

# Trabajo Fin de Grado

---

Sistema de reconocimiento de incidencias de obra y toma de decisiones automática

Work incident recognition and automatic decision-making system

Autor:

Ramón Campo Mata

Directora:

Dra. D<sup>a</sup>. Piedad Garrido

Universidad de Zaragoza

Escuela Universitaria Politécnica de Teruel

2020/2021

## Resumen

El objetivo de este Trabajo Final de Grado (TFG), orientado a la industria 4.0, es desarrollar una herramienta de lógica de negocio para una empresa de construcción en la que los Jefes de Obra puedan recoger, durante sus inspecciones, las incidencias de obra (preventa/postventa) en tiempo real, a través de una simple fotografía mediante un dispositivo móvil. Las fotografías serán procesadas para aplicar Inteligencia Artificial (IA), que reconocerá la incidencia (grietas, baldosas rotas, manchas), etiquetará la imagen y, tras una confirmación por parte del Jefe de Obra, enviará el aviso al gremio correspondiente para su tramitación/resolución. Para la identificación automática de incidencias se hará uso del servicio de "Custom Vision", dentro del apartado de "Visión" del paquete de Servicios Cognitivos de Azure, orientado a la personalización del reconocimiento de imágenes mediante el entrenamiento del modelo de visión artificial. La herramienta se apoyará sobre Office 365, utilizando distintas herramientas del Microsoft (además de Azure): se utilizará SharePoint para el almacenamiento y gestión de las incidencias y su metainformación, se hará uso de PowerApps para diseñar la aplicación móvil de gestión de incidencias (alta, edición, borrado y tramitación), una Función de Azure para la automatización del etiquetado de imágenes y la herramienta PowerAutomate para el diseño de los flujos de trabajo (aprobación y tramitación a gremio).

**Palabras clave:** SharePoint, Azure, APP de incidencias de obra, Custom Vision, Servicio Cognitivo.

## Abstract

The purpose of this Degree Final Project, oriented to industry 4.0, is to develop a business logic tool for a construction company in which Site Managers can collect, during their inspections, work incidents (pre-sale / post-sale) in real time, through a simple photograph using a mobile device. The photographs will be processed by means of Artificial Intelligence (AI) techniques, which will recognize the incident (cracks, broken tiles, stains), it will label the image and, after confirmation by training the artificial vision model. The tool will be based on Office 365, using different Microsoft tools (in addition to Azure): SharePoint will be used for the storage and management of incidents and their meta-information, PowerApps will be used to design the mobile application for incident management (high , editing, deleting and processing), an Azure Function for the automation of image labeling and the PowerAutomate for the design of workflows (approval and processing to the guild).

**Key words:** SharePoint, Azure, Work incidents APP, Custom Vision, Cognitive Service.

## Tabla de Contenidos

Resumen.....	1
Abstract .....	1
Tabla de Contenidos.....	2
1. Introducción y Objetivos .....	3
2. Estado del arte .....	4
2.1 SIS ERP Construcción .....	4
2.2 IB Building 365.....	4
2.3 Ekon Building.....	4
2.4 ConstruCloud.....	5
2.5 Telematel.....	5
2.6 Tabla comparativa .....	6
3. Análisis, Diseño e Implementación de la propuesta .....	7
3.1 Análisis.....	7
3.2 Diseño.....	8
3.2.1 Diseño de la Base de Datos .....	8
3.2.2 Aprovisionamiento para el sitio de SharePoint.....	9
3.2.3 Mockups de la APP para dispositivo móvil.....	10
3.3 Implementación .....	14
3.3.1 Tecnologías.....	14
3.3.2 Implementación del sitio de SharePoint Gestión de Incidencias.....	15
3.3.3 Implementación de la Power APP “Gestor Incidencias” .....	28
3.3.4 Creación del evento y de la función de Azure para el análisis automático de las imágenes cuando son agregadas a SharePoint.....	41
3.3.5 Entreno del servicio cognitivo .....	53
5. Conclusiones y Trabajo Futuro .....	54
6. Referencias bibliográficas .....	54
Anexo I – Diseño lógico de la Base de Datos.....	55
Anexo II – Prototipos de baja fidelidad .....	59
Anexo III – Licencia Software y Documental .....	68

## 1. Introducción y Objetivos

Este proyecto ya venía propuesto por la empresa Hiberus y se está desarrollando durante el transcurso del período de prácticas del autor en ella.

Hiberus hace años creó “Fundación hiberus” para fomentar la empleabilidad, vocaciones digitales y emprendimiento que impacta directamente en los Objetivos de Desarrollo Sostenible (ODS): Objetivo 4, Objetivo 9 y Objetivo 11.

El objetivo principal de este Trabajo Fin de Grado (TFG) surgió de pensar en que procesos de la construcción se podrían automatizar, para hacer más ágil el trabajo de un operario de obra a la hora de anotar, documentar y tramitar las incidencias de la misma. Dicho proceso de reportar la incidencia al gremio correspondiente (electricistas, fontanería, etc) se llevaría a cabo con metadatos, proporcionados por un servicio cognitivo, para las especificaciones de la incidencia y para la selección del gremio al que avisar.

El proceso que comprende este TFG va desde el momento en que un jefe de obra va a realizar una inspección, proporcionándole una APP para dispositivo móvil, en la que tendrá listadas las obras en las que su empresa le ha asignado trabajar.

Primero, el jefe de obra seleccionará la obra en la que vaya a realizar la inspección. Después establecerá el sitio de la obra en el que ha encontrado la incidencia, seleccionando los parámetros correspondientes: detectado en (preventiva/postventa), vivienda, escalera, planta y estancia, junto con un cuadro de texto en el que se pueden apuntar las anotaciones adicionales que el jefe de obra considere oportunas.

Una vez establecido el lugar en el que se ha detectado la incidencia, se da permiso para que el operario pueda agregar fotografías de la incidencia, tanto almacenadas en su dispositivo móvil, como tomadas en el mismo momento de la inspección desde la app.

Ya teniendo detallada la incidencia, el jefe de obra la puede registrar en la APP, y ésta la guardará en un sitio de SharePoint que almacenará la información y enviará las fotografías correspondientes al servicio cognitivo de Azure Custom Vision, (entrenado previamente para el reconocimiento de los tipos distintos de incidencias que se puedan dar), que se encargará de añadir las etiquetas a las imágenes en forma de metadatos. Este proceso automático del reconocimiento de las imágenes una vez que son agregadas a SharePoint y su etiquetado se realiza mediante la implementación de una función de Azure.

El sitio de SharePoint, una vez obtenga de vuelta las fotografías del servicio cognitivo, hará uso de PowerAutomate para tomar la decisión automática de a qué gremio, de los asociados a la obra, reportar la incidencia en base a estas etiquetas, para su posterior arreglo.

Este TFG consta de las siguientes secciones. En primer lugar, un apartado en el que se comentan la introducción y los objetivos, una segunda sección en la que se analizan un conjunto de aplicaciones existentes en el mercado, el análisis, diseño e implementación de la propuesta, las conclusiones y trabajo futuro y para finalizar, un conjunto de referencias bibliográficas y anexos de interés.

## 2. Estado del arte

En esta sección se van a analizar y comentar un conjunto de herramientas similares a la desarrollada, existentes en el mercado con la finalidad de poder concretar qué aportación innovadora se propone en este TFG.

Dichas herramientas son: SiS ERP Construcción, IB Building 365, Ekon Building, ConstruCloud y Telematel.

### 2.1 SiS ERP Construcción

Es un software de arquitectura abierta y flexible que ofrece gestión de almacén, pedidos, proyectos, recursos humanos, órdenes de compra y gestión financiera que utiliza aplicaciones cliente/servidor y servicios web tanto de internet como en forma de intranet, especialmente en tecnología .net (asp.net, ms framework 3.5).

Usa también aplicaciones de escritorio, servicios de Microsoft Windows y .net remoting y aplicaciones para dispositivos móviles, Microsoft Windows Mobile & Windows CE, conectividad GPS, WLAN, WIFI, GPRS y 3G.

En cuanto al modelo de datos utiliza bases de datos Microsoft SqlServer.

### 2.2 IB Building 365

IB Building unifica todas las áreas de gestión de la empresa en un único sistema de información. Se trata de un sistema flexible y adaptable, contiene programas de fácil uso, intuitivos y sencillos, es escalable, solución robusta, estable y eficaz, análisis de datos, Big Data, IoT y utiliza inteligencia artificial, blockchain, biometría, robótica colaborativa y geoposicionamiento inteligente.

IB Building es una solución vertical basada en Microsoft Dynamics Business Central que entre muchas otras funcionalidades integra servicios como Office 365 o Power BI, otras aplicaciones de Dynamics 365, PowerApps y soluciones disponibles en la AppSource de Microsoft.

### 2.3 Ekon Building

Recoge y detalla con exactitud la información de las obras como los recursos ocupados, materiales utilizados, maquinaria y terceros. En cuanto a la comunicación entre las áreas implicadas en los procesos de compra y expediciones, agrupa las compras, mejora los rapeles (descuentos que se concede a los clientes por alcanzar un consumo determinado durante un periodo de tiempo establecido) y controla la mercancía.

Realiza un seguimiento continuo y en tiempo real de cada proyecto u obra, así como asegura los márgenes de rentabilidad. Realiza valoraciones económicas completas, conoce los movimientos del material, recursos, consumos, compras, preferencias de los clientes, etc.

Este software adapta la tecnología al negocio y puede ser accedido desde cualquier dispositivo.

## **2.4 ConstruCloud**

Se trata de un software en la nube accesible para cualquier tipo de dispositivo. La solución es gratuita, sin pago de licencia ni de mantenimiento (versión con licencia de pago para empresas de mayor volumen).

Cuenta con una gestión documental (DropBox, Google Drive, etc.) ilimitada y comunicación interna e integración con whatsapp, así como geolocalización de proyectos.






ConstruCloud es un ERP que se encarga de llevar el seguimiento de gastos, emisión de partes desde la propia obra con smartphone, maquinaria, control de personal, cuentas bancarias y posee un foro propio que comunica a usuarios y desarrolladores, un sistema documental (ConstruWIKI) que tiene el mismo motor que Wikipedia, video tutoriales en el canal de Youtube Construcloud.es y un chat.

## **2.5 Telematel**

En cuanto al estudio de los proyectos y la presupuestación analiza y detalla los presupuestos con nueve niveles de detalle para definir la mano de obra y los materiales necesarios, crea librerías específicas para las obras, automatiza la importación de datos y guarda versiones de los presupuestos.

Respecto a la planificación de las obras organiza de manera lógica los trabajos, compra progresivamente el material con el avance real de la obra, asigna los trabajos a los operarios en base a su disponibilidad y conocimientos técnicos, documenta cada trabajo y o pone a disposición de los operarios y gestiona la subcontratación de mano de obra o maquinaria cuando sea necesario.

## 2.6 Tabla comparativa

Producto	Logotipo	URL	Tecnología
SiS ERP Construcción		<a href="https://www.sisgrupo.com/sis-erp-construccion/">https://www.sisgrupo.com/sis-erp-construccion/</a>	.net
IB Building		<a href="https://www.ibermatica365.com/ib-building/">https://www.ibermatica365.com/ib-building/</a>	Microsoft Dynamics Business Central
EkonBuilding		<a href="https://www.ekon.es/programas-software-erp-construccion-constructoras/">https://www.ekon.es/programas-software-erp-construccion-constructoras/</a>	-
ConstruCloud		<a href="https://construcloud.es/">https://construcloud.es/</a>	-
Telematel		<a href="https://www.telematel.com/">https://www.telematel.com/</a>	-

*Tabla 1. Comparativa entre las herramientas de construcción*

En definitiva, después de ver todas las ventajas y características que aportan estas herramientas al sector de la construcción, se ha pensado que hay una parte del proceso que se puede automatizar y ser mucho más rápida y efectiva. Y es esto mismo lo que aporta la solución planteada en este TFG, una herramienta capaz de registrar e identificar las incidencias de obra para automáticamente avisar al gremio correspondiente contratado para solucionarlas.

Como se puede observar en la Tabla 1, la mayoría de las tecnologías existentes en el mercado son propietarias y las empresas no comparten la información sobre ellas.

## 3. Análisis, Diseño e Implementación de la propuesta

### 3.1 Análisis

Para comenzar con el desarrollo de la solución se ha pensado en los requisitos que debe cumplir la APP móvil para recoger las incidencias. En cuanto al diseño se ha pensado que arquitectura tendrá el sitio de SharePoint para poder contener toda la información necesaria y relativa a las incidencias y un mockup orientativo para el diseño de la APP móvil. Finalmente en este punto se mostrará el proceso detallado de como se ha implementado toda la solución parte por parte.

Requisitos de la APP de gestión de incidencias [4] siguiendo las directrices de la norma IEEE 830:

- RQ1. El usuario habrá sido logeado previamente con su cuenta de Microsoft para acceder a la Power APP.
- RQ2. El usuario tendrá el listado de obras que le hayan sido asignadas, en el que podrá seleccionar una determinada.
- RQ3. El usuario tendrá el listado de incidencias de la obra seleccionada en el listado del RQ2. En el listado podrá seleccionar una incidencia.
- RQ4. El usuario dispondrá de una página de búsqueda de incidencias en la que podrá introducir el número de incidencia y más filtros para buscar una incidencia específica.
  - RQ4.1. Estos filtros serán: Detectado en (pre/postventa), escalera, planta y vivienda.
  - RQ4.2. Debajo de los filtros aparecerá el listado de incidencias que coinciden con los parámetros introducidos.
- RQ5. El usuario tendrá un botón con el que añadir una nueva incidencia a la obra seleccionada en el RQ2.
- RQ6. El usuario dispondrá de un formulario con los datos de la incidencia.
  - RQ6.1. Se puede acceder a este formulario seleccionando una incidencia RQ3 o añadiendo una nueva incidencia RQ5.
  - RQ6.2. El formulario dispondrá de un botón para editar los parámetros de la incidencia.
  - RQ6.3. Un botón para acceder a las fotografías de la propia incidencia.
  - RQ6.4. Un botón para eliminar la incidencia.
- RQ7. Los campos de la incidencia serán editables si se ha pulsado el botón RQ6.2.
  - RQ7.1. Los datos a rellenar son los siguientes: detectado en (pre/posventa), escalera, planta, vivienda y una caja de texto para añadir comentarios sobre la incidencia.
- RQ8. Cada incidencia dispondrá de un listado de imágenes. Este listado contendrá también un botón para añadir una nueva imagen.
  - RQ8.1. Junto a cada imagen aparecerá un botón para eliminarla.
- RQ9. A la hora de añadir una nueva imagen RQ8 se mostrará en pantalla lo que capte la cámara del dispositivo, un botón para tomar la imagen, otro para acceder a las fotografías del dispositivo, otro botón para desechar la fotografía tomada y un último botón para guardar la imagen.

### 3.2 Diseño

En lo que respecta a la fase de Diseño se ha enfocado desde tres puntos de vista: el diseño de la base de datos, el aprovisionamiento para el sitio de SharePoint y mockups de la APP móvil para el registro de incidencias.

