



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

Aplicación para la notificación y  
seguimiento de incidencias – Incident  
Tracker Service (ITS)

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Carlos Longares Alcalá

**Tutor:** Antonio Molina Marco

2018 - 2019



# Resumen

---

El objetivo de este proyecto es proporcionar herramientas a una entidad u organización para mejorar la gestión de incidencias que se produzcan en su entorno. El sistema permitirá a los usuarios o miembros de la organización la notificación de cualquier tipo de incidencia de una forma ágil y a los administradores su tramitación y seguimiento. El sistema ofrecerá también a los gestores información sobre la satisfacción de los usuarios, así como datos estadísticos del servicio. El sistema podría implantarse en distintos ámbitos, desde compañías o empresas a administraciones locales, y se desarrollará un caso particular contextualizado en la Universitat Politècnica de València. Aunque inicialmente se pretende gestionar incidencias relacionadas con las infraestructuras (averías, desperfectos, etc.) el sistema podrá ampliarse a otro tipo de incidencias relacionadas con distintos servicios de la organización, los procedimientos administrativos, o incluso como herramienta para recoger sugerencias de los usuarios.

El sistema constará de:

- Una aplicación móvil que permitirá a los usuarios dar parte de la incidencia, desde la ubicación donde se ha detectado, y conocer en todo momento el estado hasta su resolución.
- Una aplicación móvil que permitirá a los gestores la constatación y revisión de las incidencias en el lugar donde se produjeron.
- Una aplicación web que funcionara como panel o portal de administración, desde el cual se pueden administrar y resolver los distintos tipos de incidencias notificadas por los usuarios.

Las tecnologías que se pretenden utilizar para el desarrollo del sistema son:

- Aplicación móvil: Ionic (Framework basado en angular), SCSS, HTML y JavaScript.
- Aplicación web: Angular 7, SCSS, HTML y JavaScript.
- Backend: C# y Aspnet.

**Palabras clave:** Notificación de Incidencias, Seguimiento de Incidencias, programación móvil, programación web, satisfacción de usuarios.

# Abstract

---

The objective of this project is to provide tools to an entity or organization to improve the management of incidents that occur in their environment. The system will allow the users or members of the organization, the notification of any type of incident in an agile way and to the administrators its processing and follow-up. The system will also offer to managers information on user satisfaction, as well as statistical data of the service. The system could be implemented in different areas, from companies to local administrations, and a case will be developed contextualized in the *Universitat Politècnica de València*. Although initially it is intended to manage incidents related to infrastructures (breakdowns, damage, etc.) the system may be extended to other types of incidents related to different services of the organization, administrative procedures, or even as a tool to collect suggestions from users.

The system will consist of:

- A mobile application that will allow users to report the incident, from the location where it was detected, and to know the status of the incidence until it is resolved.
- A mobile application that will allow managers to verify and review incidents in the place where they occurred.
- A web application that will function as a panel or administration portal, from which you can manage and resolve the different types of incidents reported by users.

The technologies that are intended to be used for the development of the system are:

- Mobile application: Ionic (Framework based on angle), SCSS, HTML and JavaScript.
- Web application: Angular 7, SCSS, HTML and JavaScript.
- Backend: C # and Aspnet.

**Keywords:** Incident notification, incident tracking, mobile programming, web programming, user satisfaction.

# Resumen

---

L'objectiu d'aquest projecte és proporcionar eines a una entitat o organització per millorar la gestió d'incidències que es produeixin en el seu entorn. El sistema permetrà als usuaris o membres de l'organització la notificació de qualsevol tipus d'incidència d'una forma àgil i als administradors seva tramitació i seguiment. El sistema oferirà també als gestors informació sobre la satisfacció dels usuaris, així com dades estadístiques del servei. El sistema podria implantar-se a diferents àmbits, des de companyies o empreses a administracions locals, i es desenvoluparà un cas particular contextualitzat a la Universitat Politècnica de València. Encara que inicialment es pretén gestionar incidències relacionades amb les infraestructures (avaries, desperfectes, etc.) el sistema podrà ampliar-se a un altre tipus d'incidències relacionades amb diferents serveis de l'organització, els procediments administratius, o fins i tot com a eina per recollir suggeriments dels usuaris.

El sistema constarà de:

- Una aplicació mòbil que permetrà als usuaris donar part de la incidència, des de la ubicació on s'ha detectat, i conèixer en tot moment l'estat fins a la seva resolució.
- Una aplicació mòbil que permetrà als gestors la constatació i revisió de les incidències en el lloc on es van produir.
- Una aplicació web que funcionés com panell o portal d'administració, des del qual es poden administrar i resoldre els diferents tipus d'incidències notificades pels usuaris.

Les tecnologies que es pretenen utilitzar per al desenvolupament del sistema són:

- Aplicació mòbil: Ionic (Framework basat en angular), SCSS, HTML i JavaScript.
- Aplicació web: Angular 7, SCSS, HTML i JavaScript.
- Backend: C # i ASPNET.

**Paraules clau:** Notificació d'Incidències, Seguiment d'incidències, programació mòbil, programació web, satisfacció d'usuaris.

# Índice de contenidos

---

1	Introducción .....	10
1.1	Motivación.....	10
1.2	Objetivos .....	11
1.3	Metodología .....	11
1.4	Estructura.....	12
2	Estado del arte .....	13
2.1	Crítica al estado del arte.....	13
2.1.1	Service Desk .....	13
2.1.2	Zendesk.....	14
2.1.3	Jira.....	15
2.1.4	OTRS .....	16
2.1.5	ActivaLink .....	17
2.1.6	ServiceTonic .....	18
2.1.7	FreshService .....	19
2.2	Conclusiones .....	19
3	Alcance del proyecto .....	21
3.1	Ámbito.....	21
3.2	Objetivo .....	21
3.3	Especificación de requisitos funcionales.....	21
3.3.1	Aplicación web .....	21
3.3.2	Aplicación móvil .....	27
3.4	Especificación de requisitos no funcionales .....	32
3.5	Plan de trabajo.....	33
4	Análisis del problema .....	36
4.1	Diagrama de casos de uso.....	36
4.1.1	Especificación de casos de uso.....	37
4.2	Modelo de dominio .....	46
5	Diseño de la solución .....	48
5.1	Arquitectura del sistema .....	48
5.1.1	Backend y servicio de Web Api .....	48
5.1.2	Aplicación Web y Aplicación móvil .....	50

5.2 Diseño detallado .....	51
5.2.1 Backend y servicio de Web Api .....	51
5.2.2 Aplicación Web y Aplicación móvil .....	53
5.3 Tecnologías utilizadas .....	54
6 Desarrollo de la solución .....	56
6.1 Servicio Web Api.....	56
6.2 Backend .....	57
6.3 Aplicaciones basadas en Angular (Aplicación móvil y web) .....	58
7 Resultados .....	61
7.1 Aplicación Web .....	61
7.2 Aplicación móvil .....	63
8 Conclusiones .....	66
8.1 Relación del trabajo desarrollado con los estudios cursados.....	67
9 Referencias.....	68

## Índice de ilustraciones

---

Ilustración 1: Service desk aplicación web .....	13
Ilustración 2: Zendesk aplicación en la nube.....	14
Ilustración 3: Jira aplicación de escritorio.....	15
Ilustración 4: OTRS aplicación gratuita de escritorio .....	16
Ilustración 5: Activa Link aplicación de escritorio .....	17
Ilustración 6: ServiceTonic aplicación web .....	18
Ilustración 7: FreshService aplicación de escritorio.....	19
Ilustración 8 Diagrama de casos de uso .....	36
Ilustración 9 Modelo de dominio .....	46
Ilustración 10: Domain Driving Desing .....	48
Ilustración 11: Proyecto Domain .....	51
Ilustración 12: Proyecto Infrastructure.....	52
Ilustración 13: Proyecto Services .....	52
Ilustración 14: Proyecto Web Api .....	53
Ilustración 15: Carpeta Pages.....	54
Ilustración 16: Carpeta Shared .....	54
Ilustración 17: Carpeta Core .....	54
Ilustración 18: Pantalla de estadísticas .....	61

Ilustración 19: Pantalla con el listado de incidencias.....	62
Ilustración 20: Pantalla con la información de una incidencia.....	62
Ilustración 21: Login aplicación móvil.....	63
Ilustración 22: Listado de incidencias móvil .....	63

## Índice de tablas

---

Tabla 1 Requisito funcional: Autenticarse .....	22
Tabla 2 Requisito funcional: Crear cuenta de usuario.....	22
Tabla 3 Requisito funcional: Listar incidencias.....	23
Tabla 4 Requisito funcional: Actualizar incidencias.....	24
Tabla 5 Requisito funcional: Actualizar cuenta de usuario .....	24
Tabla 6 Requisito funcional: Listar histórico de incidencias.....	25
Tabla 7 Requisito funcional: Mostrar gráficas de incidencias .....	25
Tabla 8 Requisito funcional: Filtrar gráficas de incidencias .....	26
Tabla 9 Requisito funcional: Exportar listado de incidencias.....	26
Tabla 10 Requisito funcional: Autenticarse .....	27
Tabla 11 Requisito funcional: Registrar usuario.....	28
Tabla 12 Requisito funcional: Listar incidencias.....	28
Tabla 13 Requisito funcional: Actualizar incidencias.....	29
Tabla 14 Requisito funcional: Actualizar cuenta de usuario .....	30
Tabla 15 Requisito funcional: Modificar contraseña.....	30
Tabla 16 Requisito funcional: Solicitar contraseña nueva .....	31
Tabla 17 Requisito funcional: Crear incidencia .....	31
Tabla 18 Requisito funcional: Valorar incidencia.....	32
Tabla 19 Planificación de tareas .....	34
Tabla 20 Caso de uso: Crear cuenta de usuario .....	37
Tabla 21 Caso de uso: Solicitar contraseña nueva .....	38
Tabla 22 Caso de uso: Valorar incidencia.....	39
Tabla 23 Caso de uso: Actualizar incidencia.....	39
Tabla 24 Caso de uso: Listar incidencias.....	40
Tabla 25 Caso de uso: Crear incidencia .....	41
Tabla 26 Caso de uso: Autenticarse en la aplicación.....	41
Tabla 27 Caso de uso: Modificar contraseña .....	42



Tabla 28 Caso de uso: Actualizar cuenta de usuario .....	43
Tabla 29 Caso de uso: Listar histórico de incidencias.....	43
Tabla 30 Caso de uso: Exportar listado de incidencias a Excel .....	44
Tabla 31 Caso de uso: Exportar listado de incidencias a PDF .....	44
Tabla 32 Caso de uso: Filtrar gráficas de incidencias .....	45
Tabla 33 Caso de uso: Mostrar gráficas de incidencia.....	45



# 1. Introducción

---

En la actualidad, muchas empresas que disponen de un gran sistema de infraestructuras se ven obligadas a invertir una gran cantidad de recursos al mantenimiento de éstas. El mantenimiento de estas infraestructuras supone problemas a nivel de recursos y a nivel organizativo.

