertical sobre una base rotativa horizontal

Principio de alcance Tiempo de vuelo ultra rápido impulsado por la tecnología Trimble Lightning

Velocidad de escaneado² 500.000 puntos por segundo

Alcance máximo 80 m en la mayoría de las superficies con actualización opcional de 120 m

Ruido del alcance³ <2 mm en la mayoría de las superficies

Medición del alcance
Clase de láser 1, con protección de los ojos de conformidad con IEC EN60825-1

Longitud de onda láser 1,5 µm, invisible

Diámetro del rayo láser 6-10-34 mm a 10-30-100 m

Alcance mínimo 0,6 m

Alcance estándar máx. 80 m en superficies reflectantes 18-90%

Extensión del alcance⁴ 120 m en superficies reflectantes 18-90%

Ruido del alcance⁵ <2 mm en 2 m a 80 m en superficies reflectantes 18-90% en modo estándar

Error sistemático del alcance⁶ <2 mm reflectantes 18-90% en modo de extensión del alcance

Escaneado
Campo de visión 360° x 317°

Precisión angular⁵ 80 µrad

| Parámetros de escaneo | Vista preliminar | Nivel 1 | Nivel 2 | Nivel 3 |
|---|------------------|----------|----------|----------|
| Alcance Máx. ¹ | 80/120 m | 80/120 m | 80/120 m | 80/120 m |
| Duración del escaneado (minutos) ² | 02:00 | 03:00 | 05:00 | 19:00 |
| Espaciamento entre puntos a 10 m | 15,1 mm | — | — | — |
| Espaciamento entre puntos a 30 m | — | 22,6 mm | 11,3 mm | 5,7 mm |
| Espaciamento entre puntos a 300 m | — | — | — | — |
| Número de puntos | 8,7 Mpts | 34 Mpts | 138 Mpts | 555 Mpts |

ADQUISICIÓN DE IMÁGENES

Cámara HDR integrada 10 megapíxeles de resolución, campo de visión completo

Duración de la captura de imagen 1 para Estándar, 2 min para HDR

Kits de cámara externa disponibles para imágenes HDR y de alta resolución

OTRAS

Pantalla táctil TFT-LCD a color de 24 bits

Tamaño (mm) 93 (H) x 55,8 (V), equivalente a 4,3" diagonal

Resolución 800 x 480 (WVGA)

Resolución de luminancia 8 bits

Nivelación Burbuja externa, burbuja electrónica integrada

Compensador de doble eje Seleccionable Sí/No

Resolución 0,3"

Alcance ±5"

Precisión⁵ 1"

Almacenamiento de datos USB 3.0 Flash

Control remoto Operar con una tableta de Trimble u otro dispositivo móvil con WLAN, o con una PC o tableta con Windows 7 o superior mediante conexión de cable USB⁴

1 Actualización opcional que aumenta el alcance de 80 m a 120 m.
2 Velocidad de escaneo efectiva para obtener la calidad de escaneo óptima.
3 Tiempos de duración del escaneo en los modos de escaneo estándar.
4 El control remoto por cable requiere el cable USB opcional NF 237040/34.
5 Especificación dada como sigma 1.
6 A una distancia de 1,5 m a 100 m para un albedo >20%.

Las especificaciones están sujetas a cambio sin previo aviso.

ESPECIFICACIONES FÍSICAS

Dimensiones 335 mm de ancho x 386 mm de alto x 242 mm de profundidad

Peso 10,7 kg con plataforma nivelante y sin batería; 11,2 kg con plataforma nivelante y con batería

Suministro de alimentación eléctrica 76 mm de ancho x 43 mm de alto x 130 mm de profundidad; Peso: 0,66 kg

Dimensiones de la batería 89,2 mm de ancho x 20,1 mm de alto x 149,1 mm de profundidad

Peso de la batería 0,46 kg

Consumo de potencia 72 W

Tiempo de escaneado por batería >2 horas

Carcasa del instrumento 500 mm de ancho x 366 mm de alto x 625 mm de profundidad

ESPECIFICACIONES MEDIOAMBIENTALES

Rango de temperatura de funcionamiento (sin condensación atmosférica) -0 °C a +40 °C

Temperatura de almacenamiento -20 °C a +50 °C

Rango de humedad de funcionamiento Sin condensación

Condiciones de luminosidad Todas las condiciones interiores y exteriores en todo el alcance del escáner láser (sin limitaciones de luminosidad)

Clase de protección IP54



CLASS 1
LASER PRODUCT