#### 3.2.1 Diseño de la Base de Datos

Para la primera fase de diseño se han establecido las entidades con las que se va a trabajar, sus propiedades y las relaciones entre ellas

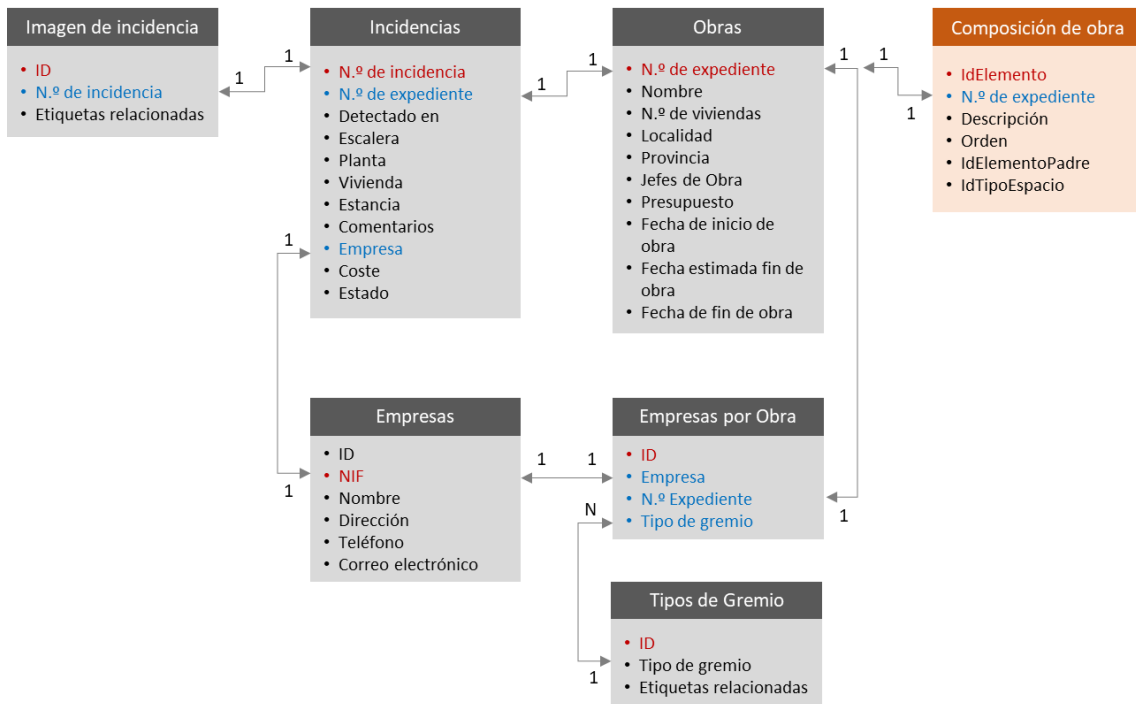
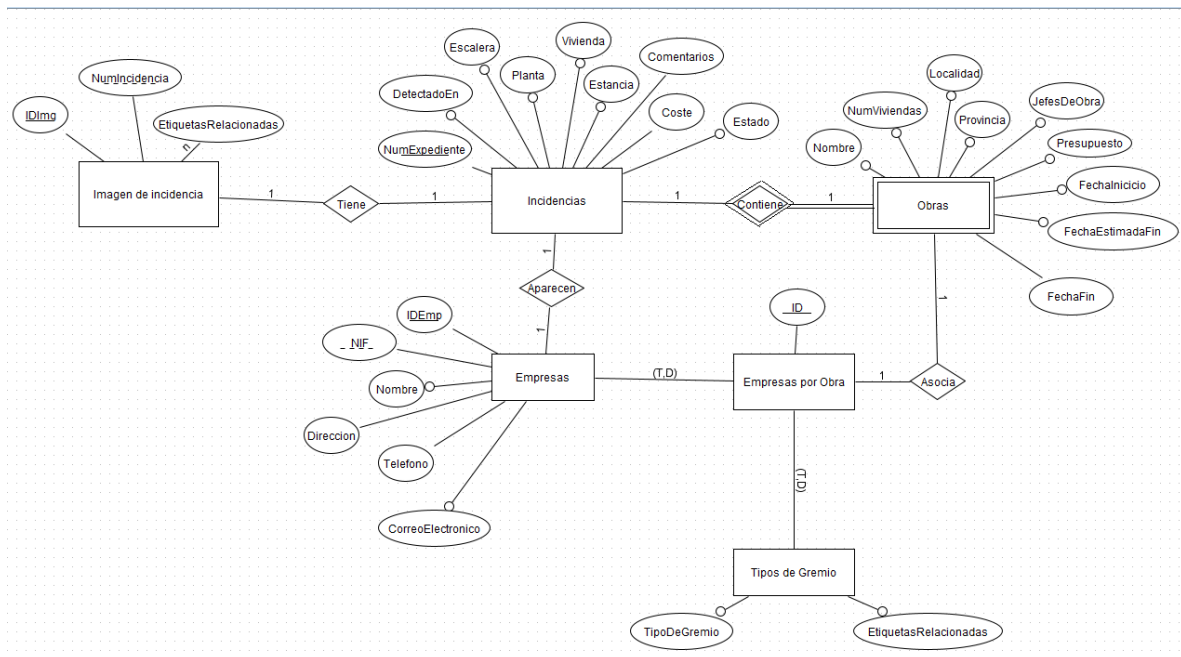


Figura 1. Esquema conceptual y físico BD

Para completar el diseño de la base de datos se aporta como Anexo I y realizado con DerEditor [6], herramienta desarrollada como TFG en la EUPT y utilizada en la asignatura de Bases de Datos, el diseño lógico.

### 3.2.2 Aprovisionamiento para el sitio de SharePoint

Para aprovisionar el sitio de SharePoint primero se han establecido las listas y bibliotecas necesarias. A continuación, para completar la fase de diseño se van a aportar una serie de prototipos de bajo nivel sobre distintas funcionalidades de la aplicación.

#### LISTAS/BIBLIOTECAS

TIPO	NOMBRE	URL	TIPO DE CONTENIDO
Lista	Obras	/Lists/Obras	Obra
Lista	Empresas	/Lists/Empresas	Empresa
Lista	Empresas por Obra	/Lists/EmpresasPorObra	Empresa por obra
Lista	Tipos de gremios	/Lists/TiposGremios	Tipo de gremio
Lista	Incidencias	/Lists/Incidencias	Incidencia
Biblioteca	Imágenes incidencias	/ImágenesIncidencias	Imagen de incidencia

Tabla 2. Listas y bibliotecas de SharePoint

Una vez definidas las listas y bibliotecas del sitio se han pasado a establecer que tipos de contenidos y columnas asociadas a estos son necesarios.

TIPO DE CONTENIDO	COLUMNA (DISPLAY NAME)	STATIC NAME	TIPO	OBSERVACIONES	OBLIGATORIO
Obra	Nombre	NombreObra	Una línea de texto		Sí
	Nº de expediente	NumExpediente	Una línea de texto	Autogenerado: YYYYXXXX - YYYY = año - XXXX = contador de 4 dígitos	Sí
	Nº de viviendas	NumViviendas	Número	Sin decimales	Sí
	Localidad	Localidad	Una línea de texto		Sí
	Provincia	Provincia	Elección	Provincias de España	Sí
	Jefes de Obra	JefesObra	Persona o grupo	Grupo específico por Obra para Jefes de Obra	Sí
	Presupuesto	Presupuesto	Moneda	2 decimales	Sí
	Fecha de inicio de obra	FechaInicioObra	Fecha	Solo fecha	Sí
	Fecha estimada fin de obra	FechaEstFin	Fecha	Solo fecha	Sí
	Fecha de fin de obra	FechaFinObra	Fecha	Solo fecha	No
Empresa	NIF	NIFEmpresa	Una línea de texto		Sí
	Nombre	NombreEmpresa	Una línea de texto		Sí
	Dirección	DirecciónEmpresa	Una línea de texto		No
	Teléfono	TelefonoContacto	Una línea de texto		No
	Correo electrónico	CorreoElectronico	Una línea de texto		Sí
Empresa por Obra	Empresa	EmpresaLookup	Búsqueda	Lista = Empresas; Campo = NIF El campo de búsqueda, además, traerá el Nombre de la Empresa	Sí
	Contrato	Contrato	Una línea de texto		Sí
	Tipo de gremio	TipoGremioLookup	Búsqueda	Lista = Tipos de gremios; Campo = Tipo de gremio Multivalor	Sí
	Obra	ObraLookup	Búsqueda	Lista = Obras; Campo = Nº expediente	Sí
Tipo de gremio	Tipo de gremio	TipoGremio	Una línea de texto		Sí
	Etiquetas relacionadas	EtiquetaRel	Metadatos administrados	Conjunto de términos IA	Sí
Incidencia	Nº de incidencia	NumIncidencia	Una línea de texto	Autogenerado: YYYYXXXXXX - YYYY = año - XXXXXX = contador de 6 dígitos	Sí
	Obra	ObraLookup	Búsqueda	Lista = Obras; Campo = Obra	Sí
	Detectado en	TipoIncidencia	Elección	Valores: Preventa / Postventa	Sí
	Escalera	EscaleraObra	Número	IdElemento	Sí
	Planta	PlantaObra	Número	IdElemento	Sí
	Vivienda	ViviendaObra	Número	IdElemento	Sí
	Estancia	EstanciaVivienda	Elección	Valores: - Baño principal - Baño general - Aseo - Dormitorio 1 - Dormitorio 2 - Dormitorio 3 - Comedor - Cocina - Hall - Terraza	Sí
	Comentarios	Comentarios	Varias líneas de texto	Sin formato	No
	Empresa	EmpresaLookup	Búsqueda	Lista = Empresas; Campo = NIF El campo de búsqueda, además, traerá el Nombre de la Empresa El desplegable solo mostrará aquellas empresas que tienen asociación con la Obra de la incidencia	No
	Coste	CosteIncidencia	Moneda	Dos decimales	No
Estado	EstadoIncidencia	Elección	Valores: Registrada / En tramitación / Tramitada Por defecto = Registrada	Sí	
Imagen de incidencia	Incidencia	IncidenciaLookup	Búsqueda	Lista = Incidencia; Campo = Nº de incidencia	Sí
	Etiquetas relacionadas	EtiquetaRel	Metadatos administrados	Conjunto de términos IA	No

Tabla 3. Columnas y Tipos de contenido de SharePoint

Se continuará con una serie de prototipos de baja fidelidad (en inglés mockup) de las distintas interfaces de la aplicación.

### 3.2.3 Mockups de la APP para dispositivo móvil

En el ANEXO II se puede consultar los prototipos de baja fidelidad de la APP para dispositivos móvil de:

- Pantalla de acceso a la aplicación
- Pantalla principal de obra
- Pantalla de búsqueda de incidencia

- d) Pantalla de alta de incidencia
- e) Pantalla de alta de incidencia
- f) Pantalla de edición de incidencia
- g) Pantalla de visualización de imágenes
- h) Pantalla de agregación de imágenes
- i) Pantalla de visualización de imagen

Este mockup de la APP móvil para el registro de incidencias de obra contiene:

**a) Pantalla de acceso a la aplicación**

- Cabecera (fija para todas las pantallas):
  - Muestra:
    - Botón Home: acceso directo a la pantalla principal
    - Nombre de la aplicación
    - Imagen del usuario conectado
- Listado de obras:
  - Muestra:
    - Obras a las que el usuario conectado tiene acceso (por estar dentro del grupo de Jefe de Obra correspondiente)

Al pulsar sobre cada elemento del listado de obras, se accede a la siguiente interfaz:

**b) Pantalla de acciones a realizar dentro de una Obra**

- Cabecera con nombre de obra (para localizar con qué obra se está trabajando)
- Muestra:
  - Nueva incidencia (alta de nueva incidencia)
  - Listado con las incidencias con estado “Registrada”, las incidencias que no se mostrarán aquí son las que tienen de estado: “En tramitación” porque se está comunicando al gremio, con lo cual, no deberían modificarse y las que tienen estado “Tramitada” porque ya está tramitada con el gremio, con lo cual, no debería modificarse). Es una forma de acceder directamente a las incidencias que sí que pueden modificarse.
    - Es posible acceder a cada incidencia, desde el botón “>” del listado -> ver figura 80
  - Botón para poder buscar cualquier incidencia -> ver figura 78

**c) Pantalla de búsqueda de incidencia dentro de una Obra**

- Cabecera con nombre de obra + botón “atrás” (<), que volverá a la figura 77
- Muestra:
  - Barra rellenable de texto para introducir el número de incidencia
  - Dropdowns para introducir los demás parámetros de búsqueda de incidencia
  - Listado con las incidencias abiertas (registradas) resultantes de aplicar los filtros de búsqueda introducidos
    - Es posible acceder a cada incidencia, desde el botón “>” del listado -> ver figura 80

**d) Pantalla de alta de incidencia**

- Cabecera con nombre de obra (para localizar con qué obra se está trabajando) + botón “atrás” (<), que volverá a la figura 77
- La pantalla vendrá con los datos precargados de Detectado en, escalera, planta y vivienda de la última incidencia registrada en la obra
- Acciones:
  - Aunque los botones de acceso a fotografías y eliminar sean visibles, estarán deshabilitados hasta que se termine de crear la incidencia
- Muestra:
  - Información de la incidencia con los campos en edición
  - Botón Guardar → se irá a la pantalla de la figura 80
  - Botón Cancelar → se volverá a la pantalla de la figura 77

**e) Pantalla de información de incidencia**

- Cabecera con nombre de obra (para localizar con qué obra se está trabajando) + botón “atrás” (<), que volverá a la figura 77
- Acciones:
  - Si la incidencia es editable = estado “Registrada)

- Edición de incidencia (editaría los datos de la pantalla) -> ver figura 81
  - Agregar fotografía -> ver figura 82
- Eliminar incidencia
  - Eliminaría su información y las fotografías asociadas
- Copiar incidencia
  - Creará una nueva incidencia para la misma Escalera / Planta / Vivienda / Estancia
  - Este botón llevará -> ver figura 79 con todos los desplegables precargados, excepto la caja de comentarios
- Muestra:
  - Información de la incidencia con los campos en no edición

**f) Pantalla de edición de incidencia**

- Cabecera con nombre de obra (para localizar con qué obra se está trabajando) + botón “atrás” (<), que volverá a la IMAGEN 2
- Acciones:
  - Aunque los botones de acceso a fotografías y eliminar sean visibles, estarán deshabilitados hasta que se termine de editar la incidencia
- Muestra:
  - Información de la incidencia con los campos en edición
  - Botón Guardar + Botón Cancelar → al pulsar ambos botones, se volverá a la pantalla de la figura 80

**g) Pantalla de visualización de imágenes**

- Cabecera con nombre de obra (para localizar con qué obra se está trabajando) + botón “atrás” (<), que volverá a la figura 80
- Muestra:
  - Botón para agregar imagen -> ver figura 83
  - Fotografías de la incidencia que se está visualizando

- Scroll hacia abajo, para ver todas las imágenes agregadas a la incidencia
- Cada una tiene un icono de una papelera para eliminar la propia imagen

#### **h) Pantalla de agregación de imágenes**

- Cabecera con nombre de obra (para localizar con qué obra se está trabajando) + botón “atrás” (<), que volverá a la figura 82
- Muestra:
  - Cámara del dispositivo
    - Pulsar en el recuadro de la imagen para tomar la fotografía -> ver figura 84
  - Botón galería para seleccionar una imagen de la galería del dispositivo, una vez seleccionada -> ver figura 84

#### **i) Pantalla de visualización de imagen**

- Cabecera con nombre de obra (para localizar con qué obra se está trabajando) + botón “atrás” (<), que volverá a la figura 83
- Muestra:
  - Imagen tomada o seleccionada en la galería
  - Botón para guardar la fotografía -> ver figura 82
  - Botón de cancelar para descartar la fotografía -> ver figura 83

## **3.3 Implementación**

### **3.3.1 Tecnologías**

En cuanto a las tecnologías usadas, para la implementación de la propuesta, se pasará a dividir por las distintas partes que tiene:

1. Sitio de SharePoint
2. Power APP
3. Servicios Cognitivos (Custom Vision)
4. Función de Azure y automatización

Parte de la implementación	Tecnologías
Sitio de SharePoint	-ID: Visual Studio 2017  -Lenguaje: C#  -Plataforma de destino: .NET Framework 4.6.1  -Consultas de datos: SharePoint Online Client Browser
Power App	-ID: El propio estudio online de PowerApp - <a href="https://eu.create.powerapps.com/studio/">https://eu.create.powerapps.com/studio/</a>  -Lenguaje: Power FX, Lenguaje de bajo nivel, basado en la creación de fórmulas de Excell
Servicios Cognitivos	API de Azure Cognitive Services
Función de Azure y automatización	-ID: Visual Studio 2019  -Lenguaje: C#  -Librería para las funciones de Azure: azure functions core tools 3.0.3442  -Plataforma de destino: .NET Core 3.1  -Flujo automático en SharePoint: Power Automate

### 3.3.2 Implementación del sitio de SharePoint Gestión de Incidencias

Lo primero que se ha realizado es crear un sitio de grupo de SharePoint con el nombre de "Gestión de Incidencias" con la URL: <https://rcampohiberus.sharepoint.com/sites/GestionIncidencias>.

Una vez creado el sitio se ha pasado a crear un proyecto nuevo de aplicación consola en C# usando la plataforma de .NET Framework 4.6.1, con el nombre de Aprovisionamiento.TFG.