Los principales problemas que conlleva el mantenimiento de estas infraestructuras son:

- Llevar un seguimiento correcto de la incidencia, es decir, tener información detallada y actualizada desde que es reportada hasta que es resuelta.
- Tener un registro de cuántas, cuándo y cómo se producen las incidencias.
- Cómo asignar los recursos destinados al mantenimiento entre los distintos departamentos de la infraestructura.

Debido a los problemas mencionados anteriormente, estas empresas buscan herramientas de software que les faciliten la tramitación y resolución de incidencias. El problema de estas herramientas de software es que carecen de una interfaz amigable o cercana al usuario y a su vez son herramientas costosas de desplegar e instalar.

Por lo tanto, en este TFG se va a abordar la creación de unas herramientas, fáciles de desplegar e instalar, con un coste económico relativamente bajo y cuya tramitación de las incidencias sea lo más amigable posible.

## 1.1 Motivación

La motivación principal de este TFG es con carácter de auto superación, desde plantear la idea inicial hasta implementarla y llevar acabo.

La idea o la temática de un proyecto que ayude a el mejor mantenimiento de las infraestructuras de la universidad salió de que los alumnos no disponen de ninguna herramienta sencilla para el reporte de incidencias o incluso para sugerencias para mejorar las infraestructuras, por ejemplo, problemas de falta de material en aulas o goteras en las clases, no son fáciles de reportar o incluso no se pueden si no eres personal de la universidad. En resumen, sería poder dar facilidades a los usuarios de las infraestructuras para que estas sean más mantenibles.

## 1.2 Objetivos

Para resolver los problemas planteados en la introducción se han de cumplir lo siguientes objetivos:

- Ofrecer una herramienta para el reporte de incidencias, ágil e intuitiva a través de una aplicación móvil.
- Ayudar en tareas de logística, dado que será posible reportar faltas de material en las aulas.
- Facilitar el mantenimiento de las infraestructuras de la UPV.
- Agilizar la comunicación de incidencias, es decir conseguir que la información de la incidencia llegue lo antes posible al personal de administración o mantenimiento.
- Tener más detalles de cada incidencia mediante fotografías y la ubicación GPS de dónde se produjo esta, aportando así mayor calidad de información.
- Llevar un registro de todas las incidencias reportadas, tanto si han sido solucionadas o no.
- Ofrecer distintas estadísticas de incidencias, para saber si se están distribuyendo bien los recursos empleados en la resolución de estas.
- Permitir la valoración de una incidencia solucionada por parte de los usuarios.

## 1.3 Metodología

El trabajo o estructura del proyecto para cumplimentar los objetivos anteriormente propuestos, estará compuesto por tres partes:

- Una aplicación móvil en Android para el reporte de las incidencias.
- Una aplicación web en la que se administrarán las incidencias reportadas por la aplicación móvil.
- Un servicio web Api el cual suministrará la información tanto a la aplicación móvil como a la aplicación web.

El desarrollo de estas tres aplicaciones se llevará a cabo mediante una metodología ágil SCRUM, que se basa en una estructura de desarrollo incremental, es decir, cualquier ciclo de desarrollo del proyecto se desgrana o divide en distintos subproyectos cada uno con sus etapas de análisis, desarrollo y testeo. Estos subproyectos se denominan Sprint y en nuestro proyecto habrá tres.

## 1.4 Estructura

En este apartado se hablará de los distintos capítulos del proyecto. Comenzaremos hablando del estado de las actuales herramientas para la resolución de incidencias y para la gestión de éstas, hablaremos de sus características y también de los problemas a los que se enfrentan los usuarios al utilizarlas. A continuación se hará un análisis del problema, comentando el alcance de éste y la especificación tanto de los casos de uso como de los requisitos no funcionales. Después se verá la arquitectura de la solución, es decir, qué patrones y qué organización se ha llevado a cabo para la implementación. Una vez visto el diseño, procederemos con la implementación, donde se hace referencia a distintos fragmentos de código, que pueden ser interesantes comentar. Una vez explicados los detalles de la implementación, se verán los resultados, mediante varias imágenes de la aplicación móvil y de la aplicación web desarrolladas. Por último se valorarán los objetivos alcanzados y la experiencia obtenida al hacer este trabajo de fin de grado.

## 2. Estado del arte

Actualmente hay una gran infinidad de herramientas software que nos ayudan con la resolución de incidencias y con el mantenimiento de las infraestructuras de una empresa<sup>1</sup>. En este capítulo se hablará de las distintas alternativas más populares, resumiendo sus principales características, a continuación qué problemas conllevan el uso de estas herramientas y finalmente hablaremos de la propuesta de este proyecto y de cómo podría ayudar a solucionar los problemas descritos en el anterior apartado.

### 2.1 Crítica al estado del arte

En el siguiente apartado se va a hablar de 7 herramientas para la gestión y solución de incidencias en medianas y grandes empresas. La mayoría de estas herramientas cuentan con un cliente web o de escritorio. Sus funcionalidades básicas son: el registro de incidencias mediante un formulario, el seguimiento de las incidencias obligando a que éstas pasen por distintos estados, un listado de todas las incidencias pendientes y la notificación de nuevas incidencias al personal de mantenimiento. Describas las características principales, vamos a analizar las características específicas de las herramientas.

#### 2.1.1 Service Desk<sup>2</sup>

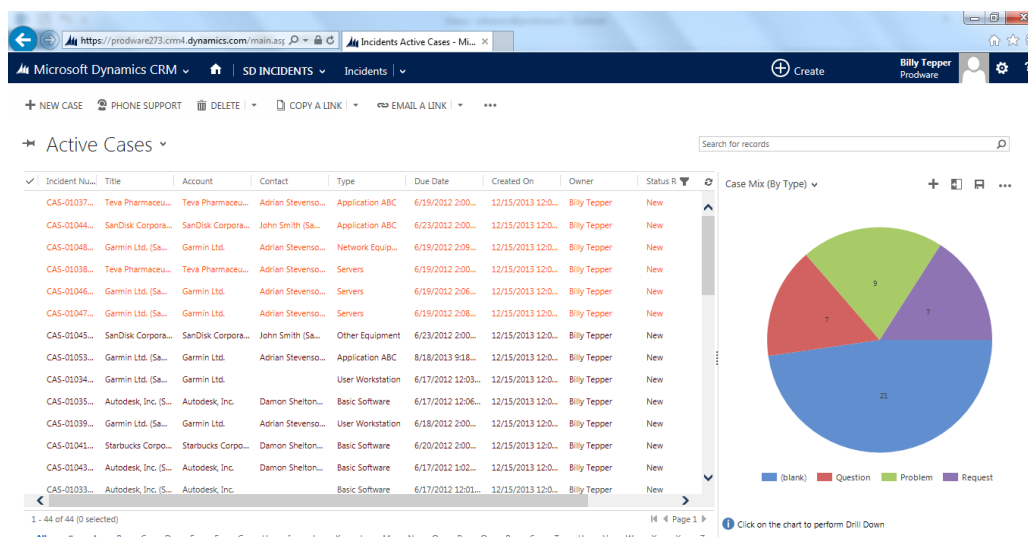


Ilustración 1: Service desk aplicación web<sup>3</sup>

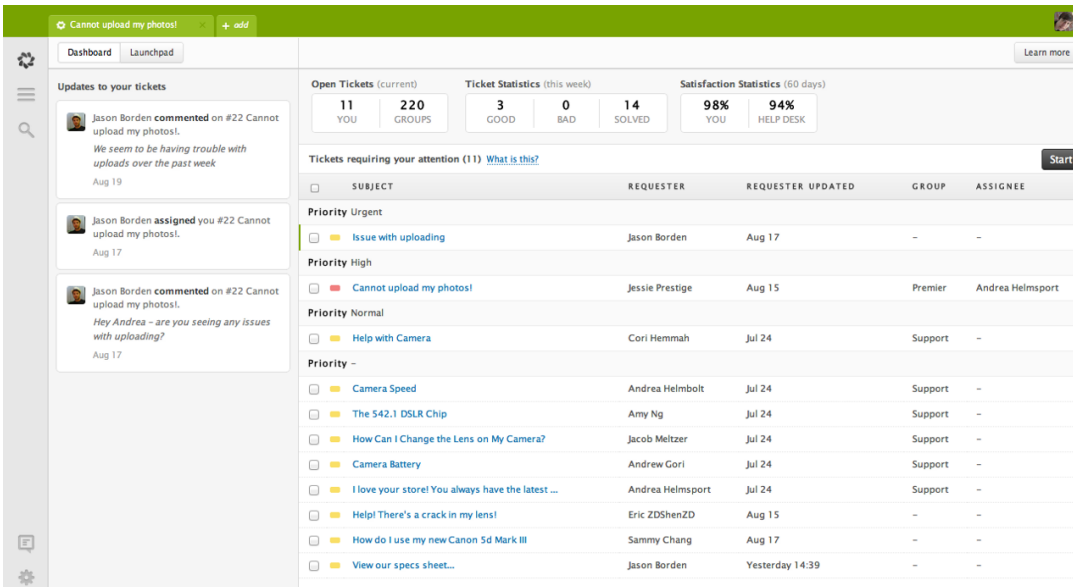
<sup>1</sup> <https://integriaims.com/las-mejores-herramientas-ticketing/>

<sup>2</sup> <https://www.dynamics-crm.es/crm-por-areas-de-gestion/service-desk>

<sup>3</sup> <https://community.dynamics.com/crm/b/microsoftdynamicscrmappstore/archive/2014/06/13/service-desk-solution-for-dynamics-crm-2013>

- Cuenta con una aplicación web (Ilustración 1), donde nos permite registrar las incidencias. Una vez dentro de la aplicación, estas incidencias serán guardadas y priorizadas, a su vez, podrán ser transmitidas directamente al personal técnico adecuado. También hay que añadir que las incidencias reportadas por correo electrónico serán registradas de manera automática en la aplicación web.
- Uno de los puntos fuerte de esta herramienta, es que evita la duplicidad de incidencias reportadas, dando la posibilidad de relacionar todas las incidencias que tengan que ver con el mismo problema, de esta manera, una vez sea solucionado el problema, todas estas incidencias relacionadas, serán cerradas al mismo tiempo.
- Otra característica importante de esta herramienta es que además de ayudarnos con el mantenimiento de las infraestructuras, nos ayuda con la renovación de éstas. Mediante el envío de autorizaciones y tiempos necesarios al personal encargado de estas renovaciones.
- La herramienta permite la creación de informes y graficas en tiempo real. Lo que nos permite calcular de una manera más eficiente la división de recursos destinados a los distintos departamentos.

## 2.1.2 Zendesk<sup>4</sup>



The screenshot displays the Zendesk dashboard interface. At the top, there's a navigation bar with 'Dashboard' and 'Launchpad' tabs. Below this, the main area is divided into several sections:

- Updates to your tickets:** A list of recent activity, including comments and assignments for ticket #22 'Cannot upload my photos!'.
- Open Tickets (current):** A summary showing 11 tickets for 'YOU' and 220 for 'GROUPS'.
- Ticket Statistics (this week):** A bar chart showing 3 GOOD, 0 BAD, and 14 SOLVED tickets.
- Satisfaction Statistics (60 days):** A bar chart showing 98% satisfaction for 'YOU' and 94% for 'HELP DESK'.
- Tickets requiring your attention (11):** A table listing tickets with columns for SUBJECT, REQUESTER, REQUESTER UPDATED, GROUP, and ASSIGNEE. The table is filtered by priority, showing Urgent, High, Normal, and Low priority tickets.

PRIORITY	SUBJECT	REQUESTER	REQUESTER UPDATED	GROUP	ASSIGNEE
Priority Urgent	Issue with uploading	Jason Borden	Aug 17	-	-
Priority High	Cannot upload my photos!	Jessie Prestige	Aug 15	Premier	Andrea Helmsport
Priority Normal	Help with Camera	Cori Hemmah	Jul 24	Support	-
Priority -	Camera Speed	Andrea Helmbolt	Jul 24	Support	-
Priority -	The 542.1 DSLR Chip	Amy Ng	Jul 24	Support	-
Priority -	How Can I Change the Lens on My Camera?	Jacob Meltzer	Jul 24	Support	-
Priority -	Camera Battery	Andrew Gori	Jul 24	Support	-
Priority -	I love your store! You always have the latest ...	Andrea Helmsport	Jul 24	Support	-
Priority -	Help! There's a crack in my lens!	Eric ZDShenZD	Aug 15	-	-
Priority -	How do I use my new Canon 5d Mark III	Sammy Chang	Aug 17	-	-
Priority -	View our specs sheet...	Jason Borden	Yesterday 14:39	-	-

Ilustración 2: Zendesk aplicación en la nube<sup>5</sup>

<sup>4</sup> <https://www.zendesk.es>

<sup>5</sup> <https://www.getapp.com/customer-service-support-software/a/zendesk/#gallery-1>

- Es una herramienta de software en la nube (Ilustración 2), con lo cual nos evita invertir recursos a la hora de instalar y desplegar la herramienta en el cliente. También hay que añadir que, al estar en la nube, nos obliga a estar conectados a Internet para poder acceder a la herramienta.
- El cliente o empresa puede personalizar su portal en Zendesk para la atención al cliente, este portal también cuenta con una página de auto ayuda y comunidades online. Cabe destacar que el tiempo y recursos ahorrados a la hora de no tener que desplegar la herramienta, tendrán que ser invertidos en tiempo de formación y personalización del portal que Zendesk nos ofrece.
- Ofrece funciones de chat directo con el servicio técnico.
- Los usuarios de Zendesk pueden utilizar la web para ver el estado de las incidencias incluso si ya están solucionadas. En el caso de que una incidencia ya resuelta, vuelva a ocurrir, tener la posibilidad de acceder a como se solucionó anteriormente, nos dará información relevante para evitar que esa incidencia vuelva a suceder.

### 2.1.3 Jira<sup>6</sup>

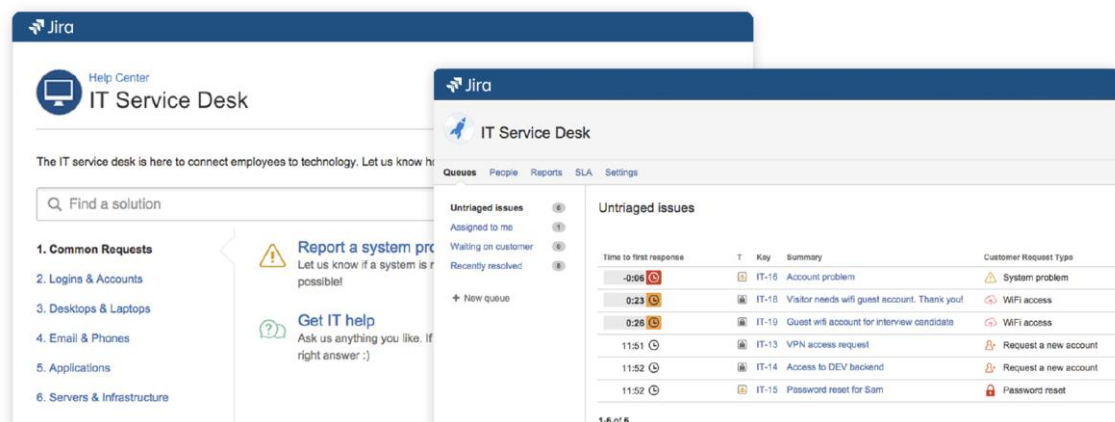


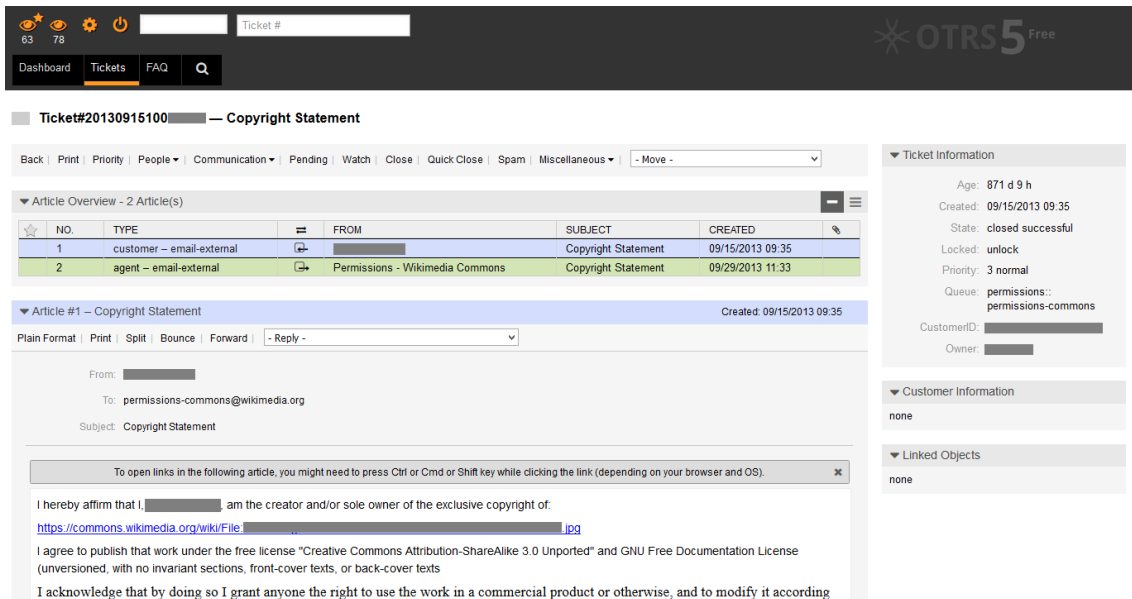
Ilustración 3: Jira aplicación de escritorio<sup>6</sup>

- Es un herramienta de gestión de incidencias para entornos con una gran sistema de infraestructuras o con una gran variedad de departamentos, con lo cual esta herramienta nos permite tener un portal de gestión de incidencias para cada departamento.
- Es una herramienta realmente flexible, ya que aparte de la herramienta en sí, nos permite instalar plugins con funcionalidades añadidas.

<sup>6</sup> <https://valiantys.com/en/products/jira-service-desk/>

- Una característica negativa de esta herramienta es que no permite la búsqueda de una incidencia por su número identificativo, algo que en la mayoría de las herramientas sí que está permitido.
- Permite la generación de informes en lenguaje SQL, permitiendo así a los usuarios crear sus propios Scripts para la obtención de los datos, pudiendo agilizar y automatizar informes de una manera más personalizada.

## 2.1.4 OTRS



The screenshot displays the OTRS5 Free web interface. At the top, there is a navigation bar with 'Dashboard', 'Tickets', and 'FAQ' tabs, and a search field. The main content area shows a ticket titled 'Ticket#20130915100 - Copyright Statement'. Below the title, there are action buttons like 'Back', 'Print', 'Priority', etc. A table lists two articles related to the ticket. The first article is selected, showing its details: 'Article #1 - Copyright Statement', created on 09/15/2013 09:35. The email content of this article is visible, starting with 'I hereby affirm that I, [redacted], am the creator and/or sole owner of the exclusive copyright of: https://commons.wikimedia.org/wiki/File:[redacted].jpg'. On the right side, there are panels for 'Ticket Information' (Age: 871 d 9 h, Created: 09/15/2013 09:35, State: closed successful, Priority: 3 normal) and 'Customer Information' (none).

Ilustración 4: OTRS aplicación gratuita de escritorio<sup>7</sup>

- Es una herramienta Open Source (Ilustración 4), al tener esta cualidad no acepta la integración de plugins, dado que el código es público y será la empresa cliente la encargada de modificar el código para añadir funcionalidades extra a la herramienta.
- Cuenta con una versión de pago OTR Business solution que es más profesional y con lo cual tiene más funcionalidades que la versión gratuita.
- Es una herramienta difícil de administrar, cuenta con paneles de control complejos, dado que la aplicación está dirigida a usuarios expertos.
- No permite la búsqueda de una incidencia directamente por su número identificativo.
- Es una herramienta con una funcionalidad muy reducida en comparación a herramientas similares, ya que no permite la generación de ningún tipo de informe, tampoco se pueden exportar las incidencias o tickets fuera de la

<sup>7</sup> [https://es.m.wikipedia.org/wiki/Archivo:OTRS\\_5\\_Wikimedia\\_\(en\).png](https://es.m.wikipedia.org/wiki/Archivo:OTRS_5_Wikimedia_(en).png)



aplicación y no lleva un seguimiento real de la incidencia ya que solo registra las fechas de creación y cierre de la incidencia.

## 2.1.5 ActivaLink<sup>8</sup>

The screenshot shows the ActivaLink desktop application interface. At the top, there is a header with 'Fundación Deportiva Municipal' and 'Reclamaciones'. Below the header is a navigation bar with buttons for 'Entrada', 'L. Técnico', 'Revisión L', 'I. Respuesta', 'Validación', 'Aprobación', 'Envío', and 'Cerrada'. The main area displays a table of claims with the following columns: 'Nº Ref. Reclamación', 'Asunto', 'Persona que reclama', 'Estado', 'Fecha de la incidencia', 'Validación', 'Enviado', 'Último día para responder', and 'Informe Respuesta'. The table contains 15 rows of data, each representing a different claim with its specific details.