La estructura de proyecto que se ha establecido es la siguiente:

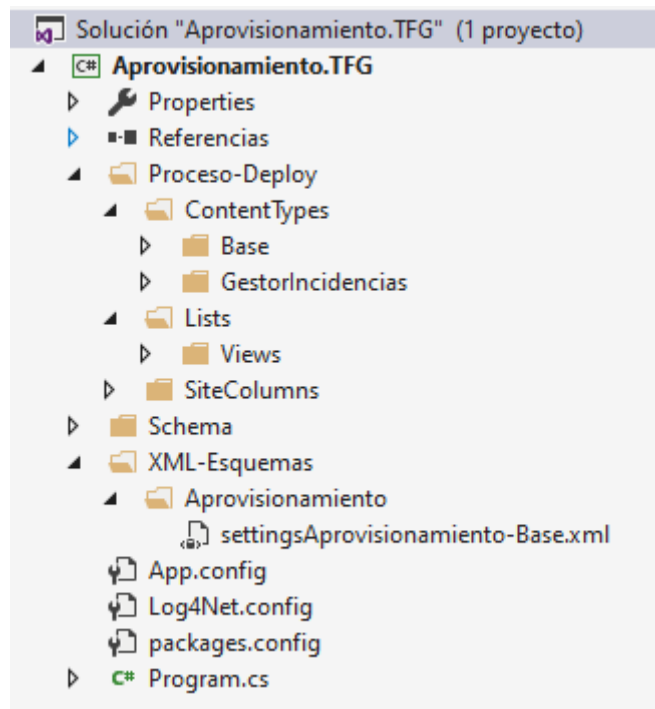


Figura 2. Estructura del Proyecto de Aprovisionamiento

-Program.cs: Contiene la clase principal de la solución que autenticará contra SP y iniciará el proceso de Aprovisionamiento.

-Proceso-Deploy

-ContentTypes

-Base: Contiene los ficheros XML que definirán los tipos de contenido base de los que heredarán el resto.

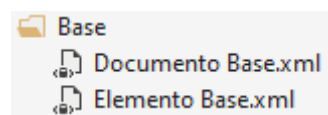


Figura 3. Estructura del Proyecto de Aprovisionamiento, Tipos de contenido Base

-GestorIncidencias: XML que definen todos los tipos de contenido que se utilizarán en el sitio.

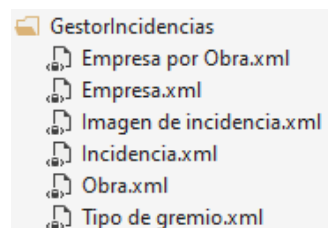


Figura 4. Estructura del Proyecto de Aprovisionamiento, Tipos de contenido Gestor de Incidencias

-Lists

-Views: XML que definen la configuración de las vistas y las columnas que se van a mostrar en las distintas listas y bibliotecas de documentos.

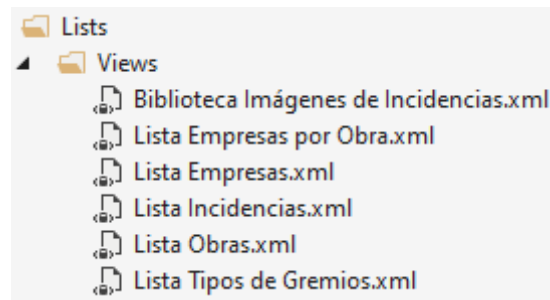


Figura 5. Estructura del Proyecto de Aprovisionamiento, Vistas de las listas

-SiteColumns: XML que definen las columnas de sitio que se utilizarán en el sitio de SharePoint.

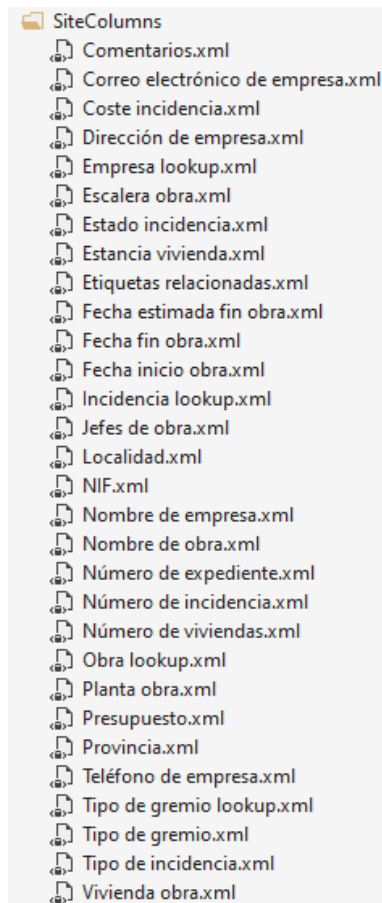


Figura 6. Estructura del Proyecto de Aprovisionamiento, Columnas de sitio

-XML-Esquemas

-Aprovisionamiento

-settingsAprovisionamiento-Base.xml: Aprovisiona las características, el almacén de términos, las columnas, los tipos de contenido y las listas del sitio.

En el caso de que se necesitara se podrían definir aquí también subsitios y carpetas.

Lo primero que se ha pasado a implementar del proyecto es su clase principal Program.cs, en cuyo Main, se ha desarrollado la autenticación contra el sitio de Gestión de Incidencias y se pone en marcha el proceso de aprovisionamiento.

Para la autenticación es necesario el nombre del Tenant en el cual está alojado el sitio que se quiere aprovisionar, el nombre del usuario con el que se va a realizar la autenticación, su contraseña y la URL propia del sitio, que será “/sites/GestionIncidencias”.

```
1 using Aprovisionamiento.0365.CORE.Configuracion;
2 using System.IO;
3 using CORE = Aprovisionamiento.0365.CORE;
4
5 namespace TFG.Aprovisionamiento
6 {
7     class Program
8     {
9         private const string TENANT_NAME = "tenantName";
10        private const string USER_NAME = "user@tenantName.onmicrosoft.com";
11        private const string USER_PASSWORD = "*****";
12        private const string SITE_URL = "/sites/GestionIncidencias";
13
14        static void Main(string[] args)
15        {
16            string relativeXMLPath = "XML-Esquemas\\Aprovisionamiento\\
17                SettingsAprovisionamiento-Base.xml";
18
19            //Ruta de trabajo actual (La que contiene la carpeta Proceso-
20            Deploy)
21            string workingDirectory = Directory.GetParent
22                (Directory.GetCurrentDirectory()).Parent.FullName;
23
24            //Fichero de configuracion de la ejecución
25            DeployConfiguration config = new DeployConfiguration()
26            {
27                XmlPath = string.Format("{0}\\{1}", workingDirectory,
28                    relativeXMLPath),
29                ArtifactsPath = workingDirectory,
30                Log4NetConfig = string.Format("{0}\\Log4Net.config",
31                    workingDirectory),
32                TenantUrl = string.Format("https://{0}.sharepoint.com",
33                    TENANT_NAME),
34                SiteUrl = SITE_URL,
35                User = USER_NAME,
36                Password = USER_PASSWORD
37            };
38
39            CORE.Aprovisionamiento aprovisionamiento = new
40                CORE.Aprovisionamiento(config);
41
42            //Inicio del despliegue
43            aprovisionamiento.Deploy();
44        }
45    }
46 }
```

Figura 7. Clase Program del Proyecto de Aprovisionamiento

Después se han pasado a implementar las columnas de sitio. Cabe destacar que la forma de implementar las columnas es muy similar en todos los tipos de columnas (texto, número, elección, etc), excepto en dos tipos que son las columnas de búsqueda (LookUp) y de Taxonomía (Taxonomy). Por lo que se ha decidido mostrar un ejemplo de cada una de las tres para entender su funcionamiento.

-Nombre de Obra.xml: Columna de texto que contiene el nombre de la Obra.

-ID: GUID propio de la columna generado desde el propio ID.

-Type: Tipo de columna.

-Name: Nombre de la columna (se utiliza el mismo nombre que StaticName).

-StaticName: Nombre interno que tendrá la columna.

-DisplayName: Nombre que se mostrará al usuario de la columna.

-Group: Nombre del grupo de columnas personalizadas en el que se va a incluir la columna

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <Elements xmlns="http://schemas.microsoft.com/sharepoint/">
3   <Field
4     ID="{90A0F552-AA3E-4DDB-AD0D-14FF258C06CA}"
5     Type="Text"
6     Name="NombreDeObra"
7     StaticName="NombreDeObra"
8     DisplayName="Nombre de obra"
9     Group="Incidencias">
10  </Field>
11 </Elements>
```

Figura 8. XML de la Columna Nombre de Obra del Proyecto de Aprovechamiento

-Empresa lookup.xml: Columna de búsqueda que se utilizará para referenciar una obra desde otra lista diferente.

Se van a explicar las propiedades distintas que no están contenidas en la columna de texto anterior.

-List: URL de la lista de la que se va a traer la información.

-WebId: subsitio (si la lista está contenida en uno) en el que se encuentra la lista.

-ShowField: Campo a mostrar de la lista en la que se busca.

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <Elements xmlns="http://schemas.microsoft.com/sharepoint/">
3   <Field
4     ID="{FC55C998-E775-4838-9592-74E5DDA42073}"
5     Type="Lookup"
6     Name="EmpresaLookup"
7     StaticName="EmpresaLookup"
8     DisplayName="Empresa"
9     List="Lists/Empresas"
10    WebId=""
11    ShowField="NIF"
12    Group="Incidencias">
13  </Field>
14 </Elements>

```

Figura 9. XML de la Columna EmpresaLookup del Proyecto de Aprovisionamiento

-Etiquetas relacionadas.xml: Columna de taxonomía para etiquetar las imágenes y así clasificarlas.

-ShowField: Lenguaje que se utilizará para los términos.

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <Elements xmlns="http://schemas.microsoft.com/sharepoint/">
3   <Field
4     ID="{74E6864E-9FD7-44CE-B071-77DB73605807}"
5     Type="TaxonomyFieldTypeMulti"
6     Name="EtiquetaRel"
7     StaticName="EtiquetaRel"
8     DisplayName="Etiquetas relacionadas"
9     ShowField="Term3082"
10    Mult="TRUE"
11    Group="Incidencias">
12  </Field>
13 </Elements>

```

Figura 10. XML de la Columna Etiquetas Relacionadas del Proyecto de Aprovisionamiento

Una vez definidas las columnas ya se puede pasar a implementar los tipos de contenido que van a contener las columnas que necesitan.

Para entender la definición de los tipos de contenido y su herencia se va a definir la implementación del elemento base y del tipo de contenido de obra que heredará del base.

-Elemento Base.xml: Tipo de contenido del que heredarán el resto de los elementos de lista.

-ID: 0x (System) + 01 (Item) + 00 + C073D13C970E478DAE544C9F3A8AE4F8 (id propio)

-Otros Tipos de contenido base: 0101 (Document), 0102 (Event)

-Name: Nombre del tipo de contenido

-Group: Nombre del grupo de tipos de contenidos personalizados en el que se va a incluir.

-Description: Texto de descripción del tipo de contenido.

-Inherits: Booleano que define si el tipo de contenido hereda de otro o no. Los elementos base de SharePoint no cuentan como herencia para tener en cuenta en esta propiedad, por lo que en este caso estarán en False ya que solamente hereda de Item.

-Version: Versión del tipo de contenido.

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <Elements xmlns="http://schemas.microsoft.com/sharepoint/">
3   <ContentType ID="0x0100C073D13C970E478DAE544C9F3A8AE4F8"
4     Name="Elemento Base"
5     Group="Tipos de contenido TFG"
6     Description=""
7     Inherits="False"
8     Version="0">
9     <FieldRefs>
10      <RemoveFieldRef ID="{fa564e0f-0c70-4ab9-b863-0177e6ddd247}"
11        Name="Title" />
12    </FieldRefs>
13  </ContentType>
14 </Elements>
```

Figura 11. XML del tipo de contenido Elemento Base del Proyecto de Aprovisionamiento

-Obra.xml: Tipo de contenido que establece los datos que va a contener una obra.

-ID: Al contrario que el tipo de contenido de Elemento Base, este elemento sí que hereda de otro, por lo que es necesario añadir al ID, el id del elemento del que hereda. 0x (System) + 01 (Item) + 00 + C073D13C970E478DAE544C9F3A8AE4F8 (id elemento base) + 00 + ea114dc6b397407e99b95f7019df3ff8 (id propio).

-Inherits: En este caso es necesario establecer esta propiedad a true.

-<FieldRefs>: Columnas que va a contener el tipo de contenido.

-ID: id de la columna

-Required: Si el campo es obligatorio a la hora de añadir el elemento a una lista.

-Name: Nombre de la columna.

-DisplayName: Nombre que se mostrará de la columna.

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <Elements xmlns="http://schemas.microsoft.com/sharepoint/">
3   <ContentType
4     ID="0x0100C073D13C970E478DAE544C9F3A8AE4F800ea114dc6b397407e99b95f7019df3ff8"
5     Name="Obra"
6     Group="Tipos de contenido TFG"
7     Description=""
8     Inherits="TRUE"
9     Version="0">
10    <FieldRefs>
11      <FieldRef ID="{90A0F552-AA3E-4DDB-AD0D-14FF258C06CA}" Required="TRUE"
12        Name="NombreDeObra" DisplayName="Nombre de obra"/>
13      <FieldRef ID="{C4804623-9801-43CC-AE50-D5AC16FC69BA}" Required="TRUE"
14        Name="NumeroDeExpediente" DisplayName="Número de expediente"/>
15      <FieldRef ID="{8EAB41A4-E5BC-4F5C-8988-8781972CD28A}" Required="TRUE"
16        Name="NumeroDeViviendas" DisplayName="Número de viviendas"/>
17      <FieldRef ID="{2387E59B-04FD-47EE-BED1-2D52AC9AB283}" Required="TRUE"
18        Name="Localidad" DisplayName="Localidad"/>
19      <FieldRef ID="{E0B2BDCF-331E-4CE1-A162-B70E68532EE3}" Required="TRUE"
20        Name="Provincia" DisplayName="Provincia"/>
21      <FieldRef ID="{894D068A-0613-4911-B5A2-EE60208CCE35}" Required="TRUE"
22        Name="JefesDeObra" DisplayName="Jefes de obra"/>
23      <FieldRef ID="{6B12FD23-EE00-43AB-B4F7-E10D7B0B6BCF}" Required="TRUE"
24        Name="Presupuesto" DisplayName="Presupuesto"/>
25      <FieldRef ID="{0E44E8AD-3339-47F6-A95F-FB585453A860}" Required="TRUE"
26        Name="FechaInicioObra" DisplayName="Fecha de inicio de obra"/>
27      <FieldRef ID="{14040566-37E2-44D2-BF16-F5608F3E295F}" Required="TRUE"
28        Name="FechaEstimadaFinObra" DisplayName="Fecha estimada fin de obra"/>
29      <FieldRef ID="{B1121D17-B8D3-4B09-9DF7-7E7673BA4895}" Required="FALSE"
30        Name="FechaFinObra" DisplayName="Fecha de fin de obra"/>
31    </FieldRefs>
32  </ContentType>
33 </Elements>

```

Figura 12. XML del tipo de contenido Obra del Proyecto de Aprovisionamiento

Lo siguiente ha sido definir las vistas que se van a utilizar, para visualizar los datos de las listas y bibliotecas del sitio de SharePoint.

-Lista Obras.xml: Define la configuración de la vista de la lista de obras.

-Type: Tipo de colección, en este caso al tratarse de una lista se utilizará GenericList.

-Name: Nombre de la vista.

-ViewTypeKind: Establece el tipo de vista, 1 establece el tipo de vista Standar usado en SharePoint.

-Otros tipos de vista son: Datasheet, Calendar, Gantt y Custom.

-ViewFields: Columnas que se van a mostrar en la lista.

-RowLimit: Límite de filas a mostrar.

-DefaultView: Si este parámetro está a true, establece la vista como por defecto de la lista.

-Paged: Establece si la vista de lista es una vista de página.

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <ListViews>
3   <List Type="GenericList">
4     <View
5       Name="Lista Obras"
6       ViewTypeKind="1"
7
8       ViewFields="NombreDeObra,NumeroDeExpediente,NumeroDeViviendas,Localidad,Provincia,
9       JefesDeObra,Presupuesto,FechaInicioObra,FechaEstimadaFinObra,FechaFinObra"
10      RowLimit="30"
11      DefaultView="true"
12      Paged="true">
13     <ViewQuery></ViewQuery>
14   </View>
15 </List>
16 </ListViews>

```

Figura 13. XML de la vista de la lista de Obras del Proyecto de Aproveccionamiento

-Biblioteca Imágenes de incidencias.xml: La creación de la vista de una biblioteca es muy similar a la de las listas, las especificaciones que cambian son las siguientes:

- Type: En este caso se usará DocumentLibrary para establecer que la vista es de una biblioteca de documentos.