Nº Ref. Reclamación	Asunto	Persona que reclama	Estado	Fecha de la incidencia	Validación	Enviado	Último día para responder	Informe Respuesta
17/00015	Falta de luz		Entrada Reclamación	23/10/2017				
17/00519	Limpieza y vestuarios		Entrada Reclamación	13/12/2017				
17/00542	Mantenimiento		Entrada Reclamación	21/11/2017			03/02/2018	
17/00056	Atención y pagos		Entrada Reclamación	07/11/2017			07/12/2017	
17/00068	cambios de horarios		Entrada Reclamación	27/11/2017			27/12/2017	
18/00078	Limpieza piscina		Entrada Reclamación	11/02/2018			25/03/2018	
18/00071	Robos		Entrada Reclamación	20/01/2018			23/03/2018	
18/00047	Temperatura		Entrada Reclamación	29/01/2018			14/03/2018	
18/00063	Vestuario y duchas		Entrada Reclamación	03/02/2018			21/03/2018	
17/00057	Campo de futbol, Nazaret		Entrada Reclamación	12/11/2017			12/12/2017	
17/00537	vestuarios		Entrada Reclamación	23/12/2017			25/01/2018	
17/00550	Carreras populares		Entrada Reclamación	20/11/2017			04/02/2018	
17/00502	DEVOLUCIÓN CURSO		Entrada Reclamación	22/11/2017			22/12/2017	

Ilustración 5: Activa Link aplicación de escritorio<sup>9</sup>

- Registro de la incidencia: Se produce por vía telemática o por correo electrónico, es decir el usuario no rellena ningún tipo de formulario, sino que es el operador el que se encarga de ello.
- Gestión de la incidencia: La incidencia una vez registrada tendrá que pasar por todos los estados especificados por la empresa cliente.
- Resolución de la incidencia: Una vez cerrada la incidencia los operadores se pondrán en contacto con el usuario que registro la incidencia.
- Es una herramienta OpenSource.

<sup>8</sup> <https://www.activalink.com/>

<sup>9</sup> <https://www.activalink.com/software-la-gestion-incidencias>

## 2.1.6 ServiceTonic<sup>10</sup>

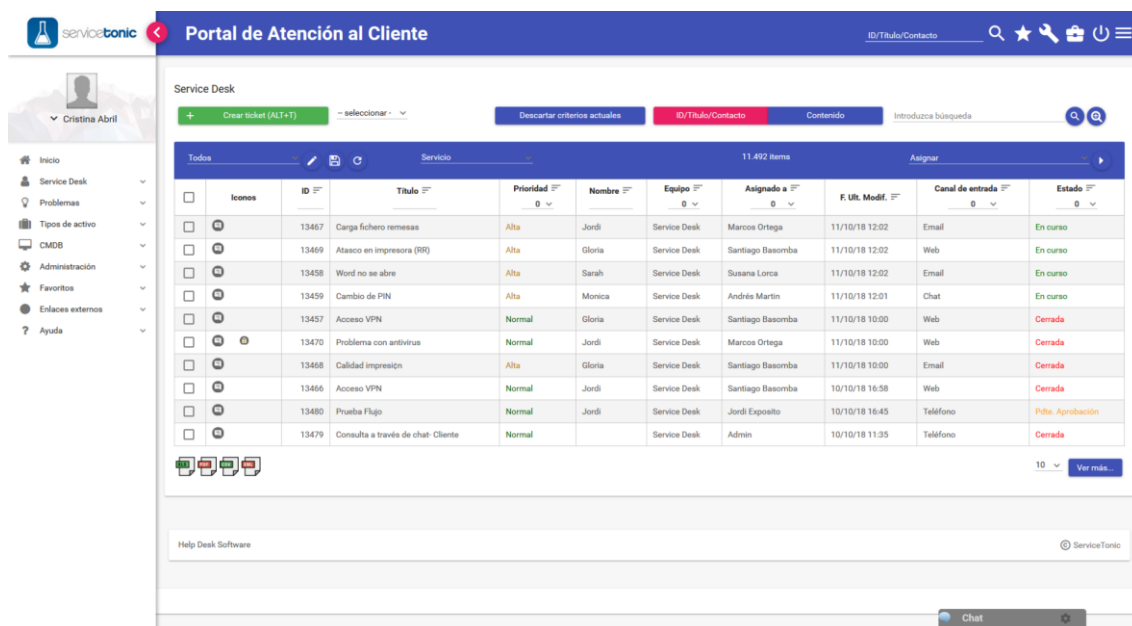
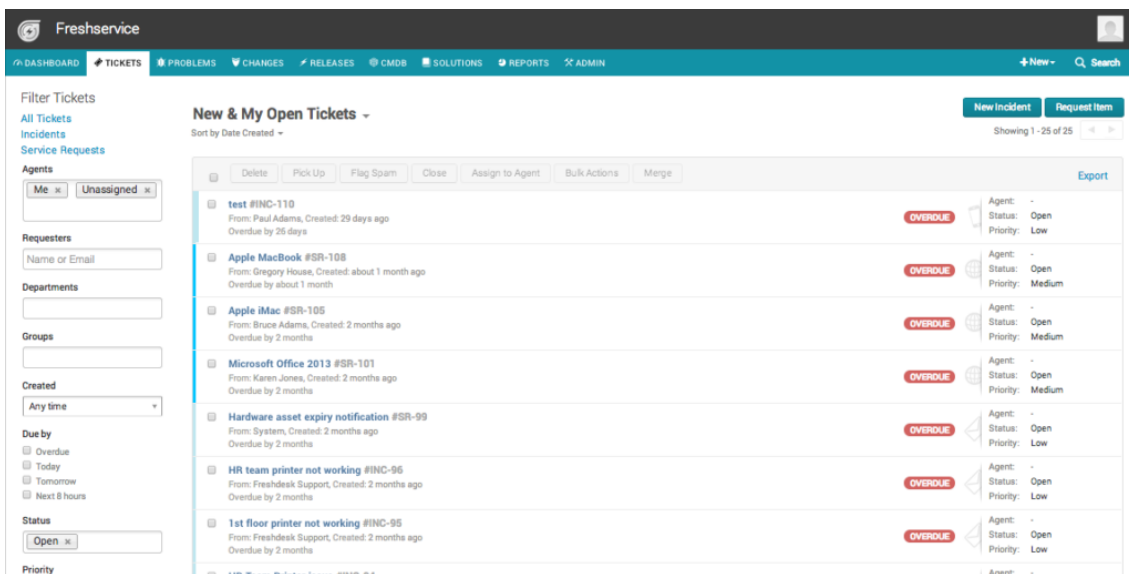


Ilustración 6: ServiceTonic aplicación web<sup>10</sup>

- Cuenta con una aplicación web (Ilustración 6) para el registro de incidencias. En el caso de que no se pueda hacer uso de esta herramienta se puede mantener una conversación con el personal técnico mediante el chat de la aplicación, para que esté sea el encargado de registrar la incidencias.
- La aplicación web es totalmente configurable, permitiendo al usuario acceder a información y documentos, crear y consultar tickets y consultar a la base de datos.
- Permite la generación de informes en formato PDF, Excel y HTML.
- Permite la monitorización de redes, servidores y hardware de red. Siendo más fácil registrar la disponibilidad y el rendimiento de los componentes IT de la infraestructura de la empresa cliente.
- A la hora de la gestión de las incidencias, permite a la empresa cliente definir los estados por los que tiene que pasar cada incidencia. Este punto es de gran relevancia, ya que permitirá una mejor adaptación de la herramienta en la empresa cliente.
- Cuando la incidencia se soluciona, permite al usuario la realización de una encuesta para conocer el nivel de satisfacción con respecto a la solución proporcionada. Esta característica nos puede proporcionar información relevante en cuanto a la gestión de los recursos entre los distintos departamentos.

<sup>10</sup> <https://www.servicetonic.es/help-desk-software/>

## 2.1.7 FreshService<sup>11</sup>



The screenshot displays the FreshService web application interface. The top navigation bar includes 'DASHBOARD', 'TICKETS', 'PROBLEMS', 'CHANGES', 'RELEASES', 'CMDB', 'SOLUTIONS', 'REPORTS', and 'ADMIN'. The main content area is titled 'New & My Open Tickets' and shows a list of tickets with columns for ticket ID, description, status, priority, and agent. The tickets listed include 'test #INC-110', 'Apple MacBook #SR-108', 'Apple iMac #SR-105', 'Microsoft Office 2013 #SR-101', 'Hardware asset expiry notification #SR-99', 'HR team printer not working #INC-95', and '1st floor printer not working #INC-95'. The interface also features a left sidebar with filters for agents, requesters, departments, groups, created dates, due by dates, status, and priority.

Ilustración 7: FreshService aplicación de escritorio<sup>11</sup>

- Permite la vinculación de varias incidencias registradas con un mismo problema.
- Gestiona los cambios en las infraestructuras, permitiendo a los usuarios mediante una aplicación web, evaluar y aprobar los cambios.
- Permite la automatización de tareas para la priorización, categorización, asignación y cierre de las incidencias registradas.
- Permite la generación de informes de manera personalizada.
- Crea respuestas predefinidas para responder a las consultas de manera más agilizada.

## 2.2 Conclusiones

Actualmente las herramientas de gestión, monitorización y solución de incidencias están muy extendidas por las grandes y medianas empresas, ya que ayudan al correcto mantenimiento de las distintas infraestructuras que éstas puedan tener. Pero estas herramientas tienen varios problemas a abordar:

- Si la herramienta es de escritorio, son difíciles de desplegar e instalar.
- Suelen ser herramientas complejas para el uso de usuarios no avanzados.
- La mayoría de estas herramientas no permiten al usuario afectado detallar con exactitud la incidencia, ya que en la mayoría de los casos será el operador quien la registre.

<sup>11</sup> <https://freshservice.com>

- Ninguna de ellas cuenta con una aplicación móvil, es decir para el registro de incidencias, se debe: llamar al servicio de mantenimiento para que registren ellos la incidencia, mandar un correo o utilizar la aplicación web o de escritorio en PC para solicitar la incidencia.
- Suelen ser de pago, algunas de las herramientas cuentan con una versión gratuita o OpenSource, pero con funcionalidad reducida o sin capacidad de actualizar la herramienta.
- Estas herramientas carecen de formularios con un diseño más intuitivo, suelen ser formularios complejos, poco *friendly-user*.

# 3. Alcance del proyecto

---

## 3.1 Ámbito

El objetivo de este proyecto es la creación de un sistema que permita reportar incidencias que ocurran dentro de una empresa con un gran sistema de infraestructuras. En este caso trabajaremos con la UPV (Universidad Politécnica de Valencia) como empresa de ejemplo.

## 3.2 Objetivo

El objetivo de este proyecto es el de crear una herramienta software para la gestión de incidencias, con un diseño más intuitivo, con la que usuarios no expertos puedan detallar con más exactitud los detalles de la incidencia. Para ello se creará una aplicación móvil, la cual permitirá al usuario registrar las incidencias directamente.

## 3.3 Especificación de requisitos funcionales

### 3.3.1 Aplicación web

#### 3.3.1.1 Administrador

<b>Autenticarse</b>	
Introducción	Permite a un administrador autenticarse en la aplicación web.
Entradas	Correo y contraseña
Proceso	Comenzando en la pantalla de login de la aplicación web, el administrador deberá introducir tanto su correo como la contraseña.