- ContentTypeName: Nombre del tipo de contenido que va a mostrar la vista.

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <ListViews>
3   <List Type="DocumentLibrary">
4     <View
5       Name="Biblioteca Imágenes de Incidencias"
6       ViewTypeKind="1"
7       ViewFields="IncidenciaLookup,EtiquetaRel"
8       RowLimit="30"
9       DefaultView="true"
10      ContentTypeName="Imagen de incidencia"
11      Paged="true">
12     <ViewQuery></ViewQuery>
13   </View>
14 </List>
15 </ListViews>

```

Figura 14. XML de la vista de la biblioteca de Imágenes de Incidencias del Proyecto de Aproveccionamiento

Para terminar con el aprovisionamiento, se va a pasar a explicar la clase que contiene la configuración global y agrupa todo lo que se va a aprovisionar en el sitio.

- settingsAproveccionamientoBase.xml: Este fichero se divide en varias secciones, las cuales son:

```

1 <?xml version="1.0" encoding="utf-8" ?>
2 <Tenant xmlns="my://HiberusAprovisionamiento" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="my://HiberusAprovisionamie
3 <Credentials Account="contoso@contoso.onmicrosoft.com" Password="sSpP2018" SPOVersion="SPOnline"/>
4 <Site Title="Intranet" Url="/sites/pre_intranet" LCID="1033" TimeZoneId="13" Description="Intranet" CurrentResourceUsage="300" StorageUsage="3" :
5 <Features>...</Features>
6 <TermStore Provision="false">...</TermStore>
7 <Webs>
8 <RootWeb>
9 <SiteColumns SourcePath="Proceso-Deploy\" Provision="true">...</SiteColumns>
10 <ContentTypes SourcePath="Proceso-Deploy\" Provision="true">...</ContentTypes>
11 <Lists Provision="true">...</Lists>
12 </RootWeb>
13 </Webs>
14 </Site>
15 </Tenant>

```

Figura 15. XML de la estructura de settingsAprovisionamientoBase del Proyecto de Aprovisionamiento

-<Features>: En esta sección se introducen las características que se quieren activar en el sitio por su id. En este caso se activará solo la característica de Taxonomía para permitir el uso de taxonomías en columnas.

```

5 <Features>
6 <Feature ID="73EF14B1-13A9-416b-A9B5-ECECA2B0604C"/>
7 </Features>

```

Figura 16. XML de Características de settingsAprovisionamientoBase del Proyecto de Aprovisionamiento

-<TermStore>: Aquí se definirá el grupo de conjunto de términos <TermGroup>, el conjunto de términos para el etiquetado de imágenes de incidencias <TermSet> y los términos que irán dentro del conjunto <Term>. Cabe destacar que hay que añadir, tanto al conjunto como a los términos, el idioma al cual pertenecen. En este caso será LCID="1033" para asociarlos al español.

```

8 <TermStore Provision="false">
9 <TermGroups>
10 <TermGroup GroupName="TFG">
11 <TermSet LCID="1033" Name="Etiqueta">
12 <Term LCID="1033" Name="Grieta"/>
13 <Term LCID="1033" Name="Agujero"/>
14 <Term LCID="1033" Name="Humedad"/>
15 <Term LCID="1033" Name="Tuberia"/>
16 <Term LCID="1033" Name="Cable"/>
17 <Term LCID="1033" Name="Parquet"/>
18 <Term LCID="1033" Name="Persiana"/>
19 </TermSet>
20 </TermGroup>
21 </TermGroups>
22 </TermStore>

```

Figura 17. XML del Almacén de Términos de settingsAprovisionamientoBase del Proyecto de Aprovisionamiento

-<SiteColumns>: Aquí se establecen las columnas a aprovisionar en el sitio. Todos los tipos de columnas, excepto las de búsqueda y taxonomías, se referencian en esta región del XML solamente con el nombre de su fichero XML.

```

56      <!--SiteColumns-->
57
58      <Field SourceXML="Jefes de obra.xml" />
59      <Field SourceXML="Comentarios.xml" />
60      <Field SourceXML="Correo electrónico de empresa.xml" />
61      <Field SourceXML="Coste incidencia.xml" />
62      <Field SourceXML="Dirección de empresa.xml" />
63      <Field SourceXML="Escalera obra.xml" />
64      <Field SourceXML="Estado incidencia.xml" />
65      <Field SourceXML="Estancia vivienda.xml" />
66      <Field SourceXML="Fecha estimada fin obra.xml" />
67      <Field SourceXML="Fecha fin obra.xml" />
68      <Field SourceXML="Fecha inicio obra.xml" />
69      <Field SourceXML="Localidad.xml" />
70      <Field SourceXML="NIF.xml" />
71      <Field SourceXML="Nombre de empresa.xml" />
72      <Field SourceXML="Nombre de obra.xml" />
73      <Field SourceXML="Número de expediente.xml" />
74      <Field SourceXML="Número de incidencia.xml" />
75      <Field SourceXML="Número de viviendas.xml" />
76      <Field SourceXML="Planta obra.xml" />
77      <Field SourceXML="Presupuesto.xml" />
78      <Field SourceXML="Provincia.xml" />
79      <Field SourceXML="Teléfono de empresa.xml" />
80      <Field SourceXML="Tipo de gremio.xml" />
81      <Field SourceXML="Tipo de incidencia.xml" />
82      <Field SourceXML="Vivienda obra.xml" />
--

```

Figura 18. XML de las Columnas generales de settingsAprovisionamientoBase del Proyecto de Aprovisionamiento

En cuanto a las columnas de Búsqueda, aparte del nombre de su XML, hay que añadir la información de la lista a la que hacen referencia.

- Name: Nombre de la lista.
- TemplateType: Tipo de lista.
- UrlPath: URL de la lista.
- EnableContentTypes: Si permite tipos de contenido, se establecerá a true, ya que se están utilizando tipos de contenido personalizados.
- EnableAttachments: Si se permiten añadir adjuntos.

```

27      <!--Lookup-->
28
29      <Field SourceXML="Empresa lookup.xml">
30        <RequiredStructureHierarchy>
31          <RequiredList Name="Empresas" TemplateType="GenericList"
32            EnableVersioning="false" UrlPath="Lists/Empresas"
33            EnableContentTypes="true" EnableAttachments="false"/>
34        </RequiredStructureHierarchy>
35      </Field>
36      <Field SourceXML="Incidencia lookup.xml">
37        <RequiredStructureHierarchy>
38          <RequiredList Name="Incidencias" TemplateType="GenericList"
39            EnableVersioning="false" UrlPath="Lists/Incidencias"
40            EnableContentTypes="true" EnableAttachments="false"/>
41        </RequiredStructureHierarchy>
42      </Field>
43      <Field SourceXML="Obra lookup.xml">
44        <RequiredStructureHierarchy>
45          <RequiredList Name="Obras" TemplateType="GenericList"
46            EnableVersioning="false" UrlPath="Lists/Obras"
47            EnableContentTypes="true" EnableAttachments="false"/>
48        </RequiredStructureHierarchy>
49      </Field>
50      <Field SourceXML="Tipo de gremio lookup.xml">
51        <RequiredStructureHierarchy>
52          <RequiredList Name="Tipos de gremios" TemplateType="GenericList"
53            EnableVersioning="false" UrlPath="Lists/TiposGremios"
54            EnableContentTypes="true" EnableAttachments="false"/>
55        </RequiredStructureHierarchy>
56      </Field>

```

Figura 19. XML de las Columnas de Búsqueda de settingsAprovisionamientoBase del Proyecto de Aprovisionamiento

Y por último, las columnas de Taxonomías en las cuales hay que establecer el conjunto de términos que contiene y el nombre del grupo al que pertenece ese conjunto.

```

51      <!--Taxonomía-->
52
53      <Field SourceXML="Etiquetas relacionadas.xml" TermGroupName="TFG"
54        TermSetName="Etiqueta"/>

```

Figura 20. XML de las Columnas de Metadatos Administrados de settingsAprovisionamientoBase del Proyecto de Aprovisionamiento

-<ContentTypes>: En esta región se establecen los tipos de contenido que se van a aprovisionar mediante las rutas de sus ficheros XML, correspondientes dentro de la solución.

```

85      <ContentTypes SourcePath="Proceso-Deploy\ContentTypes"
86        Provision="true">
87        <!--Base-->
88        <ContentType SourceXML="Base\Elemento Base.xml"/>
89        <ContentType SourceXML="Base\Documento Base.xml"/>
90
91        <!--Gestor Incidencias-->
92        <ContentType SourceXML="GestorIncidencias\Empresa por Obra.xml"/>
93        <ContentType SourceXML="GestorIncidencias\Empresa.xml"/>
94        <ContentType SourceXML="GestorIncidencias\Imagen de incidencia.xml"/>
95        <ContentType SourceXML="GestorIncidencias\Incidencia.xml"/>
96        <ContentType SourceXML="GestorIncidencias\Obra.xml"/>
97        <ContentType SourceXML="GestorIncidencias\Tipo de gremio.xml"/>

```

Figura 21. XML de los Tipos de Contenido de settingsAprovisionamientoBase del Proyecto de Aprovisionamiento

-<Lists>: En la última parte se definen las Listas y bibliotecas a incluir en el sitio, las cuales tienen las siguientes propiedades:

-Name: Nombre de la lista.

-UrlPath: URL en la que estará contenida la lista dentro del sitio.

-TemplateType: Tipo de lista.

-EnableContentTypes: Si permite el uso de tipos de contenido personalizado.

-ViewSourcePath: Ruta dentro de la solución donde se encuentra el XML correspondiente a la vista de la lista.

```
99      <Lists Provision="true">
100      <!--Listas-->
101      <List Name="Obras" UrlPath="Lists\Obras" TemplateType="GenericList"
102            EnableContentTypes="true" ViewSourcePath="Proceso-Deploy/Lists/
103            Views/Lista obras.xml">
104            <ContentTypes>
105              <RemoveContentType Name="Elemento"/>
106              <ContentType Name="Obra"/>
107            </ContentTypes>
108          </List>
109          <List Name="Empresas" UrlPath="Lists\Empresas"
110            TemplateType="GenericList" EnableContentTypes="true"
111            ViewSourcePath="Proceso-Deploy/Lists/Views/Lista Empresas.xml">
112            <ContentTypes>
113              <RemoveContentType Name="Elemento"/>
114              <ContentType Name="Empresa"/>
115            </ContentTypes>
116          </List>
117          <List Name="Empresas por Obra" UrlPath="Lists\EmpresasPorObra"
118            TemplateType="GenericList" EnableContentTypes="true"
119            ViewSourcePath="Proceso-Deploy/Lists/Views/Lista Empresas por
120            Obra.xml">
121            <ContentTypes>
122              <RemoveContentType Name="Elemento"/>
123              <ContentType Name="Empresa por Obra"/>
124            </ContentTypes>
125          </List>
126          <List Name="Tipos de gremio" UrlPath="Lists\TiposGremios"
127            TemplateType="GenericList" EnableContentTypes="true"
128            ViewSourcePath="Proceso-Deploy/Lists/Views/Lista Tipos de
129            Gremios.xml">
130            <ContentTypes>
131              <RemoveContentType Name="Elemento"/>
132              <ContentType Name="Tipo de gremio"/>
133            </ContentTypes>
134          </List>
135          <List Name="Incidencias" UrlPath="Lists\Incidencias"
136            TemplateType="GenericList" EnableContentTypes="true"
137            ViewSourcePath="Proceso-Deploy/Lists/Views/Lista Incidencias.xml">
138            <ContentTypes>
139              <RemoveContentType Name="Elemento"/>
140              <ContentType Name="Incidencia"/>
141            </ContentTypes>
142          </List>
```

```

132 <!--Bibliotecas-->
133 <List Name="Imágenes incidencias" UrlPath="ImagenesIncidencias"
    TemplateType="DocumentLibrary" EnableContentTypes="true"
    EnableAttachments="false" ViewsSourcePath="Proceso-Deploy/Lists/
134 Views/Biblioteca Imágenes de Incidencias.xml">
    <ContentTypes>
135 <RemoveContentType Name="Documento"/>
136 <ContentType Name="Imagen de incidencia"/>
137 </ContentTypes>
138 </List>
139 </Lists>

```

Figura 22. XML de las Listas y Bibliotecas de settingsAprovisionamientoBase del Proyecto de Aprovisionamiento

### 3.3.3 Implementación de la Power APP “Gestor Incidencias”

Lo primero para este desarrollo ha sido crear las diferentes pantallas establecidas en el mockup.

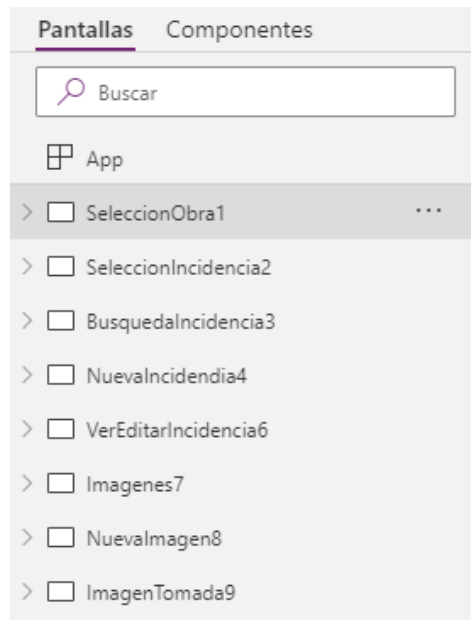


Figura 23. Esquema de Pantallas de la Power APP

Después se ha pasado a crear un componente para la cabecera de la APP ya que va a ser común para todas las pantallas en la que se muestra el título de “Gestor Incidencias” y la foto de perfil del usuario logueado.

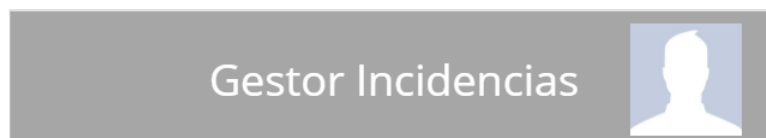


Figura 24. Componente Cabecera de la Power APP

Que tendrá como propiedades solamente un label y una imagen.

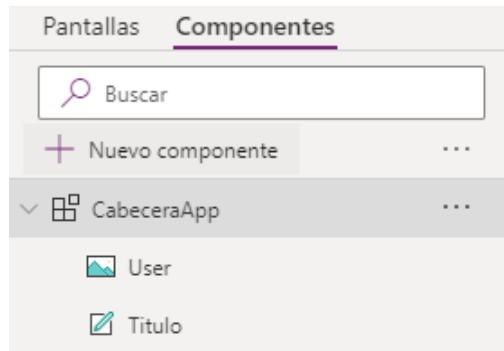


Figura 25. Esquema del Componente Cabecera de la Power APP

Para mostrar la foto de perfil, en la imagen, se usa la siguiente fórmula en la propiedad Image de la imagen User.

DATOS

Image