Salida	<ul style="list-style-type: none"> <li>• Quedar autenticado en la aplicación y redirigir al administrador a la pantalla principal de la aplicación web.</li> <li>• Mensaje de error en el caso de haber dejado vacío algún campo.</li> <li>• Mensaje de error si el correo electrónico o la contraseña no coinciden.</li> </ul>
--------	---

Tabla 1 Requisito funcional: Autenticarse

<b>Crear cuenta de usuario</b>	
Introducción	Crear la cuenta de un usuario desde la aplicación web.
Entradas	Nombre, apellidos, edad, teléfono, correo, contraseña y ubicación por defecto.
Proceso	Para cumplir con este requisito, empezaremos en la pantalla de login de la aplicación web, el administrador se autenticará, una vez dentro de la aplicación, aparecerá un menú a la izquierda, el cual una de sus opciones será ‘usuarios’, una vez dentro de la opción usuarios , la aplicación nos mostrará un listado de usuarios, pulsaremos sobre la opción añadir usuario y rellenaremos el formulario.
Salida	<ul style="list-style-type: none"> <li>• Nuevo usuario registrado en la base de datos e incluido en el listado de usuarios.</li> <li>• Mensaje de error en el caso de haber dejado vacío algún campo.</li> <li>• Mensaje de error si el correo electrónico ya existe en las base de datos de la aplicación.</li> </ul>

Tabla 2 Requisito funcional: Crear cuenta de usuario

<b>Listar incidencias</b>	
Introducción	Muestra una tabla con todas las incidencias reportadas actualmente.
Entradas	No hay entradas.
Proceso	El administrador deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal en la cual habrá un menú a la derecha con la opción 'Incidencias', al pulsar sobre éste, se le mostrará una tabla filtrada y paginada con todas las incidencias, ordenadas en orden descendente por fecha de última modificación.
Salida	<ul style="list-style-type: none"> <li>Listar todas las incidencias en formato de tabla.</li> </ul>

Tabla 3 Requisito funcional: Listar incidencias

<b>Actualizar incidencias</b>	
Introducción	Permite la modificación de los datos de una incidencia creada previamente desde la aplicación móvil.
Entradas	Nombre, descripción, propietario, ubicación, tipo, estado, imágenes adjuntas y posición GPS.
Proceso	El administrador deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal en la cual habrá un menú a la derecha con la opción 'Incidencias', al pulsar sobre éste, se le mostrar una tabla con el listado de incidencias, al pulsar sobre una incidencia, se le mostrar el formulario para poder modificar cualquier dato de ésta.

Salida	<ul style="list-style-type: none"> <li>• La incidencia deberá quedar modificada en la base de datos y actualizada en el listado que se muestra en la aplicación web.</li> <li>• Mensaje de error en el caso de que algún campo este vacío.</li> </ul>
--------	---

Tabla 4 Requisito funcional: Actualizar incidencias

<b>Actualizar cuenta de usuario</b>	
Introducción	Permite la modificación de los datos de una cuenta de usuario.
Entradas	Nombre, apellidos, edad, teléfono, correo, contraseña y ubicación por defecto.
Proceso	El administrador deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal en la cual habrá un menú a la derecha con la opción 'Usuarios, al pulsar sobre éste, se le mostrar una tabla con el listado de usuarios, al pulsar sobre una incidencia, se le mostrar el formulario para poder modificar cualquier dato de éste.
Salida	<ul style="list-style-type: none"> <li>• El usuario deberá quedar modificado en la base de datos y actualizado en el listado que se muestra en la aplicación web.</li> <li>• Mensaje de error en el caso de que algún campo este vacío.</li> </ul>

Tabla 5 Requisito funcional: Actualizar cuenta de usuario

<b>Listar histórico de incidencias</b>	
Introducción	Muestra una tabla con el histórico de incidencias gestionadas y reportadas.
Entradas	No hay entradas.



Proceso	El administrador deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal en la cual habrá un menú a la derecha con la opción 'Histórico incidencias', al pulsar sobre éste, se le mostrará una tabla filtrada y paginada con todas las incidencias, ordenadas en orden descendente por fecha de última modificación.
Salida	<ul style="list-style-type: none"> <li>Listar todas el histórico de incidencias en formato de tabla.</li> </ul>

Tabla 6 Requisito funcional: Listar histórico de incidencias

<b>Mostrar gráficas de incidencias</b>	
Introducción	Muestra un panel con distintas gráficas sobre las incidencias gestionadas y reportadas.
Entradas	No hay entradas.
Proceso	El administrador deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal en la cual se mostrarán 4 gráficas, dos de ellas de área mostrando cuantas incidencias se han reportado y cerrado por mes. Otras dos gráficas de tipo tarta mostrando cuantas incidencias registradas hay por tipo y por estado.
Salida	<ul style="list-style-type: none"> <li>Mostrar el panel principal con las 4 gráficas.</li> </ul>

Tabla 7 Requisito funcional: Mostrar gráficas de incidencias

<b>Filtrar gráficas de incidencias</b>	
Introducción	Permite filtrar las distintas gráficas del panel principal.
Entradas	No hay entradas.

Proceso	El administrador deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal en la cual se mostrarán 4 gráficas. Para el filtrado de estas deberá pulsar sobre la opción de la leyenda de la gráfica deseada a filtrar.
Salida	<ul style="list-style-type: none"> <li>Mostrará la gráfica filtrada e indicara de manera visual en la leyenda cual es el campo filtrado.</li> </ul>

*Tabla 8 Requisito funcional: Filtrar gráficas de incidencias*

<b>Exportar listado de incidencias</b>	
Introducción	Permite exportar la tabla de incidencias reportadas a formato PDF y Excel.
Entradas	No hay entradas.
Proceso	El administrador deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal en la cual habrá un menú a la derecha con la opción 'Incidencias'. Al pulsar sobre esta aparecerá una tabla con el listado de incidencias. Pulsar sobre el botón exportar.
Salida	<ul style="list-style-type: none"> <li>Generará un documento PDF o Excel con el listado de todas las incidencias mostradas en la tabla de la aplicación web.</li> </ul>

*Tabla 9 Requisito funcional: Exportar listado de incidencias*

### 3.3.2 Aplicación móvil

#### 3.3.2.1 Administrador y usuario

Por motivos de duplicidad se ha decidido usar los mismos requisitos funcionales en la aplicación móvil tanto para el administrador como para el usuario, ya que tienen permiso para hacer las mismas operaciones, la única diferencia es que el administrador podrá ver y editar cualquier incidencia, mientras que el usuario solo las que haya reportado.

<b>Autenticarse</b>	
Introducción	Permite a un administrador o a un usuario autenticarse en la aplicación móvil.
Entradas	Correo y contraseña
Proceso	Comenzando en la pantalla de login de la aplicación móvil, el administrador o usuario deberá introducir tanto su correo como la contraseña.
Salida	<ul style="list-style-type: none"><li>• Quedar autenticado en la aplicación y redirigir al administrador a la pantalla con el listado de incidencias de la aplicación móvil.</li><li>• Mensaje de error en el caso de haber dejado vacío algún campo.</li><li>• Mensaje de error si el correo electrónico o la contraseña no coinciden.</li></ul>

Tabla 10 Requisito funcional: Autenticarse

<b>Registrar usuario</b>	
Introducción	Permite a un nuevo usuario darse de alta en el sistema, a través de la aplicación móvil.

Entradas	Nombre, apellidos, edad, teléfono, correo, contraseña y ubicación por defecto.
Proceso	Para cumplir con este requisito, empezaremos en la pantalla de login de la aplicación móvil en la cual habrá una opción llamada 'Registrarse', accediendo a esta opción se le redirigirá al nuevo usuario a la pantalla con el formulario para darse de alta en el sistema.
Salida	<ul style="list-style-type: none"> <li>• Nuevo usuario registrado en la base de datos.</li> <li>• Mensaje de error en el caso de haber dejado vacío algún campo.</li> <li>• Mensaje de error si el correo electrónico ya existe en las base de datos de la aplicación.</li> </ul>

Tabla 11 Requisito funcional: Registrar usuario

<b>Listar incidencias</b>	
Introducción	Muestra una lista con todas las incidencias reportadas actualmente.
Entradas	No hay entradas.
Proceso	El administrador o usuario deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal de la aplicación, que es el listado de incidencias reportadas, de las cuales se mostrará una pequeña descripción, el estado, el tipo y la fecha de la última modificación, ordenadas en orden descendente por fecha de última modificación, .
Salida	<ul style="list-style-type: none"> <li>• Listar todas las incidencias en formato de lista.</li> </ul>

Tabla 12 Requisito funcional: Listar incidencias

<b>Actualizar incidencias</b>	
Introducción	Permite la modificación de los datos de una incidencia.
Entradas	Nombre, descripción, ubicación, tipo e imágenes adjuntas.
Proceso	El administrador o usuario deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal con el listado de incidencias, al seleccionar una incidencia se le mostrar un formulario con los campos de esta que son modificables.
Salida	<ul style="list-style-type: none"> <li>• La incidencia deberá quedar modificada en la base de datos y actualizada en el listado que se muestra en la aplicación móvil.</li> <li>• Mensaje de error en el caso de que algún campo este vacío.</li> </ul>

Tabla 13 Requisito funcional: Actualizar incidencias

<b>Actualizar cuenta de usuario</b>	
Introducción	Permite la modificación de los datos de una cuenta de usuario.
Entradas	Nombre, apellidos, teléfono, correo y ubicación por defecto.
Proceso	El administrador o usuario deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal, en la parte superior derecha habrá un botón para acceder a la información de la cuenta de usuario, una vez accedida a la información se mostrar un formulario con los campos a modificar.

Salida	<ul style="list-style-type: none"> <li>• El usuario deberá quedar modificado en la base de datos y su información en el formulario de la aplicación móvil.</li> <li>• Mensaje de error en el caso de que algún campo este vacío.</li> </ul>
--------	---

Tabla 14 Requisito funcional: Actualizar cuenta de usuario

<b>Modificar contraseña</b>	
Introducción	Permite la modificación de la contraseña de una cuenta de usuario.
Entradas	Antigua contraseña y nueva contraseña.
Proceso	El administrador o usuario deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal, en la parte superior derecha habrá un botón para acceder a la información de la cuenta de usuario. En esta información habrá un botón para modificar la contraseña, el cual nos conducirá a un formulario en el que se pedirá la antigua contraseña y la nueva contraseña.
Salida	<ul style="list-style-type: none"> <li>• La contraseña deberá quedar modificada y el sistema mostrará un mensaje de confirmación.</li> <li>• Mensaje de error en el caso de que algún campo este vacío</li> <li>• Mensaje de error en el caso de que la antigua contraseña no exista.</li> </ul>

Tabla 15 Requisito funcional: Modificar contraseña

<b>Solicitar contraseña nueva</b>	
Introducción	Permite la generación de una contraseña nueva de una cuenta de usuario.