```
UsuariosdeOffice365.UserPhotoV2(User  
().Email)
```

Figura 26. Fórmula que muestra la foto de perfil del usuario conectado de la Power APP

A continuación, se va a proceder a explicar y mostrar el proceso de creación de las distintas pantallas.

a) Pantalla de selección de obra.

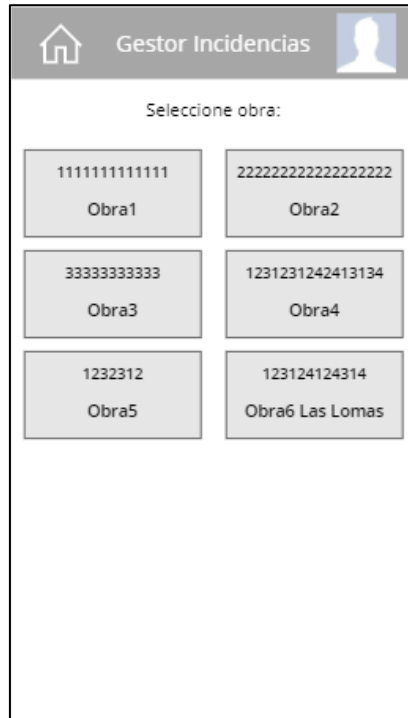


Figura 27. Pantalla de Selección de Obra de la Power APP

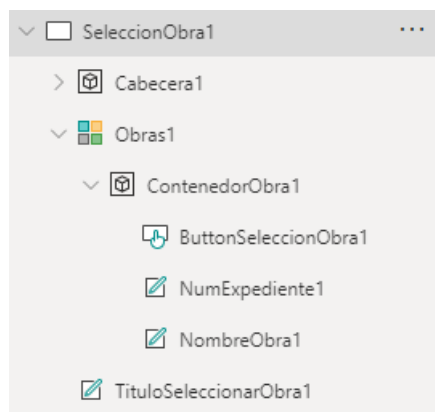


Figura 28. Esquema de la Pantalla de Selección de Obra de la Power APP

En esta pantalla se ha utilizado el componente Cabecera (igual que en todas las demás) y se ha añadido un Label, que mostrará el mensaje de seleccione una obra, y una galería que se conectará al origen de datos de la lista de obras de SharePoint, para traer todo su contenido y mostrar el número de expediente y el nombre de la obra. Dentro también de la galería se ha implementado un botón para guardar la obra en una variable global de Power Apps y, poder acceder a ella más adelante. El funcionamiento del botón se definirá en su propiedad onSelect y aparte de hacer el set de la variable, cambiará a la pantalla de selección de incidencia.

## ACCIÓN

OnSelect

```
Set( ObraSeleccionada; ThisItem )  
And  
Navigate(SeleccionIncidencia2)
```

Figura 29. Fórmula de selección de Obra de la Power APP

Una vez seleccionada la incidencia se pasa a la pantalla de selección de incidencia.

### b) Pantalla de selección de incidencia.

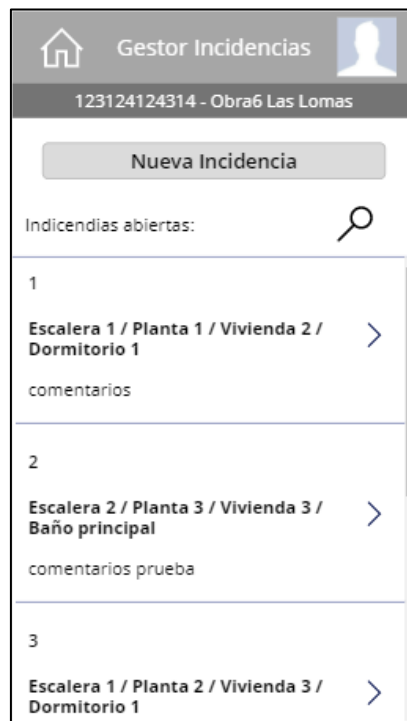


Figura 30. Pantalla de Selección de Incidencia de la Power APP

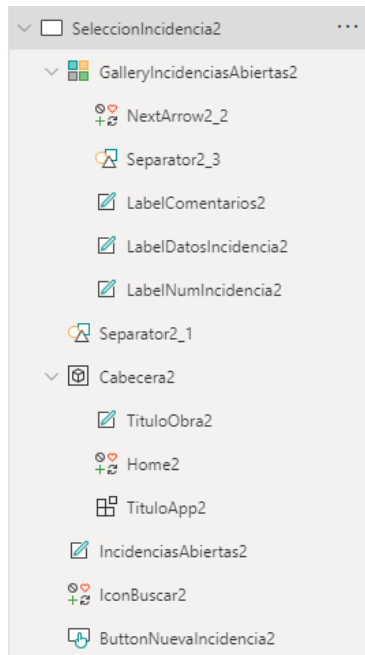


Figura 31. Esquema de la Pantalla de Selección de Incidencia de la Power APP

En esta pantalla se ha implementado una galería en la que se muestran las incidencias asociadas a la obra seleccionada, en la primera pantalla y, cuyo estado sea igual a “Registrada”. Para obtener estas incidencias, se filtra la lista de incidencias de la siguiente manera: en el campo ítems de la galería se busca la cadena “Registrada” en la columna Estado y se obtiene también el campo ID de la obra seleccionada (en la variable global), para compararlo con el ID de la columna de búsqueda que apunta a obra.

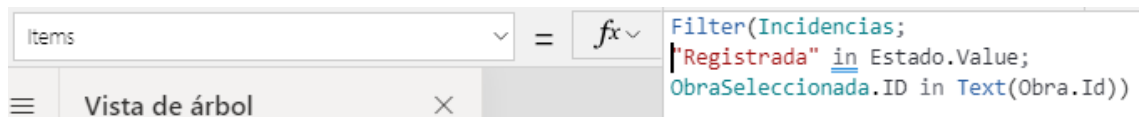


Figura 32. Fórmula para filtrar la galería de Incidencias de la Power APP

Si se selecciona una incidencia, ésta se guarda en la variable global “IncidenciaSeleccionada”, y se pasa a la pantalla de ver y editar incidencia. Esta pantalla contiene también un botón para acceder a la pantalla de nueva incidencia y otro, para acceder a la pantalla de búsqueda de incidencia.

c) Pantalla de búsqueda de incidencia.

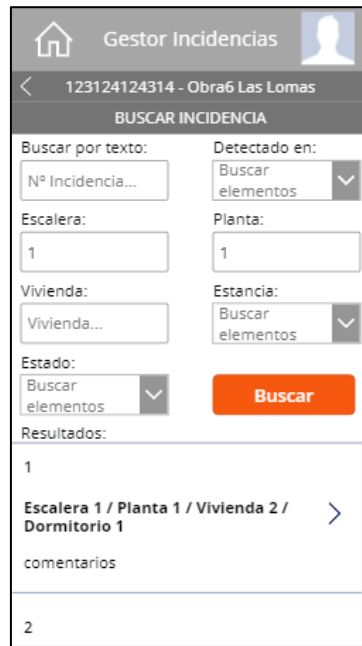


Figura 33. Pantalla de Búsqueda de Incidencias de la Power APP

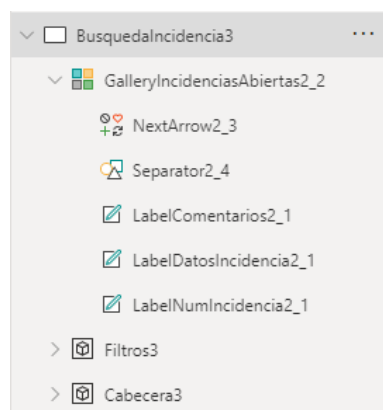


Figura 34. Esquema de la Pantalla de Búsqueda de Incidencias de la Power APP

En esta pantalla se definen TextInputs y ComboBoxes, para las distintas columnas de la lista de Incidencias por las que se van a filtrar los resultados de la búsqueda. También se establece un botón para buscar y aplicar los distintos filtros que se hayan seleccionado.

La galería que se define, en esta pantalla, sólo mostrará resultados una vez que se haya pulsado el botón Buscar. La fórmula para filtrar los items, filtra sólo las categorías asociadas a la obra seleccionada, en la primera pantalla, y aplica los filtros introducidos por el usuario. Para el filtrado la fórmula comprueba si el contenido de los filtros está vacío, en cuyo caso se devuelve true (para no aplicar ese filtro) y, si el componente contiene un valor éste se compara con la columna correspondiente de la lista de incidencias.

```

fx
If(Buscar = true;
Filter(Incidencias;
ObraSeleccionada.ID in Text(Obra.Id);
If(IsBlank(TextInputBuscarTexto3.Text); true ; TextInputBuscarTexto3.Text in 'Número de incidencia');
If(IsBlank(ComboBoxDetectadoEn3.Selected.Value); true ; ComboBoxDetectadoEn3.Selected.Value in 'Detectado en'.Value);
If(IsBlank(TextInputEscalera3.Text); true ; TextInputEscalera3.Text in Escalera);
If(IsBlank(TextInputPlanta3.Text); true ; TextInputPlanta3.Text in Planta);
If(IsBlank(TextInputVivienda3.Text); true ; TextInputVivienda3.Text in Vivienda);
If(IsBlank(ComboBoxEstancia3.Selected.Value); true ; ComboBoxEstancia3.Selected.Value in Estancia.Value);
If(IsBlank(ComboBoxEstado3.Selected.Value); true ; ComboBoxEstado3.Selected.Value in Estado.Value)
))

```

Figura 35. Fórmula que aplica los filtros de búsqueda de Incidencias de la Power APP

**d) Pantalla de nueva incidencia.**

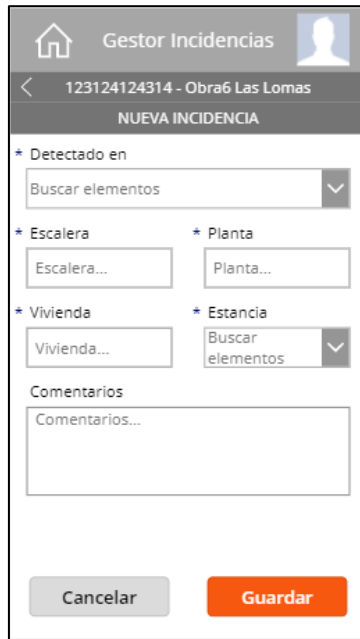


Figura 36. Pantalla de Nueva Incidencia de la Power APP

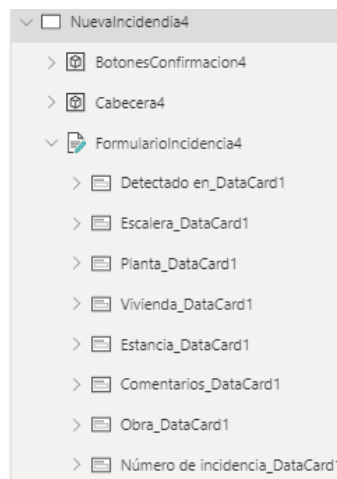


Figura 37. Esquema de la Pantalla de Nueva Incidencia de la Power APP

En esta pantalla, se implementa un formulario para registrar nuevas incidencias en SharePoint. Este formulario contiene los campos de Tipo de incidencia, escalera, planta, vivienda, estancia, comentarios, obra asociada y número de incidencia.

Todos los campos son visibles y rellenos por el usuario excepto los campos de obra y de número de incidencia que se autogeneran a la hora de registrar la incidencia.

Campo de obra: En la propiedad “DefaultSelectedItems” se define la siguiente fórmula, para asociar su valor con la obra seleccionada en la pantalla de selección de obra.

```

fx ~ If(
  FormularioIncidencia4.Mode = New;
  {
    '@odata.type': "#Microsoft.Azure.Connectors.SharePoint.SPListExpandedReference";
    Id: Value(ObraSeleccionada.ID);
    Value: LookUp(
      Obras;
      ID = Value(ObraSeleccionada.ID);
      NombreDeObra
    )
  };
  ThisItem.Obra
)

```

Figura 38. Fórmula para asignar la Obra asociada a la Incidencia de la Power APP

Campo de número de incidencia: Para este campo se aplica a su propiedad Default la siguiente fórmula, que concatena el año en el momento que se registra la incidencia junto con un contador de cuatro dígitos, que se rellena teniendo en cuenta si hay alguna incidencia registrada en ese mismo año. En caso negativo, los cuatro últimos dígitos del número de la incidencia se establecen a “0000” y en caso afirmativo, se suma 1 a los cuatro últimos dígitos de la última incidencia registrada.

```

Concatenate(Text(Year( Now() ));
If( Text(Year( Now() )) = Left( Text(Max(Incidencias;'Número de incidencia')); 4 );
Right(Text(Max(Incidencias;'Número de incidencia') + 1); 4);
"0000"
)
)

```

Figura 39. Fórmula para asignar el Número de Incidencia de la Power APP

De esta manera las incidencias registradas para el año 2021 tendrían los siguientes valores: “20210000”, “20210001”, “20210002”, ...

Lo último que se establece en esta pantalla es un botón, para registrar la incidencia en la lista de SharePoint. Como el formulario está en modo New, al realizar el SubmitForm la incidencia se añade como una nueva a la lista. Aparte de esto, el botón también establece una variable que controla el modo en el que se presenta el formulario de la pantalla de visualización y edición en modo “Ver”.

```

SubmitForm(FormularioIncidencia4) And
Set(ModoFormulario; "Ver")
And
Set(IncidenciaSeleccionada; FormularioIncidencia4.LastSubmit)
And
ResetForm(FormularioIncidencia4)
And
Navigate(VerEditarIncidencia6;Fade)

```

Figura 40. Fórmula para Añadir una Incidencia a SharePoint de la Power APP

e) Pantalla de visualización y edición de incidencia.

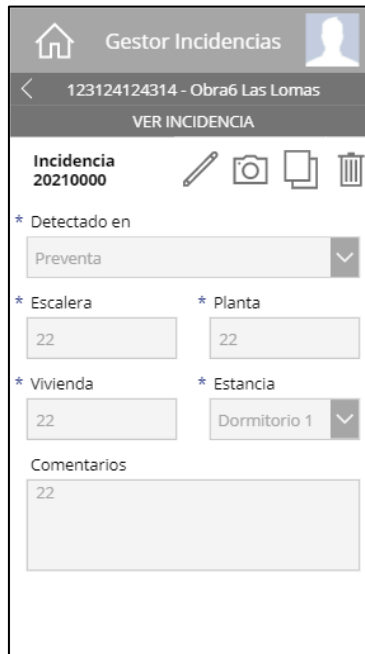


Figura 41. Pantalla de Visualización de Incidencia de la Power APP

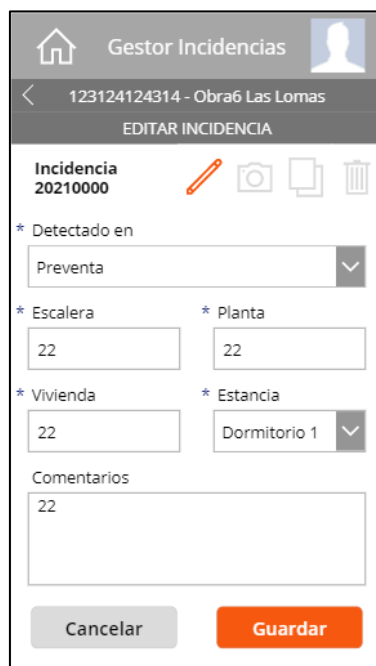


Figura 42. Pantalla de Edición de Incidencia de la Power APP

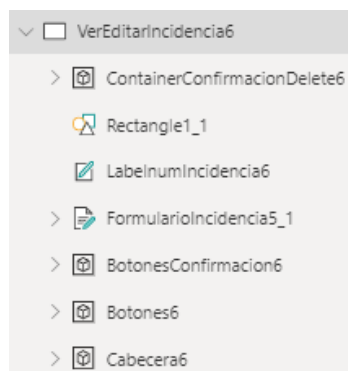


Figura 43. Esquema de la Pantalla de Visualización y Edición de Incidencia de la Power APP

A esta pantalla se accede al seleccionar una incidencia, en la pantalla de selección de incidencia y al acabar de registrar una nueva incidencia. En ambos casos, esta pantalla muestra la incidencia seleccionada o creada en un formulario en modo Ver, lo que quiere decir que los campos del formulario no se pueden editar.

Si se pulsa el icono del lápiz se establece el formulario en modo edición, lo que hace que los campos del formulario sean editables, se muestren los botones de guardar, que actualiza la incidencia preseleccionada de la lista de SharePoint y, el botón de cancelar que cancela la edición y vuelve a poner el formulario en modo Ver. En el modo Edición, el resto de botones están deshabilitados.

Si se pulsa el icono de la cámara, se pasa a la pantalla de imágenes de incidencia.

Si se pulsa el icono de las dos hojas, se va a la pantalla de nueva incidencia y se preseleccionan los campos, de su formulario, con los valores de la incidencia que se estaba visualizando.

Ejemplo del campo del formulario de Nueva Incidencia a la hora de copiar una incidencia:

```
If(ModoFormulario = "Copiar";DataCardValue4_2)
```

Figura 44. Fórmula de un campo del formulario para copiar el valor de otro de la Power APP

Si se pulsa el icono de la papelera, se muestra una ventana emergente para confirmar el eliminado y, si se confirma, se elimina la incidencia seleccionada en el formulario de la lista de Incidencias y se recorre la biblioteca de imágenes de incidencias para borrar todas las imágenes que tengan esa incidencia asociada con la instrucción Remove.