Entradas	Correo.
Proceso	Para la cumplimentación de este requisito bastara con estar en la pantalla de login, rellenar el campo correo y pulsar sobre una opción ‘¿Olvidaste tu contraseña?’.
Salida	<ul style="list-style-type: none"> <li>• Se generará una contraseña nueva para el usuario del correo y se mostrará por parte del sistema un mensaje de confirmación.</li> <li>• Mensaje de error si el correo no existe.</li> </ul>

Tabla 16 Requisito funcional: Solicitar contraseña nueva

<b>Crear incidencia</b>	
Introducción	Permite la creación o reporte de una incidencia.
Entradas	Descripción, ubicación, tipo, imágenes adjuntas y posición GPS.
Proceso	El administrador o usuario deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal con el listado de incidencias, en la cual habrá una opción ‘+’ que conducirá al usuario al formulario de la creación de incidencias.
Salida	<ul style="list-style-type: none"> <li>• La incidencia deberá quedar registrada en la base de datos y actualizado el listado que se muestra en la aplicación móvil con la nueva incidencia.</li> <li>• Mensaje de error en el caso de que algún campo este vacío.</li> </ul>

Tabla 17 Requisito funcional: Crear incidencia

<b>Valorar incidencia</b>	
Introducción	Permite la valoración de una incidencia.
Entradas	Puntuaciones en cuanto a velocidad y eficiencia de la solución suministrada.
Proceso	El administrador o usuario deberá autenticarse, una vez dentro de la aplicación, se le redirigirá a la pantalla principal con el listado de incidencias, saltará un dialogo por parte del sistema si existe alguna incidencia en el listado con el estado 'pendiente de valoración'. Esta dialogo ofrecerá la opción de valorar alguna de estas incidencias y mostrar un formulario con los aspectos a valorar de ésta.
Salida	<ul style="list-style-type: none"> <li>• La incidencia deberá quedar valorada en la base de datos y actualizado el listado que se muestra en la aplicación móvil sin la incidencia recién valorada.</li> <li>• Mensaje de error en el caso de que algún campo este vacío.</li> </ul>

Tabla 18 Requisito funcional: Valorar incidencia

### 3.4 Especificación de requisitos no funcionales

- La aplicación donde queda el registro de todas las incidencias ha de ser web.
- La aplicación para el propio reporte de incidencias tiene que ser para móviles.
- La aplicación móvil debe funcionar en la versión de Android Lollipop v 5.0 y superiores.
- Los tiempos de carga de la aplicación web no deben superar los 15 segundos.
- Abstraerse a la LOPD con todos los datos de los usuarios.
- La aplicación web debe funcionar en las últimas versiones de cada uno de los navegadores más utilizados.
- La aplicación web ha de ser *Responsive*. La interfaz ha de adaptarse a los distintos tamaños y resoluciones de los distintos dispositivos.



### 3.5 Plan de trabajo

El proyecto está dividido en tres subproyectos, el primer proyecto esta orientado a la creación de servicio web api y un Backend el cual suministre la información a los otros dos subproyectos. El segundo subproyecto es la aplicación web, orientada a la administración de las incidencias. Como ultimo subproyecto tenemos la aplicación móvil, que estará mas orientada al reporte de las incidencias.

A continuación, se muestra el plan de trabajo seguido para la implementación de estos tres subproyectos. Para ello se han planificado tres Sprints de tres semanas cada uno. Veamos mas en detalle cada uno.

Tarea	Sprint	Fecha inicio	Fecha fin
Creación de la BDD	1	28/01/2019	30/01/2019
Implementación del ORM con Entity Framework	1	31/01/2019	03/02/2019
Listar incidencias en la aplicación web	1	04/02/2019	07/02/2019
Mostrar incidencia en la aplicación web	1	08/02/2019	10/02/2019
Listar registros de incidencias	1	11/02/2019	14/02/2019
Actualizar incidencia en la aplicación web	1	15/02/2019	16/02/2019
Exportar incidencias en formato PDF y Excel	1	17/02/2019	18/02/2019
Listar incidencias en la aplicación móvil	2	18/02/2019	20/02/2019
Crear incidencias en la aplicación móvil	2	21/02/2019	22/02/2019
Mostrar incidencia en la aplicación móvil	2	23/02/2019	25/02/2019
Actualizar incidencia en la aplicación móvil	2	26/02/2019	28/02/2019
Crear usuario	2	01/03/2019	03/03/2019



Mostrar cuenta de usuario	2	04/03/2019	05/03/2019
Modificar contraseña de usuario	2	06/03/2019	07/03/2019
Autenticarse en la aplicación móvil y web	2	08/03/2019	10/03/2019
Registro de usuario en la aplicación móvil	3	11/03/2019	13/03/2019
Solicitar contraseña nueva en la aplicación móvil	3	14/03/2019	15/03/2019
Actualizar información del perfil de la cuenta en la aplicación móvil	3	14/03/2019	15/03/2019
Modificar contraseña en la aplicación móvil	3	16/03/2019	17/03/2019
Valorar incidencia	3	18/03/2019	20/03/2019
Mostrar gráficas de estadísticas de incidencias	3	21/03/2019	23/03/2019
Filtrar gráficas de estadísticas de incidencias	3	24/03/2019	26/03/2019

Tabla 19 Planificación de tareas

Como se puede apreciar en la tabla 15, las tareas para el Sprint 1 están mas enfocadas en la aplicación web, pero dado que es el primer Sprint, es imprescindible crear la base de datos e implementar el ORM para el acceso a esta. Cabe destacar que cada tarea enfocada a la aplicación web tiene como subapartado el desarrollo de un servicio web api que lo dote de la información necesaria.

Desde un punto de vista mas personal, las tareas relacionadas con el desarrollo de la aplicación web me fueron más llevaderas y más fáciles de implementar. En cuanto a la implementación del ORM, invertí más tiempo formándome que en el propio desarrollo de este. En resumen, las tareas mas pesadas o costosas de este Sprint fueron: la implementación del ORM, el listado en la aplicación web de las incidencias, y la actualización de una incidencia en concreto.

El segundo Sprint esta mas enfocado en el desarrollo de la aplicación móvil, en este sprint ya no se contemplan tareas en sí de desarrollo del ORM, sino que pasan a ser

subapartados de las tareas. Cabe destacar que al igual que en el sprint anterior cada tarea tiene también su propio subapartado para el desarrollo del web api.

El tercer y último Sprint está enfocado a tareas de autenticación y mantenimiento de las cuentas de los usuarios, también se han incluido diversas tareas para el mostrado de gráficas y estadísticas y una última tarea para la valoración de incidencias resueltas.

## 4. Análisis del problema

### 4.1 Diagrama de casos de uso



Ilustración 8 Diagrama de casos de uso <sup>12</sup>

Se han identificado dos tipos de actores: usuario y administrador, como se puede apreciar en la ilustración 1, el actor usuario podrá gestionar su cuenta y las incidencias reportadas, estas acciones solo podrán ser llevadas a cabo desde la aplicación móvil, por otro lado, el actor administrador además de poder realizar las mismas acciones que un usuario normal, podrá usar la aplicación web para acceder a históricos y graficas con distintas estadísticas sobre las incidencias.

<sup>12</sup> <https://www.draw.io/>

### 4.1.1 Especificación de casos de uso

<b>Caso de uso: Crear cuenta de usuario</b>	
Resumen	Crea una nueva cuenta de usuario desde la aplicación móvil, para poder hacer uso de ésta.
Actor	Usuario, Administrador
Precondición	<ul style="list-style-type: none"><li>• Tener acceso al servicio Web Api de la aplicación.</li><li>• El correo introducido debe existir en el dominio de la universidad.</li><li>• El correo introducido no debe estar en uso por otra cuenta.</li></ul>
Descripción	Flujo Básico: <ol style="list-style-type: none"><li>1. Pulsar el botón 'Registrarse' en la pantalla de login de la aplicación web.</li><li>2. Rellenar todos los campos del formulario.</li><li>3. Pulsar el botón 'Crear cuenta'.</li></ol>
Postcondición	El sistema enviará un correo electrónico al usuario confirmando la creación de la cuenta.

Tabla 20 Caso de uso: Crear cuenta de usuario

<b>Caso de uso: Solicitar contraseña nueva</b>	
Resumen	Genera una contraseña nueva para el usuario desde la aplicación móvil.
Actor	Usuario, Administrador
Precondición	<ul style="list-style-type: none"><li>• Tener acceso al servicio Web Api de la aplicación.</li><li>• El correo electrónico introducido ha de existir en la base de datos de la aplicación.</li></ul>

Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación móvil.</li> <li>2. Rellenar el campo 'Correo'.</li> <li>3. Pulsar el botón '¿Olvidaste tu contraseña?'.</li> <li>4. Pulsar Aceptar en el dialogo de confirmación que se mostrará por pantalla.</li> </ol>
Postcondición	El sistema enviará un correo electrónico con la nueva contraseña generada.

Tabla 21 Caso de uso: Solicitar contraseña nueva

<b>Caso de uso: Valorar incidencia</b>	
Resumen	Se mostrará una breve encuesta al usuario, para que éste valore la resolución de la incidencia que reportó.
Actor	Usuario
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• Estar autenticado en la aplicación.</li> <li>• La incidencia para valorar debe estar en estado 'pendiente de valoración'.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación móvil.</li> <li>2. Autenticarse en la aplicación.</li> <li>3. Una vez se muestre el listado de incidencias, si alguna está en estado 'pendiente de valoración', se mostrará un mensaje por pantalla informando al usuario de que hay incidencias pendientes de valoración.</li> <li>4. Listar incidencias pendientes de valoración.</li> <li>5. Escoger incidencia a valorar.</li> <li>6. Complimentar la encuesta rellenando todos los campos.</li> <li>7. Pulsar sobre el botón 'Valorar'.</li> </ol>
Postcondición	La incidencia quedará valorada y desaparecerá del listado de incidencias del usuario.

Tabla 22 Caso de uso: Valorar incidencia

<b>Caso de uso: Actualizar incidencia</b>	
Resumen	Tener acceso a la información de una incidencia para poder modificar alguno de sus datos.
Actor	Usuario, Administrador
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• Estar autenticado en la aplicación.</li> <li>• La incidencia ha de tener como mínimo o bien una descripción o alguna foto adjunta.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación móvil.</li> <li>2. Autenticarse en la aplicación.</li> <li>3. Listar incidencias reportadas.</li> <li>4. Escoger la incidencia a modificar de la lista.</li> <li>5. Modificar la información de la incidencia.</li> <li>6. Pulsar sobre el botón 'Guardar cambios'.</li> </ol>
Postcondición	La fecha de última modificación de la incidencia será modificada a la fecha actual.

Tabla 23 Caso de uso: Actualizar incidencia

<b>Caso de uso: Listar incidencias</b>	
Resumen	Muestra una lista de incidencias, de las cuales se muestra una pequeña descripción, el estado, el tipo y la fecha de la última modificación.
Actor	Usuario, Administrador
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• Estar autenticado en la aplicación.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación móvil.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Autenticarse en la aplicación.</li> <li>3. Listar incidencias reportadas.</li> </ol> <p>Flujo alternativo:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación web.</li> <li>2. Autenticarse en la aplicación (solo los administradores podrán).</li> <li>3. Listar incidencias reportadas.</li> </ol>
Postcondición	<p>Si el actor es usuario solo podrá ver sus incidencias.</p> <p>Si el actor es administrador podrá ver todas las incidencias de todos los usuarios.</p>

Tabla 24 Caso de uso: Listar incidencias

<b>Caso de uso: Crear incidencia</b>	
Resumen	Crea y reporta una incidencia, obteniendo la posición GPS en el momento de registrarla. Como datos se aportarán: título, descripción, escuela/zona, tipo y fotografías.
Actor	Usuario, Administrador
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• Estar autenticado en la aplicación.</li> <li>• La incidencia ha de tener como mínimo o bien una descripción o alguna foto adjunta.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación móvil.</li> <li>2. Autenticarse en la aplicación.</li> <li>3. Listar incidencias reportadas.</li> <li>4. Pulsar sobre el botón '+'.                             <ol style="list-style-type: none"> <li>5. Rellenar todos los campos del formulario de la nueva incidencia.</li> </ol> </li> <li>6. Pulsar sobre el botón 'Crear incidencia'.</li> </ol>



Postcondición	Se mostrará el listado de incidencias, con la nueva incidencia incluida, esta incidencia tendrá como fecha de última modificación la fecha actual.
---------------	--

Tabla 25 Caso de uso: Crear incidencia

<b>Caso de uso: Autenticarse en la aplicación</b>	
Resumen	Permite al usuario o administrador autenticarse en la aplicación a través de sus credenciales.
Actor	Usuario, Administrador
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• El correo y la contraseña deben existir en la base de datos.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación móvil.</li> <li>2. Rellenar los campos 'Correo' y 'Contraseña'.</li> <li>3. Pulsar el botón 'Login'.</li> </ol> <p>Flujo alternativo:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación web (solo administradores).</li> <li>2. Rellenar los campos 'Correo' y 'Contraseña'.</li> <li>3. Pulsar el botón 'Login'.</li> </ol>
Postcondición	Si las credenciales son correctas, el usuario quedará autenticado y se le mostrará el listado de incidencias.

Tabla 26 Caso de uso: Autenticarse en la aplicación

<b>Caso de uso: Modificar contraseña</b>	
Resumen	Permite al usuario o administrador cambiar su contraseña actual.
Actor	Usuario, Administrador
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> </ul>

	<ul style="list-style-type: none"> <li>• Estar autenticado en la aplicación.</li> <li>• Su antigua contraseña debe existir en la base de datos.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación móvil.</li> <li>2. Autenticarse en la aplicación.</li> <li>3. Listar incidencias reportadas.</li> <li>4. Pulsar sobre el botón con el icono de usuario ubicado en la parte superior derecha de la aplicación.</li> <li>5. Pulsar sobre la opción 'Mi cuenta'.</li> <li>6. Pulsar sobre el botón 'Modificar contraseña'.</li> <li>7. Rellenar formulario con la antigua contraseña y la nueva.</li> <li>8. Pulsar sobre el botón 'Modificar contraseña'.</li> <li>9. Pulsar sobre la opción 'Aceptar' en el dialogo de confirmación.</li> </ol>
Postcondición	Se volverá a mostrar la pantalla con la información de la cuenta del usuario y se mostrará un mensaje de confirmación.

Tabla 27 Caso de uso: Modificar contraseña

<b>Caso de uso: Actualizar cuenta de usuario</b>	
Resumen	Permite al usuario acceder a su cuenta de usuario para así poder modificar información de esta.
Actor	Usuario, Administrador
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• Estar autenticado en la aplicación.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación móvil.</li> <li>2. Autenticarse en la aplicación.</li> <li>3. Listar incidencias reportadas.</li> <li>4. Pulsar sobre el botón con el icono de usuario ubicado en la parte superior derecha de la aplicación.</li> </ol>

	<ol style="list-style-type: none"> <li>5. Pulsar sobre la opción 'Mi cuenta'.</li> <li>6. Modificar información del formulario de la cuenta.</li> <li>7. Pulsar sobre el botón 'Guardar cambios'.</li> </ol>
Postcondición	Se volverá a mostrar la pantalla de la cuenta de usuario con la nueva información de la cuenta y se mostrará un mensaje de confirmación.

Tabla 28 Caso de uso: Actualizar cuenta de usuario

<b>Caso de uso: Listar histórico de incidencias</b>	
Resumen	Muestra una lista de incidencias ya cerradas o solucionadas, de las cuales se muestra una pequeña descripción, el estado, el tipo y la fecha de la última modificación.
Actor	Administrador
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• Estar autenticado en la aplicación.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación web.</li> <li>2. Autenticarse en la aplicación.</li> <li>3. Pulsar sobre el botón 'Histórico de incidencias'.</li> </ol>
Postcondición	Mostrará una tabla paginada con las 10 primeras incidencias, en orden descendiente por fecha de última modificación.

Tabla 29 Caso de uso: Listar histórico de incidencias

<b>Caso de uso: Exportar listado de incidencias a Excel</b>	
Resumen	Permite al administrador obtener un documento en formato Excel de todas las incidencias registradas en el sistema.
Actor	Administrador

Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• Estar autenticado en la aplicación.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación web.</li> <li>2. Autenticarse en la aplicación.</li> <li>3. Pulsar sobre el botón 'Incidencias'</li> <li>4. Pulsar sobre el botón 'Exportar a Excel'</li> </ol>
Postcondición	Generará un documento Excel con todas las incidencias registradas en el sistema.

Tabla 30 Caso de uso: Exportar listado de incidencias a Excel

<b>Caso de uso: Exportar listado de incidencias a PDF</b>	
Resumen	Permite al administrador obtener un documento en formato PDF de todas las incidencias registradas en el sistema.
Actor	Administrador
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• Estar autenticado en la aplicación.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación web.</li> <li>2. Autenticarse en la aplicación.</li> <li>3. Pulsar sobre el botón 'Incidencias'</li> <li>4. Pulsar sobre el botón 'Exportar a PDF'</li> </ol>
Postcondición	Generará un documento PDF con todas las incidencias registradas en el sistema.

Tabla 31 Caso de uso: Exportar listado de incidencias a PDF

<b>Caso de uso: Filtrar gráficas de incidencias</b>	
Resumen	Permite al administrador filtrar las distintas gráficas que hay en el pantalla principal de la aplicación web.
Actor	Administrador
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• Estar autenticado en la aplicación.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación web.</li> <li>2. Autenticarse en la aplicación.</li> <li>3. Pulsar sobre cualquiera de las leyendas de las gráficas para que éstas se filtren por los distintos atributos que tiene.</li> </ol>
Postcondición	La opción de la leyenda por la cual se haya filtrado deberá tener alguna indicación visual, de manera que el usuario sepa que la gráfica esta filtrada por esa opción.

Tabla 32 Caso de uso: Filtrar gráficas de incidencias

<b>Caso de uso: Mostrar gráficas de incidencias</b>	
Resumen	Permite al administrador ver varias gráficas sobre las incidencias reportadas y registradas.
Actor	Administrador
Precondición	<ul style="list-style-type: none"> <li>• Tener acceso al servicio Web Api de la aplicación.</li> <li>• Estar autenticado en la aplicación.</li> </ul>
Descripción	<p>Flujo básico:</p> <ol style="list-style-type: none"> <li>1. Acceder a la pantalla de login de la aplicación web.</li> <li>2. Autenticarse en la aplicación.</li> <li>3. Mostrar la pantalla principal de la aplicación web con las distintas gráficas.</li> </ol>

Tabla 33 Caso de uso: Mostrar gráficas de incidencia

## 4.2 Modelo de dominio

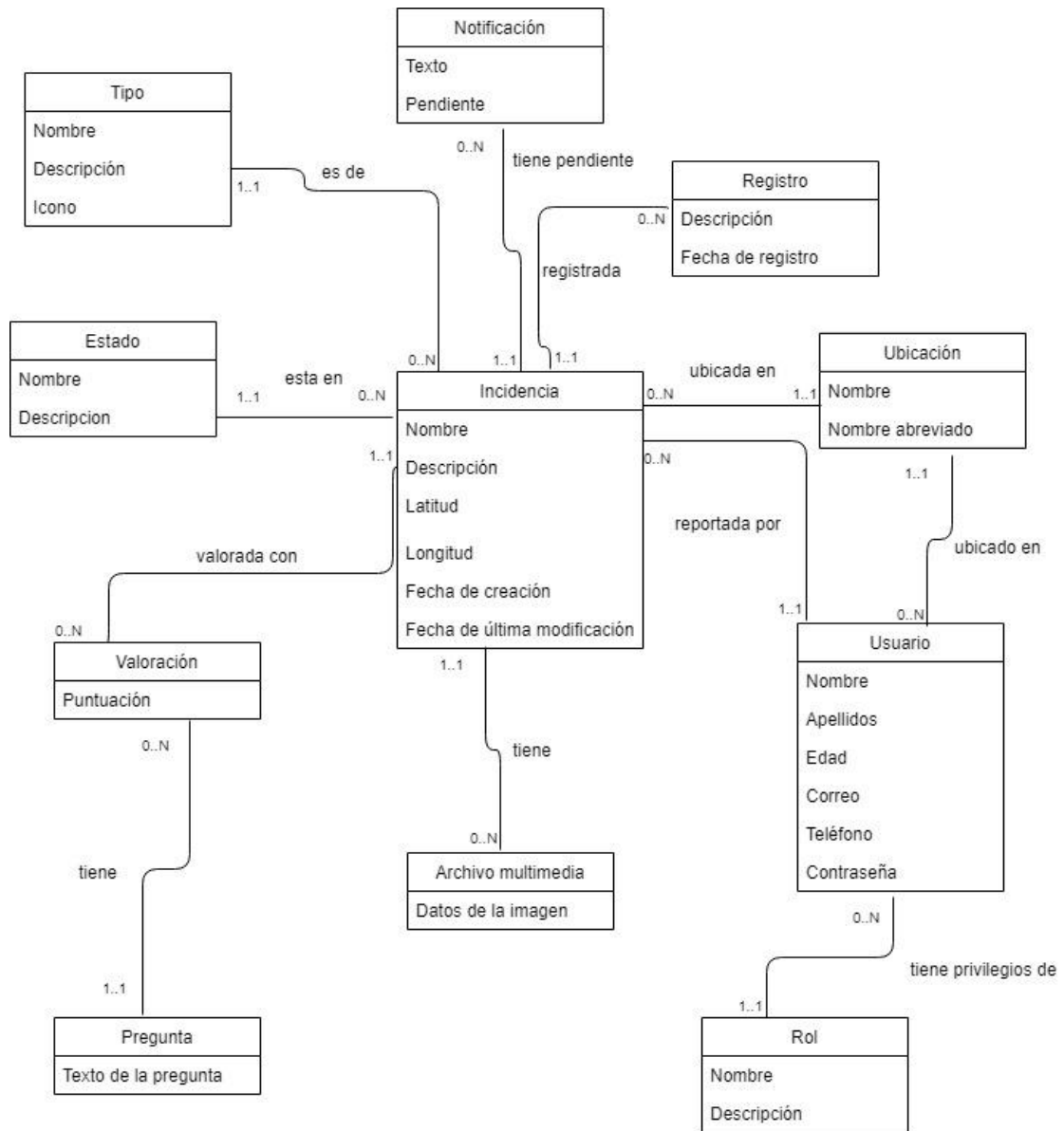


Ilustración 9 Modelo de dominio<sup>8</sup>

Como se puede apreciar en la Ilustración 2, nuestro modelo de dominio está compuesto por 11 clases. Este tipo de modelos nos ayudan a abstraer la información relevante de nuestro dominio en forma de entidades y también nos ayudan a apreciar cómo están relacionadas entre sí. Por otro lado, siempre es interesante matizar algunos aspectos más específicos de nuestro dominio.