```
Remove(Incidencias; IncidenciaSeleccionada)
;;
ForAll('Imágenes incidencias';
If(IncidenciaSeleccionada.ID = ThisRecord.Incidencia.Id;
Collect(ImagenesBorrar;ThisRecord)
))
;;
Remove('Imágenes incidencias'; ImagenesBorrar)
;;
Navigate(SeleccionIncidencia2;Fade)
```

Figura 45. Fórmula para eliminar una Incidencia y sus Imágenes asociadas de la Power APP

f) **Pantalla de visualización de imágenes de incidencia.**



Figura 46. Pantalla de Visualización de Imágenes de Incidencia de la Power APP

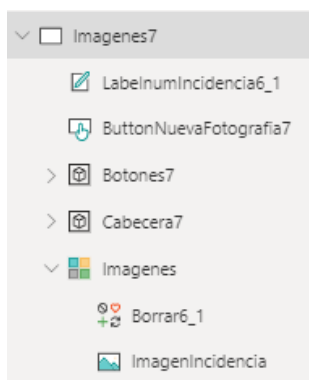


Figura 47. Esquema de la Pantalla de Visualización de Imágenes de Incidencia de la Power APP

Esta pantalla contiene los mismos botones, con la misma funcionalidad, que la pantalla de vista y edición de incidencia. También contiene un botón para agregar nuevas imágenes a la incidencia, que lleva a la siguiente pantalla de agregación de imágenes y, se muestran las imágenes asociadas a la incidencia seleccionada en una galería. Cada imagen tiene un icono de una papelera, con el que se puede borrar la imagen de la biblioteca de Share Point. Según que eliminar se seleccione se muestra un mensaje de confirmación u otro mediante un condicional. En el caso de eliminar una incidencia también se eliminan todas sus imágenes asociadas como en el caso anterior.

Formula de confirmación del eliminado incidencia e imagen.

```

If(ItemEliminar = "Imagen";
Remove('Imágenes incidencias'; Imagenes.Selected);
ForAll('Imágenes incidencias';
If(IncidenciaSeleccionada.ID = ThisRecord.Incidencia.Id;
Collect(ImagenesBorrar;ThisRecord)
))
;;
Remove('Imágenes incidencias'; ImagenesBorrar)
;;
Remove(Incidencias; IncidenciaSeleccionada))
;;
If(ItemEliminar = "Incidencia";
Navigate(SeleccionIncidencia2;Fade)
)
;;
Set(MensajeConfirmacion; "Oculto")

```

Figura 48. Fórmula para eliminar una Incidencia y sus Imágenes asociadas o solo una Imagen asociada de la Power APP

**g) Pantalla de agregación de imagen de incidencia.**



Figura 49. Pantalla de Agregación de Imágenes de Incidencias de la Power APP

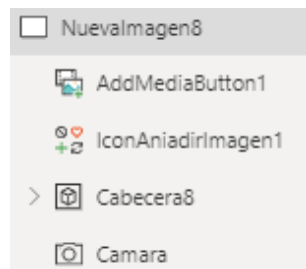
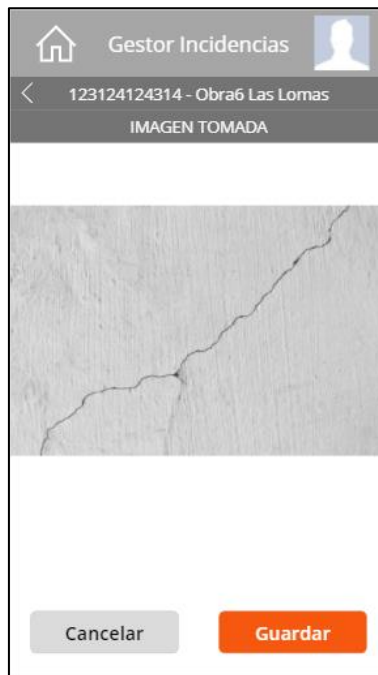


Figura 50. Esquema de la Pantalla de Agregación de Imágenes de Incidencias de la Power APP

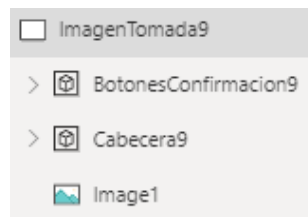
En esta pantalla, se usa el componente Cámara que accede a la cámara del dispositivo para tomar imágenes. Al pulsar la visualización de la imagen, que está tomando la cámara, se realiza la fotografía y se pasa a la pantalla de imagen tomada.

También contiene un icono que dispara el componente “Agregar Imagen”, con el que acceder a la galería del dispositivo para seleccionar una. Después de seleccionar la imagen, se pasa también a la pantalla de imagen tomada.

**h) Pantalla de visualización de la imagen tomada o seleccionada de la galería.**



*Figura 51. Pantalla de Visualización de la Imagen tomada o seleccionada de la Power APP*



*Figura 52. Esquema de la Pantalla de Visualización de la Imagen tomada o seleccionada de la Power APP*

Esta pantalla muestra la imagen tomada mediante la cámara o seleccionada en la galería. Si se pulsa el botón cancelar, la imagen se descarta y se vuelve a la pantalla de Nueva Imagen. Si se pulsa el botón guardar, la imagen se agrega a la biblioteca de imágenes de incidencias de Share Point, se le asocia la incidencia correspondiente y se vuelve a la pantalla de imágenes de incidencia.

### 3.3.4 Creación del evento y de la función de Azure para el análisis automático de las imágenes cuando son agregadas a SharePoint.

Lo primero que se necesita para esta parte es una aplicación de consola para asociar y desasociar eventos de lista de SharePoint.

Esta aplicación se ha implementado en Visual Studio 2017.

Se define una clase que contiene una lista en la que se definen los eventos a asociar o desasociar.

```
public static class EventCollectionConfig
{
    // Obtiene la configuracion
    public static IList<EventReceiverDefinitionCreationInformation>
    GetConfiguration()
    {
        IList<EventReceiverDefinitionCreationInformation> eventCollection =
        new List<EventReceiverDefinitionCreationInformation>()
        {
            new EventReceiverDefinitionCreationInformation()
            {
                EventType = EventReceiverType.ItemAdded,
                ReceiverAssembly = "Microsoft.SharePoint, Version=16.0.0.0,
                Culture=neutral, PublicKeyToken=71e9bce111e9429c",
                ReceiverUrl = "https://northeurope0.pushnp.svc.ms/
                notifications?token=047a3b41-234f-45f7-9963-9e8d2beb3ed1",
                ReceiverName = "ContosoProyectosItemAdded"
            },
            new EventReceiverDefinitionCreationInformation()
            {
                EventType = EventReceiverType.ItemAdded,
                ReceiverAssembly = "Contoso, Version=1.0.0.0,
                Culture=neutral, PublicKeyToken=null",
                ReceiverUrl = "https://northeurope0.pushnp.svc.ms/
                notifications?token=047a3b41-234f-45f7-9963-9e8d2beb3ed1",
                ReceiverName = "ContosoProyectosItemAdded"
            },
            new EventReceiverDefinitionCreationInformation()
            {
                EventType = EventReceiverType.ItemUpdated,
                ReceiverAssembly = "Contoso, Version=1.0.0.0,
                Culture=neutral, PublicKeyToken=null",
                ReceiverUrl = "https://northeurope0.pushnp.svc.ms/
                notifications?token=047a3b41-234f-45f7-9963-9e8d2beb3ed1",
                ReceiverName = "ContosoProyectosItemUpdated"
            }
        };
        return eventCollection;
    }
}
```

Figura 53. Clase EventCollectionConfig del proyecto de asociación de Eventos

Posteriormente, se implementa el método AttachEventToList, para asociar eventos, y el método DettachEventToList, para desasociarlos.

Lo primero, es establecer el contexto autenticándose contra el sitio de SharePoint.

```

ClientContext context = null;

try
{
    context = new ClientContext(Constants.SPOWeb) { Credentials =
        new SharePointOnlineCredentials(Constants.SPOAccount,
            Constants.SPOPassword) };
}

```

Figura 54. Clase EventCollectionConfig del proyecto de asociación de Eventos

Posteriormente se obtiene la biblioteca de Imágenes de Incidencias, junto con su propiedad EventReceivers, que contiene los eventos asociados a ella.

```

foreach (var listName in Constantes.ListNames)
{
    try
    {
        List list_Maestro = context.Web.Lists.GetByTitle
            (listName);

        if (list_Maestro == null)
        {
            throw new Exception($"lista {listName} no
                encontrada");
        }

        context.Load(list_Maestro, w => w.EventReceivers);
        context.ExecuteQuery();

        EventReceiverDefinitionCollection eventReceivers =
            list_Maestro.EventReceivers;
    }
}

```

Figura 55. Código para la obtención de una Lista de SharePoint del proyecto de asociación de Eventos

Finalmente, para el añadido de eventos, se recorre la lista de eventos de la clase EventCollectionConfig donde se definieron anteriormente los eventos. Se busca por la propiedad ReceiverName, si el evento existe ya en la biblioteca, en cuyo caso se añadirá el nuevo evento y si no existe, no se hará nada.

```

--
40         foreach (EventReceiverDefinitionCreationInformation
eventDefinition in EventCollectionConfig.GetConfiguration
())
41         {
42             try
43             {
44                 if (eventReceivers == null)
45                 {
46                     Console.WriteLine($"eventos de la lista
{listName} no obtenidos");
47                     Console.Read();
48                     continue;
49                 }
50
51                 var eventReceiver = eventReceivers.Where(w =>
w.ReceiverName ==
eventDefinition.ReceiverName).FirstOrDefault();
52
53                 if (eventReceiver != null)
54                 {
55                     throw new Exception($"El evento ya está
asociado, [Evento: {eventDefinition.ReceiverName}] [Lista:
{listName}]");
56                 }
57
58                 Console.WriteLine("Añadiendo evento");
59
60                 list_Maestro.EventReceivers.Add
(eventDefinition);
61                 context.ExecuteQuery();
62
63             }
catch (Exception ex)

```

Figura 56. Código para añadir eventos a una lista de SharePoint del proyecto de asociación de Eventos

En el caso de desasociar eventos, el funcionamiento es muy similar, sólo que cuando se encuentre la coincidencia entre el evento definido y el evento de la biblioteca, éste se eliminará.

```

134                 if (eventReceiver.ReceiverName ==
eventDefinition.ReceiverName) {
135                     eventReceiver.DeleteObject();
136                     context.ExecuteQuery();
137                 }

```

Figura 57. Código para eliminar eventos a una lista de SharePoint del proyecto de asociación de Eventos

Ya con lo necesario, para controlar los eventos de la biblioteca, se crea el evento `ImagenIncidenciaItemAdded` en la clase `EventCollectionConfig` y se ejecuta el programa para añadirlo a la biblioteca.

Se accede al sitio de SharePoint mediante la herramienta de SharePoint Online Client Browser para confirmar que el evento creado de la biblioteca de imágenes de incidencia ha sido añadido correctamente.

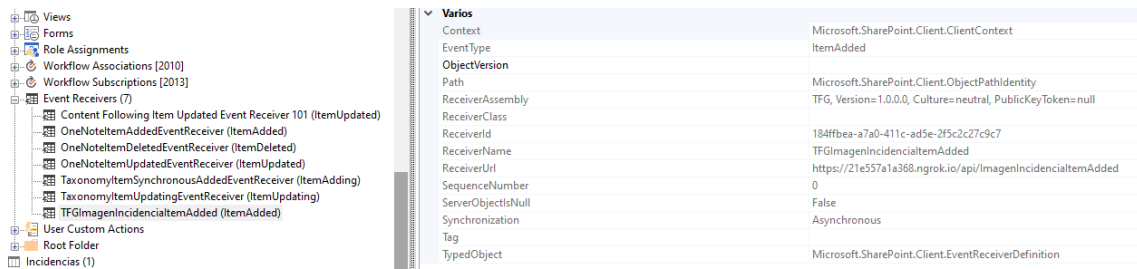


Figura 58. Confirmación del añadido del evento mediante SharePoint Client Browser

A continuación, se instalan las herramientas necesarias para la creación de funciones de Azure mediante el siguiente comando:

```
npm i -g azure-functions-core-tools@3.0.3442
```

Ahora se pasa a crear un directorio donde se guardará la solución, el cual es:

```
C:\Users\rcampo\Cientes\TFG\Procesos\TFG.ImagenIncidencia.Items
```

Luego, se crea el scaffolding de la solución dentro de la carpeta TFG.ImagenIncidencia.Items utilizando el comando "func init" y se elige dotnet.

```
C:\Users\rcampo\Cientes\TFG\Procesos\TFG.ImagenIncidencia.Items>func init
Use the up/down arrow keys to select a worker runtime:
dotnet
dotnet (isolated process)
node
python
powershell
custom
```

Figura 59. Comando func init para la creación del scaffolding del proyecto de las funciones de Azure

El comando nos genera el siguiente scaffolding:

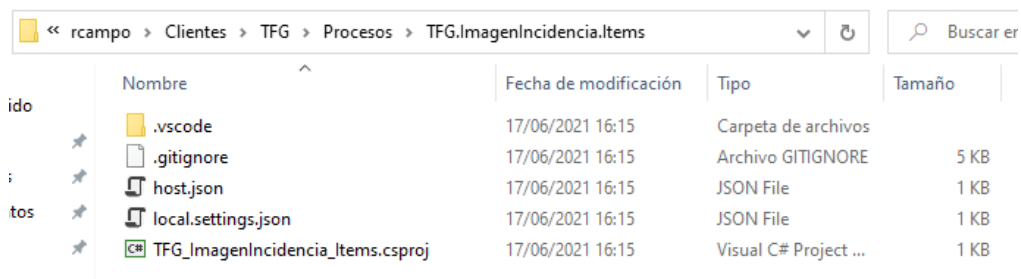


Figura 60. Scaffolding del proyecto de las funciones de Azure

Ahora se pasa a crear la función de Azure, a la que se llamará cuando se reciba un evento de que se ha añadido una imagen a la biblioteca.

Comando para crear la función de Azure:

```
func new
```

Se selecciona la función que es de tipo HttpTrigger.

```
C:\Users\rcampo\Clientes\TFG\Procesos\TFG.ImagenIncidencia.Items>func new
Use the up/down arrow keys to select a template:
QueueTrigger
HttpTrigger
BlobTrigger
TimerTrigger
DurableFunctionsOrchestration
SendGrid
EventHubTrigger
ServiceBusQueueTrigger
ServiceBusTopicTrigger
EventGridTrigger
CosmosDBTrigger
IoTHubTrigger
```

Figura 61. Comando `func new` para la creación de la función de Azure

Y se le da un nombre a la función:

```
C:\Users\rcampo\Clientes\TFG\Procesos\TFG.ImagenIncidencia.Items>func new
Use the up/down arrow keys to select a template:Function name: ImagenIncidenciaItemAdded
ImagenIncidenciaItemAdded

The function "ImagenIncidenciaItemAdded" was created successfully from the "HttpTrigger" template.
C:\Users\rcampo\Clientes\TFG\Procesos\TFG.ImagenIncidencia.Items>
```

Figura 62. Dando nombre a la función de Azure

Una vez creada la función, se creará una solución vacía en Visual Studio 2019, con el nombre de "TFG.Solution.Procesos" y en la ruta, "C:\Users\rcampo\Clientes\TFG\Procesos", a la que se le añadirá el proyecto creado que contiene la función de Azure.

Lo primero que se necesita para implementar la función de Azure es autenticarse contra SharePoint, una vez ha llegado el evento de añadido de una imagen.

Cabe destacar que se utiliza Visual Studio 2019 para hacer uso .NET Core 3.1. y que éste no admite la autenticación como se mostró en la figura 54, por lo que se procede a realizar una autenticación mediante certificado.

Para la creación del certificado usando la consola de Windows PowerShell, como administrador, se ejecutan los dos siguientes comandos; el primero para establecer la contraseña del certificado y el segundo que crea el propio certificado.

```
-$certPassword =ConvertTo-SecureString "password" -AsPlainText -Force
-New-PnPAzureCertificate -OutPfx C:\Users\rcampo\Certificados\Prueba\Prueba.pfx -OutCert
C:\Users\rcampo\Certificados\Prueba\Prueba.cer -CertificatePassword $certPassword
```

Esto generará el siguiente certificado:



 Certificado.cer	22/06/2021 11:09	Certificado de seguridad	1 KB
 Certificado.pfx	22/06/2021 11:09	Personal Information Exchange	3 KB

Figura 63. Certificado

Una vez creado se registra una nueva aplicación en el portal de Azure y dentro de ésta, se pasa a cargar el certificado generado: Certificado.cer. Y se agregan permisos de control total al sitio de SharePoint para la aplicación registrada.

Ahora se pasa a añadir el Certificado.pfx al proyecto TFG.Solution.Procesos.

Ya para terminar con la autenticación, queda implementar la clase TokenProvider en el proyecto, que será la encargada de llevar a cabo la autenticación.

Para instanciar la clase TokenProvider se necesitan las siguientes variables que contendrán el ID del cliente, el ID del Tenant de SharePoint, la contraseña del certificado y los bytes del propio certificado.

```
class TokenProvider
{
    readonly string _clientId = null;
    readonly string _tenantId = null;
    readonly string _certificatePassword = null;
    readonly byte[] _certificateBytes = null;

    public TokenProvider(string clientId, string tenantId, string
        certificatePassword, byte[] certificateBytes)
    {
        _clientId = string.IsNullOrWhiteSpace(clientId) ? throw new
            ArgumentNullException(nameof(clientId)) : clientId;
        _tenantId = string.IsNullOrWhiteSpace(tenantId) ? throw new
            ArgumentNullException(nameof(tenantId)) : tenantId;
        _certificatePassword = string.IsNullOrWhiteSpace
            (certificatePassword) ? throw new ArgumentNullException(nameof
            (certificatePassword)) : certificatePassword;

        if (certificateBytes == null)
        {
            throw new ArgumentNullException(nameof(certificateBytes));
        }

        _certificateBytes = certificateBytes;
    }
}
```

Figura 64. Constructor de la Clase TokenProvider del proyecto de la Función de Azure

Y para la obtención del token de autenticación mediante certificado se implementan los siguientes métodos en la misma clase:

```

29     public string GetToken(string scope)
30     {
31         string rootWebPath = new Uri(scope).GetLeftPart
            (UriPartial.Authority);
32         IList<string> scopes = new List<string>() { $"{rootWebPath.TrimEnd
            ('/')}/.default" };
33
34         X509Certificate2 certificate = GetCertificate(_certificateBytes,
            _certificatePassword);
35
36         var confidentialClient = ConfidentialClientApplicationBuilder
37             .Create(_clientId)
38             .WithTenantId(_tenantId)
39             .WithCertificate(certificate)
40             .Build();
41
42         var token = confidentialClient.AcquireTokenForClient
            (scopes).ExecuteAsync().GetAwaiter().GetResult();
43
44         return token.AccessToken;
45     }
46
47     public static X509Certificate2 GetCertificate(string path, string
            password) => new X509Certificate2(path, password,
            X509KeyStorageFlags.MachineKeySet);
48
49     public static X509Certificate2 GetCertificate(byte[] content, string
            password) => new X509Certificate2(content, password,
            X509KeyStorageFlags.MachineKeySet);

```

Figura 65. Métodos de la Clase TokenProvider para la obtención del Token mediante Certificado del proyecto de la Función de Azure

A continuación, se lee el certificado de la propia carpeta del proyecto.

```

#region Leer Certificado

var workingDirectory = $"{System.IO.Directory.GetParent
(Path.GetDirectoryName(Assembly.GetExecutingAssembly
()).Location)}.FullName.EnsureSlash()";

string fileCert = $"{workingDirectory.EnsureSlash()}
{Constantes.certificateName}";
using FileStream fs = System.IO.File.OpenRead(fileCert);
byte[] certContent = ReadFully(fs);

#endregion

```

Figura 66. Lectura del certificado en el proyecto de la Función de Azure

Y se obtiene el Token para instanciar el ClientContext de SharePoint.

```

111     private static ClientContext GetSPClientContext (Byte[] certContent)
112     {
113         var tokenProvider = new TokenProvider(clientId:
            Constantes.clientId, tenantId: Constantes.tenantId,
            certificatePassword: Constantes.certificatePassword,
            certificateBytes: certContent);
114
115         string token = tokenProvider.GetToken(Constantes.SPUrl);
116
117         if (string.IsNullOrEmpty(token))
118         {
119             throw new ArgumentNullException(nameof(token));
120         }
121
122         // Prueba SP
123         ClientContext context = new ClientContext(Constantes.SPUrl);
124
125         context.ExecutingWebRequest += (sender, e) =>
126         {
127             e.WebRequestExecutor.RequestHeaders["Authorization"] =
                $"Bearer {token}";
128         };
129
130         return context;
131     }

```

*Figura 67. Instanciando el contexto de SharePoint en el proyecto de la Función de Azure*

Con esto ya se tiene acceso a SharePoint por lo que se pasa a leer el evento recibido y sacar las propiedades necesarias con las que se trabajará posteriormente. Para esto se programa la función ParseValues que leerá el XML del evento recibido y devolverá la información sobre la lista y la imagen que se ha añadido.

```

71     private static InfoItemAdded ParseValues (XDocument xDoc)
72     {
73         const string XMLNODE_WebUrl = "WebUrl";
74         const string XMLNODE_ListItemId = "ListItemId";
75         const string XMLNODE_ListId = "ListId";
76
77         var descendantCollection = xDoc.Root.Descendants();
78         InfoItemAdded infoItemAdded = new InfoItemAdded();
79
80         foreach (System.Xml.Linq.XElement descendant in
81             descendantCollection)
82         {
83             string localName = descendant.Name?.LocalName;
84
85             if (string.IsNullOrEmpty(localName))
86             {
87                 continue;
88             }
89
90             switch (localName)
91             {
92                 case XMLNODE_WebUrl:
93                     infoItemAdded.WebUrl = descendant.Value;
94                     break;
95
96                 case XMLNODE_ListId:
97                     infoItemAdded.ListId = descendant.Value;
98                     break;
99
100                case XMLNODE_ListItemId:
101                    infoItemAdded.ListItemId = descendant.Value;
102                    break;
103
104                default:
105                    break;
106            }
107
108         return infoItemAdded;
109     }

```

Figura 68. Función ParseValues del proyecto de la Función de Azure

Esta función se llama después de haber leído el contenido del evento, como se ve en el siguiente fragmento de código:

```

22     public static class ImagenIncidenciaItemAdded
23     {
24         [FunctionName("ImagenIncidenciaItemAdded")]
25         public static async Task<IActionResult> Run([HttpTrigger
26             (AuthorizationLevel.Function, "get", "post", Route = null)]
27             HttpRequest req, ILogger log)
28         {
29             #region Leer Evento
30
31             InfoItemAdded infoItemAdded = null;
32
33             XDocument doc = XDocument.Load(new System.IO.MemoryStream
34                 (ReadFully(req.Body)));
35             infoItemAdded = ParseValues(doc);
36
37             #endregion

```

Figura 69. Lectura del evento y llamada a ParseValues del proyecto de la Función de Azure

El siguiente paso es obtener la imagen de la incidencia que se ha agregado a SharePoint. Para este fin se implementa la función `GetImagenIncidencia`, que recibirá como parámetros el contexto, el Id de la lista y el Id del elemento agregado a ella.

```
133     private static File GetImagenIncidencia(ClientContext context, string
134     listId, string listItemId)
135     {
136         // Obtener la biblioteca
137         List libraryImágenes = context.Web.Lists.GetById(new Guid
138         (listId));
139
140         ListItem imagenIncidencia = libraryImágenes.GetItemById
141         (listItemId);
142
143         context.Load(imagenIncidencia, i => i.File);
144         context.ExecuteQuery();
145
146         File imagen = imagenIncidencia.File;
147         return imagen;
148     }
149     ...
```

*Figura 70. Lectura del evento y llamada a `ParseValues` del proyecto de la Función de Azure*

Con la imagen obtenida, se programan dos funciones para instanciar el Servicio Cognitivo y analizar la imagen para obtener la etiqueta que se le agregará posteriormente en SharePoint.

La función que instancia el cliente de Custom Vision necesita el Id del proyecto de los servicios cognitivos, la imagen a analizar, la API key y el endPoint del servicio. Finalmente, llama a la función que realizará la predicción:

```
148     private static string GetEtiquetaPrediction()
149     {
150         Guid projectId = new Guid(Constantes.projectGuid);
151         var fileLocation = Constantes.imagenIncidenciaPath;
152
153         CustomVisionPredictionClient predictionApi = new
154         CustomVisionPredictionClient()
155         {
156             ApiKey = Constantes.apiKey,
157             Endpoint = Constantes.endPoint
158         };
159
160         string etiquetaImagen = Predict(predictionApi, projectId,
161         Constantes.iterationModelName, fileLocation);
162
163         return etiquetaImagen;
164     }
165     ...
```

*Figura 71. Función que instancia el cliente de Custom Vision del proyecto de la Función de Azure*

La función que se encarga de realizar la predicción primero lee los bytes de la imagen y llama a la API de los Servicios Cognitivos para realizar la predicción. Finalmente se itera en los resultados de la predicción para obtener el nombre de la etiqueta con la mayor probabilidad.

```

164 private static string Predict(CustomVisionPredictionClient
predictionApi, Guid projectID, string modelName, string
fileLocation)
165 {
166     string etiquetaResultado = "";
167     int i = 0;
168     double probabilidad = 0;
169
170     using (var stream = new MemoryStream(System.IO.File.ReadAllBytes
(fileLocation)))
171     {
172         Console.WriteLine("Realizando el análisis de la imagen");
173
174         var predictionResult = predictionApi.ClassifyImage(projectID,
modelName, stream);
175
176         foreach (var c in predictionResult.Predictions)
177         {
178             if (i == 0)
179             {
180                 etiquetaResultado = c.TagName;
181                 probabilidad = c.Probability;
182             }
183             else
184             {
185                 if (probabilidad < c.Probability)
186                 {
187                     etiquetaResultado = c.TagName;
188                     probabilidad = c.Probability;
189                 }
190             }
191             i++;
192         }
193     }
194     return etiquetaResultado;
195 }

```

*Figura 72. Función que realiza la predicción de la imagen en el proyecto de la Función de Azure*

Obtenida la etiqueta se necesita recuperar el TermSet de SharePoint que contiene las diferentes etiquetas. Para ello, utilizando el contexto de SharePoint instanciado en la figura 67, se recupera el TermStore del sitio de SharePoint, luego se recupera el TermGroup TFG que es el grupo de términos creados para este TFG. Finalmente se Obtiene el TermSet de etiquetas.

```

197     private static TermSet GetTermSet(ClientContext context) {
198         // Obtener TermStore
199         TaxonomySession taxonomySession =
200             TaxonomySession.GetTaxonomySession(context);
201         taxonomySession.UpdateCache();
202         TermStore termStore =
203             taxonomySession.GetDefaultSiteCollectionTermStore();
204         context.Load(termStore,
205             termStoreArg => termStoreArg.WorkingLanguage,
206             termStoreArg => termStoreArg.Id,
207             termStoreArg => termStoreArg.Groups.Include(
208                 groupArg => groupArg.Id,
209                 groupArg => groupArg.Name
210             )
211         );
212         context.ExecuteQuery();
213
214         // Obtener TermGroup
215         TermGroup termGroup = termStore.Groups.ToList().FirstOrDefault(g
216             => g.Id == new Guid(Constants.termGroupId));
217         if (termGroup == null)
218         {
219             throw new Exception("No existe el grupo de terminos");
220         }
221         context.Load(termGroup);
222         context.ExecuteQuery();
223
224         // Obtener el TermSet
225         TermSet termSet = termStore.GetTermSet(new Guid
226             (Constants.termSetId));
227         context.Load(termSet);
228         context.ExecuteQuery();
229
230         return termSet;
231     }

```

Figura 73. Función para la obtención del TermSet Etiquetas del proyecto de la Función de Azure

Para acabar con el etiquetado de la imagen, se busca en el TermSet obtenido la etiqueta devuelta por los Servicios Cognitivos y se obtiene el término. De éste se obtiene el Id y el nombre, para concatenarlos como se muestra en la figura 74, y de esta manera introducir el término en la columna de Etiqueta de la imagen agregada a la biblioteca.

```

229     private static void SetEtiqueta(string etiqueta, string listId, string
230         listItemId, ClientContext context)
231     {
232         // Obtener el termSet de Etiquetas
233         TermSet termSetEtiquetas = GetTermSet(context);
234
235         // Obtener el termino asociado a la etiqueta
236         Term termEtiqueta = termSetEtiquetas.Terms.GetByName(etiqueta);
237
238         context.Load(termEtiqueta, t => t.Id, t => t.Name);
239         context.ExecuteQuery();
240
241         Guid guidTermEtiqueta = termEtiqueta.Id;
242         string nameTermEtiqueta = termEtiqueta.Name;
243
244         // Obtener imagen de incidencia
245         List libraryImágenesIncidencias = context.Web.Lists.GetById(new
246             Guid(listId));
247
248         ListItem imagenIncidencia = libraryImágenesIncidencias.GetItemById
249             (listItemId);
250
251         context.Load(imagenIncidencia);
252         context.ExecuteQuery();
253
254         // Asociar la etiqueta a la imagen
255         imagenIncidencia[Constants.fieldNameEtiqueta] = "-1;#" +
256             nameTermEtiqueta + "|" + guidTermEtiqueta;
257
258         imagenIncidencia.Update();
259         context.ExecuteQuery();
260     }

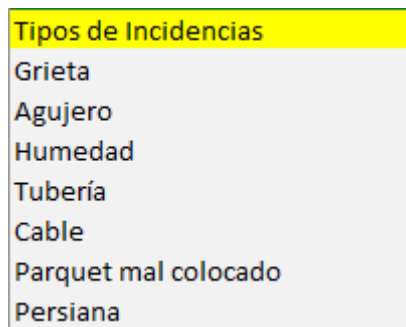
```

Figura 74. Función para Etiquetar la Imagen de Incidencia del proyecto de la Función de Azure

Una vez etiquetada la imagen es necesario implementar otra función de Azure, similar a la de `ImagenIncidencialtemAdded`, que en este caso será `ImagenIncidencialtemUpdated`, para que cuando la imagen se actualice en SharePoint, cuando se realice el etiquetado, llegue el evento y mediante código en el mismo proyecto, que se acaba de detallar, se agregue la columna de búsqueda de empresa a la imagen. Esta empresa se elegirá teniendo en cuenta que esté asociada a la Obra a la que pertenece la incidencia y que contenga la misma etiqueta que la incidencia.

### 3.3.5 Entreno del servicio cognitivo

Para comenzar con el entrenamiento del servicio cognitivo, lo primero, es clasificar los distintos tipos de incidencias.

Una lista vertical de tipos de incidencias. El título 'Tipos de Incidencias' está en un recuadro amarillo. Los tipos de incidencias listados son: Grieta, Agujero, Humedad, Tubería, Cable, Parquet mal colocado y Persiana.

Tipos de Incidencias
Grieta
Agujero
Humedad
Tubería
Cable
Parquet mal colocado
Persiana

*Figura 75. Tipos de Incidencias que van a ser clasificadas por los Servicios Cognitivos*

Una vez hecha la clasificación, se ha pasado a buscar alrededor de 50 imágenes, por tipo de incidencia, ya que el servicio recomienda un mínimo de 30 imágenes para que la clasificación sea fiable.

Ahora se pasa a crear un proyecto nuevo de Custom Vision, del cual se establece su nombre, que es de tipo Clasificador, Multiclase (una etiqueta por imagen) y de dominio general, ya que ninguno de los tipos de dominios concuerda con las incidencias de obras.

Después se crean las etiquetas mostradas en la figura 75 y posteriormente se cargan las imágenes, en partes, asociándolas a su etiqueta correspondiente.

Finalmente, se entrena el clasificador y este muestra el Porcentaje de precisión y de coincidencia que tiene con las imágenes cargadas.

## 4. Conclusiones y Trabajo Futuro

Para concluir con este TFG cabe decir que realizar este proyecto ha sido una experiencia muy enriquecedora, tanto personal como profesionalmente, ya que se ha podido ver lo que alcanza realmente un proyecto de una empresa y ha sido una oportunidad para poder trabajar con muchas tecnologías distintas e hilarlas entre sí. Ha sido un trabajo que ha llevado mucho tiempo, pero es satisfactoria la sensación de haberlo podido llevar a cabo y verlo finalmente en funcionamiento.

En cuanto a lo realizado creo que se ha llegado al objetivo del proyecto que era el de agilizar una parte del proceso de la construcción y hacerla mucho más sencilla, rápida y automática para ahorrar de esta manera horas de trabajo.

Sobre el trabajo futuro que se podría llevar a cabo para ampliar este TFG, se podría implementar un panel de cuadro de mando con PowerBI para mostrar al departamento de Gerencia de la empresa información como; cuántas incidencias por obra tienen, cuáles están resueltas, qué impacto económico tienen, etc.

Por otra parte se podría desarrollar unos servicios de geolocalización para saber, desde el edificio en construcción, en qué planta, escalera, vivienda y estancia se encuentra el trabajador a la hora de recoger la incidencia, para catalogar, en un solo paso, Unidad de venta de la incidencia más todos los datos de la propia incidencia.

## 5. Referencias bibliográficas

[1] INTELIGENCIA ARTIFICIAL Y SERVICIOS COGNITIVOS PARA LA TRANSFORMACIÓN DIGITAL. [CONSULTA: 22 ABRIL 2021]. DISPONIBLE EN: <https://directivosygerentes.es/digital/digital-talent/workshops/inteligencia-artificial-servicios-cognitivos-la-transformacion-digital>

[2] QUE SON LOS SERVICIOS COGNITIVOS. [CONSULTA: 22 ABRIL 2021]. DISPONIBLE EN: <https://itblogsgeti.com/2018/02/27/que-son-los-servicios-cognitivos/comment-page-1/>

[3] LOS 6 MEJORES SOFTWARE ERP PARA LA CONSTRUCCIÓN Y GESTIÓN DE OBRAS. [CONSULTA: 25 ABRIL 2021]. DISPONIBLE EN: <https://www.softwaredoit.es/software-construccion/index.html>

[4]. IEEE 830. (1998). IEEE Recommended Practice for Software Requirements Specifications. New York: IEEE.

[5]. ISO/IEC/IEEE 29148. (2011). Systems and software engineering — Life cycle processes — Requirements engineering. New York: IEEE.

[6]. DerEditor <http://dereditor.sourceforge.net/>

[7] The GNU Operating System and the Free Software Movement, 2021. Gnu.org [online]

## Anexo I – Diseño lógico de la Base de Datos

/\* Entidad Imagen de incidencia: \*/

Imagen de incidencia (IDImg integer, NumIncidencia:varchar, EtiquetasRelacionadas:varchar )

Clave Primaria { IDImg }

Unico { NumIncidencia }

/\* Entidad Incidencias: \*/

Incidencias (NumExpediente varchar, Comentarios:varchar, Coste:integer, Estado:varchar, Estancia:varchar, Vivienda:integer, Planta:integer, Escalera:integer, DetectadoEn:varchar )

Clave Primaria { NumExpediente }

Valor No Nulo { Estado }

Valor No Nulo { Estancia }

Valor No Nulo { Vivienda }

Valor No Nulo { Planta }

Valor No Nulo { Escalera }

Valor No Nulo { DetectadoEn }

/\* Entidad Empresas: \*/

Empresas (IDEmp integer, NIF:varchar, Nombre:varchar, Direccion:varchar, Telefono:varchar, CorreoElectronico:varchar )

Clave Primaria { IDEmp }

Unico { NIF }

Valor No Nulo { Nombre }

Valor No Nulo { CorreoElectronico }

Restriccion Empresas\_Totales

Empresasx: Empresas, Empresas por Obrax: Empresas por Obra

V Empresasx(Empresas(Empresasx) -> (E Empresas por Obrax(Empresas por Obra(Empresas por Obrax) ^ Empresas por Obrax.IDEmp=Empresasx.IDEmp)))

Restriccion Empresas\_Disjuntas

Empresas por Obrax: Empresas por Obra

∃ Empresas por Obrax (Empresas por Obra(Empresas por Obrax) ^ ( ))

/\* Entidad Obras: \*/

Obras (NumExpediente varchar, FechaEstimadaFin:date, FechaInicio:date, FechaFin:date, Presupuesto:varchar, JefesDeObra:varchar, Provincia:varchar, Localidad:varchar, NumViviendas:integer, Nombre:varchar, Telefono:varchar )

Clave Primaria { NumExpediente }

Valor No Nulo { FechaEstimadaFin }

Valor No Nulo { FechaInicio }

Valor No Nulo { Presupuesto }

Valor No Nulo { JefesDeObra }

Valor No Nulo { Provincia }

Valor No Nulo { Localidad }

Valor No Nulo { NumViviendas }

Valor No Nulo { Nombre }

Clave Ajena { NumExpediente } hace referencia a Contiene

/\* Entidad Tipos de Gremio: \*/

Tipos de Gremio (IDEmp integer, TipoDeGremio:varchar, EtiquetasRelacionadas:varchar )

Clave Primaria { IDEmp }

Valor No Nulo { TipoDeGremio }

Valor No Nulo { EtiquetasRelacionadas }

Clave Ajena { IDEmp } hace referencia a Empresas por Obra

/\* Entidad Empresas por Obra: \*/

Empresas por Obra (IDEmp integer )

Clave Primaria { IDEmp }

Clave Ajena { IDEmp } hace referencia a Empresas

Restriccion Empresas por Obra\_Totales

Empresas por Obrax: Empresas por Obra, Tipos de Gremio: Tipos de Gremio

V Empresas por Obrax(Empresas por Obra(Empresas por Obrax) -> (E Tipos de Gremiox(Tipos de Gremio(Tipos de Gremiox) ^ Tipos de Gremiox.IDEmp=Empresas por Obrax.IDEmp)))

Restriccion Empresas por Obra\_Disjuntas

Tipos de Gremiox: Tipos de Gremio

∃ Tipos de Gremiox (Tipos de Gremio(Tipos de Gremiox) ^ ( ))

/\* Relacion Aparecen: \*/

Aparecen (NumExpediente varchar, IDEmp integer )

Clave Primaria { NumExpediente }

Valor No Nulo { IDEmp }

Unico { IDEmp }

Clave Ajena { NumExpediente } hace referencia a Incidencias

Clave Ajena { IDEmp } hace referencia a Empresas

/\* Relacion Asocia: \*/

Asocia (IDEmp integer, IDImg integer, NumExpediente varchar )

Clave Primaria { IDEmp }

Valor No Nulo { IDImg, NumExpediente }

Unico { IDImg, NumExpediente }

Clave Ajena { IDEmp } hace referencia a Empresas por Obra

Clave Ajena { IDImg, NumExpediente } hace referencia a Obras

/\* Relacion Tiene: \*/

Tiene (IDImg integer, NumExpediente varchar )

Clave Primaria { IDImg }

Valor No Nulo { NumExpediente }

Unico { NumExpediente }

Clave Ajena { IDImg } hace referencia a Imagen de incidencia

Clave Ajena { NumExpediente } hace referencia a Incidencias

/\* Relacion Contiene: \*/

Contiene (NumExpediente varchar, NumExpediente varchar )

Clave Primaria { NumExpediente }

Valor No Nulo { NumExpediente }

Unico { IDImg, NumExpediente}

Clave Ajena { NumExpediente } hace referencia a Incidencias

Clave Ajena { NumExpediente } hace referencia a Obras

## Anexo II – Prototipos de baja fidelidad

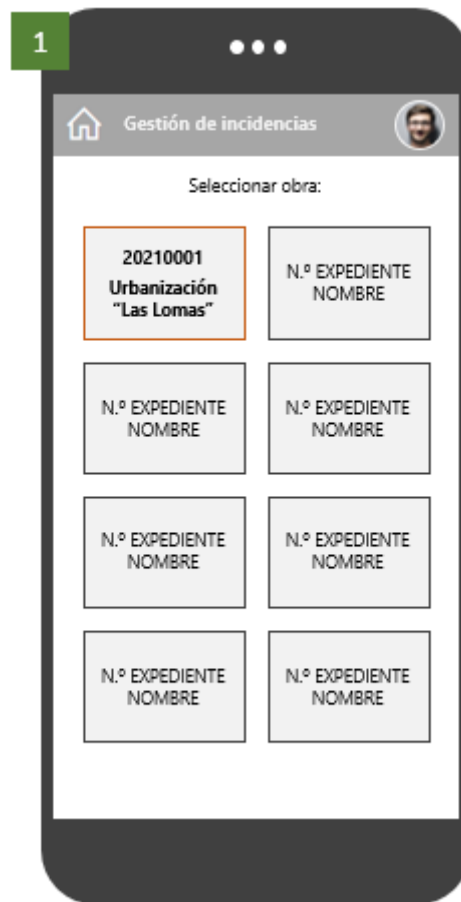


Figura 76. Pantalla de acceso a la aplicación



Figura 77. Pantalla principal de Obra

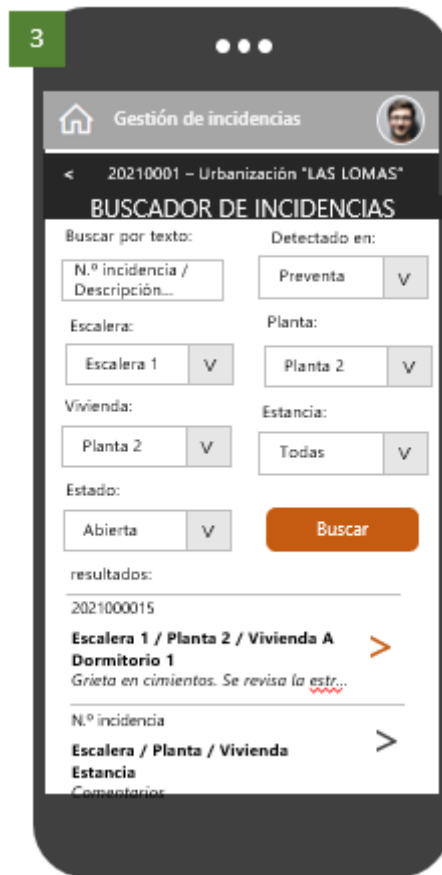


Figura 78. Pantalla de búsqueda de Incidencia

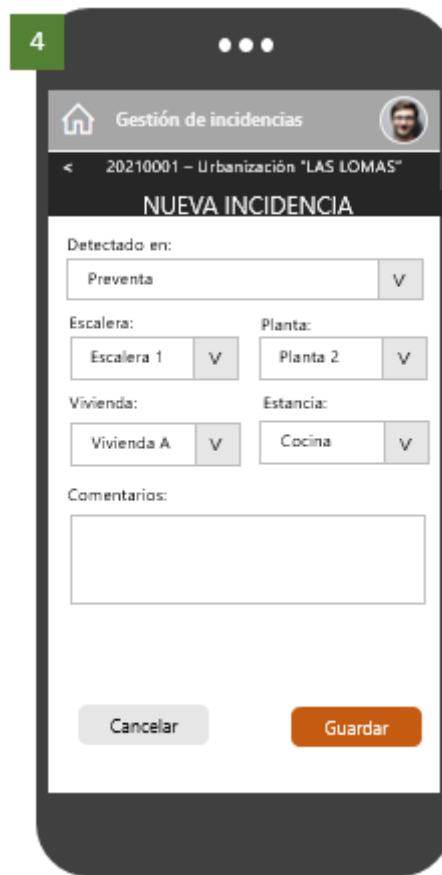


Figura 79. Pantalla de Alta de Incidencia

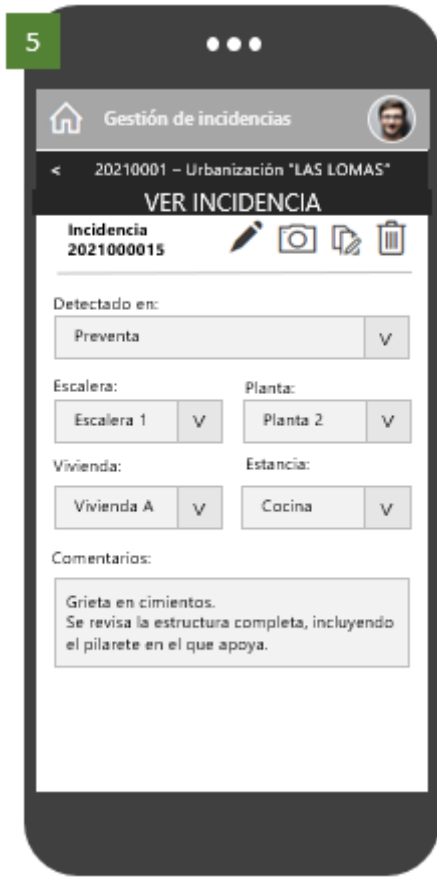


Figura 80. Pantalla de alta de Incidencia

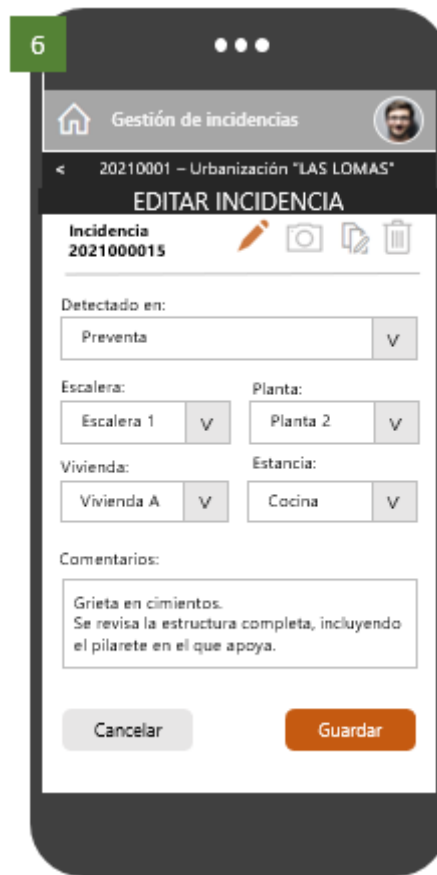


Figura 81. Pantalla de edición de Incidencia



Figura 82. Pantalla de Imágenes de Incidencia



Figura 83. Pantalla de agregación de imágenes



Figura 84. Pantalla de visualización de imagen

## Anexo III – Licencia Software y Documental

Las licencias software de las tecnologías utilizadas en este proyecto son:

Tenant de M365 (O365):

- Suscripción de Microsoft 365 Developer para crear soluciones independientes del entorno de producción empresarial.
- Suscripción de Microsoft 365 E5 Developer con 25 licencias de usuario.
- Tiene una duración de 90 días y es gratuita solo para fines de desarrollo (codificación de soluciones), con posibilidad de renovación (de forma automática por parte de MS siempre que se cumplan las condiciones de uso para entornos de desarrollo).
- Incluye servicios gratuitos de Azure.

Suscripción Azure (servicios de pago por consumo):

- Servicios Cognitivos para reconocimiento de imágenes
- Azure Functions para capturar eventos asociados a la biblioteca de imágenes

Visual Studio 2019:

- Versión Community - gratuita

Al igual que sucede en el mundo del software, se tienen que buscar formas de garantizar las libertades asociadas al trabajo elaborado y su inviolabilidad futura. Para garantizar que la libertad esté asociada al documento se buscan métodos, uno de ellos es la licencia GNU Free Documentation License (GFDL).



*Figura 85. Logotipo de la licencia GNU*

El propósito de esta Licencia es hacer que en el caso de este TFG sea "gratuito" en el sentido de libertad: para asegurar a todos la libertad efectiva de copiarlo y redistribuirlo, con o sin modificarlo, ya sea comercial o no comercialmente. En segundo lugar, esta licencia preserva para el autor y el editor una forma de obtener crédito por su trabajo, sin ser considerado responsable de las modificaciones realizadas por otros. Es una especie de "copyleft", lo que significa que las obras derivadas del documento deben ser libres en el mismo sentido.

Si por algún motivo se emplea este documento y se modifica, debe realizar una serie de acciones indicadas en el sitio web oficial de GNU [7]

Tampoco hay que olvidar que este documento, por defecto, está al amparo de la licencia



,por su inclusión en el Repositorio Institucional de Documentos de la Universidad de Zaragoza: ZAGUAN.