<sup>8</sup> <https://www.draw.io/>

## Matizaciones

- Las entidades tipo, estado, pregunta y rol podrían haberse incluido como parte de otra entidad, pero dado que van a tener cierto nivel de complejidad, se ha optado por crear una entidad aparte para cada una, de esta manera evitaremos información duplicada.
- La relación que existe entre las entidades incidencia y registro, que es una a muchos, puede ser mal interpretada, lo que se busca con esta relación, es que a cada cambio que sufra la incidencia, ya sea la propia creación de la incidencia o la actualización de algún dato de ésta, sea registrada o almacenada en la entidad registro, teniendo pues en la entidad registro, un historial de todos los cambios por los que ha pasado una incidencia.
- Una incidencia puede tener varias valoraciones, es decir se podrán valorar varias facetas de la incidencia como la velocidad o la eficacia con la que ha sido resuelta.



## 5. Diseño de la solución

### 5.1 Arquitectura del sistema

El actual proyecto como se ha mencionado anteriormente cuenta con tres subproyectos. A continuación, se mostrarán que arquitecturas se han empleado en cada uno de ellos.

#### 5.1.1 Backend y servicio de Web Api

Para la implementación de este proyecto se ha seguido el diseño guiado por el dominio DDD (*Domain Driven Desing*)<sup>13</sup> el cual será explicado de manera más detallada a continuación:

Primero de todo, hay que preguntarse qué es el dominio, el dominio en si es el área temática o campo a la que un usuario aplica un software. También hay que tener claro el concepto de modelo de dominio, que en si es la representación de la terminología y los conceptos claves del problema.

Una vez tenemos estos dos conceptos claros, el DDD se centra en tener un modelo rico, expresivo y en constante evolución para resolver problemas de una forma semántica.

Este diseño o patrón de desarrollo está compuesto por varias capas conceptuales.

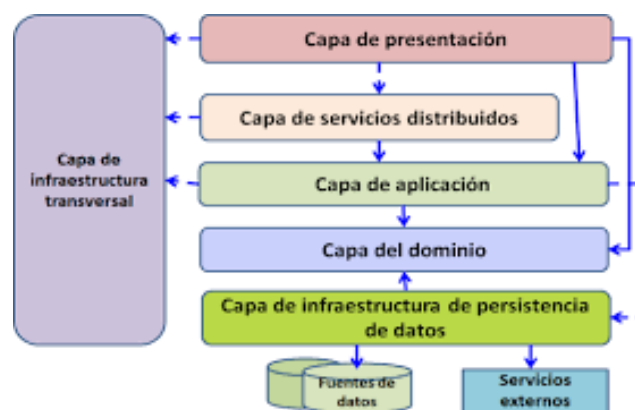


Ilustración 10: Domain Driving Desing<sup>14</sup>

<sup>13</sup> [https://github.com/jatubio/5minutos\\_laravel/wiki/Resumen-sobre-DDD.-Domain-Driven-Design](https://github.com/jatubio/5minutos_laravel/wiki/Resumen-sobre-DDD.-Domain-Driven-Design)

<sup>14</sup> <http://xurxodeveloper.blogspot.com/2014/01/ddd-la-logica-de-dominio-es-el-corazon.html>



Como se puede apreciar en la ilustración 3, este patrón está compuesto por 6 capas, pero en nuestro proyecto vamos a hacer uso de 4 capas solo, puesto que cuando se sigue un patrón de diseño, no es necesario ceñirse totalmente a lo que recomiende el patrón. Hemos obviado la capa de infraestructura transversal puesto que esta capa está destinada a funcionalidades accedidas por cualquier capa, como podría ser la autenticación, que en nuestro caso es manejada por el web api. También se ha decidido prescindir de la capa de servicios distribuidos ya que la capa de servicios distribuidos y de presentación prácticamente se solapan, ya que en la capa de servicios distribuidos tendremos el web api con la que el usuario interactuara a través de peticiones HTTP. A continuación, se van a mencionar las capas empleadas en nuestro proyecto y una breve descripción de sus características<sup>15</sup>.

### **Interfaz de usuario (*User Interface*)**

Esta capa muestra la información al usuario, interpreta sus acciones y las envía a la aplicación, ya bien sea mediante una interfaz gráfica o un consola de comandos.

### **Aplicación (*Application*)**

Es la que se encarga de organizar todos los elementos de la aplicación. No contiene lógica de negocio ni mantiene el estado de los objetos de negocio. Es responsable de mantener el estado de la aplicación y del flujo de esta.

### **Dominio (*Domain*)**

Es el núcleo de nuestro proyecto. Contiene toda la información sobre nuestro dominio, a su vez contiene todas las reglas de negocio y es responsable de mantener el estado de los objetos de negocio.

### **Infraestructura (*Infrastructure*)**

Esta capa sirve para la persistencia de los objetos, a su vez también sirve para proveer la comunicación entre el resto de las capas. Es la capa más cercana a la base de datos y donde de normal se implementa el código del ORM para el acceso a los datos de la base de datos.

---

<sup>15</sup> <https://devexperto.com/domain-driven-design-1/>



Dado que son capas conceptuales, su implementación puede ser muy variada y en una misma aplicación, tendremos partes o componentes que formen parte de cada una de estas capas.

### 5.1.2 Aplicación Web y Aplicación móvil

Para la implementación de la aplicación web se ha utilizado Angular y para la implementación de la aplicación móvil Ionic, que es un framework basado en Angular

también, así pues, este es el diseño o arquitectura seguido en una aplicación de Angular.

Las aplicaciones web basadas en Angular están fragmentadas en múltiples partes, a estas partes se las conocen como componentes. Estos componentes suelen estar comunicados por módulos, que de una manera contextual engloban componentes que tengan funcionalidades parecidas o con el mismo fin. Por último, tenemos los servicios o proveedores, que son los que se encargan de suministrar la información a los componentes de la aplicación.

A continuación, veremos mas en detalle como esta compuesto un componente, y como funciona un servicio en angular.

Un componente está dividido en tres partes:

- La primera parte contiene la estructura del componente (HTML).
- La segunda parte contiene la lógica del componente (TypeScript).
- La tercera parte contiene los estilos del componente para dotarlo de una mejor visualización (CSS).

Una vez tenemos claro como está compuesto un componente, hay que abordar el problema de obtener la información a mostrar, esto es mediante servicios.

Un servicio, no es mas que un objeto que mantiene el estado y que es igual para todos los componentes de la aplicación. Estos objetos son “inyectados” en los componentes y son proveídos en cualquier módulo de la aplicación. Lo explicado anteriormente resume el paradigma de la inyección de dependencias<sup>16</sup>. En palabras más técnicas, la inyección de dependencias es un patrón de diseño orientado a objetos, en el que se suministran

---

<sup>16</sup> [https://es.wikipedia.org/wiki/Inyecci%C3%B3n\\_de\\_dependencias](https://es.wikipedia.org/wiki/Inyecci%C3%B3n_de_dependencias)

objetos a una clase en lugar de ser la propia clase la que cree dichos objetos. Esos objetos cumplen contratos que necesitan nuestras clases para poder funcionar (de ahí el concepto de dependencia).

## 5.2 Diseño detallado

En este apartado vamos a mencionar de una manera más detallada el diseño y organización de las clases que interactúan en cada uno de los tres subproyectos de la solución.

### 5.2.1 Backend y servicio de Web Api.

El servicio Web Api y el Backend están formados por cuatro proyectos los cuales todos pertenecen a la misma solución, cada proyecto implementa cada una de las capas vistas anteriormente del DDD, a continuación, se mostrarán las carpetas que contienen cada uno de los proyectos y que uso o finalidad tienen.

#### Proyecto *Domain*

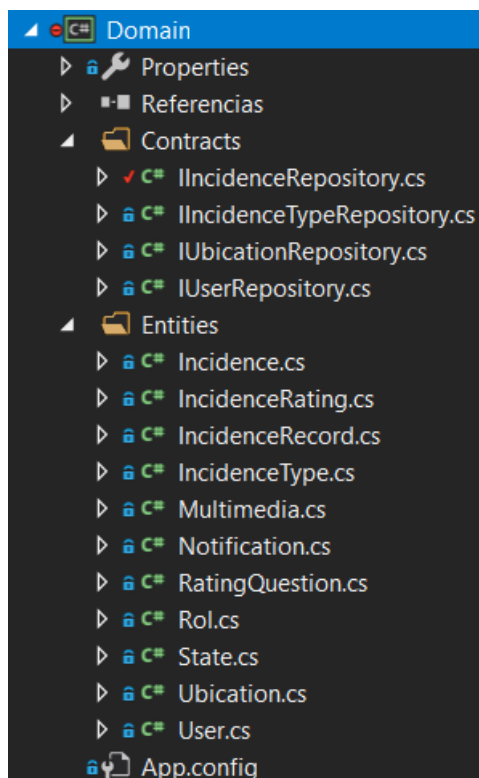


Ilustración 11: Proyecto Domain

Como se puede apreciar en la ilustración 4 este proyecto consta principalmente de dos carpetas *Contracts* y *Entites*.

En entities es donde tenemos las clases que definen el dominio o problema de nuestro proyecto de una manera sencilla y abstracta. Por ejemplo, la clase *User* solo contiene los atributos que debe de tener un usuario. Dicho de otro modo, esta carpeta contiene la definición del problema a resolver en nuestro proyecto.

En Contracts tenemos todas las interfaces que deben implementar los repositorios (clases del proyecto *Infraestructure*), para poder efectuar acciones sobre las entidades del dominio.

El resto de los proyectos han de ser capaces de obtener información de este, pero no al contrario, ya que este proyecto no debe depender de ningún otro, dado que es el núcleo de nuestra aplicación.

### Proyecto *Infrastructure*

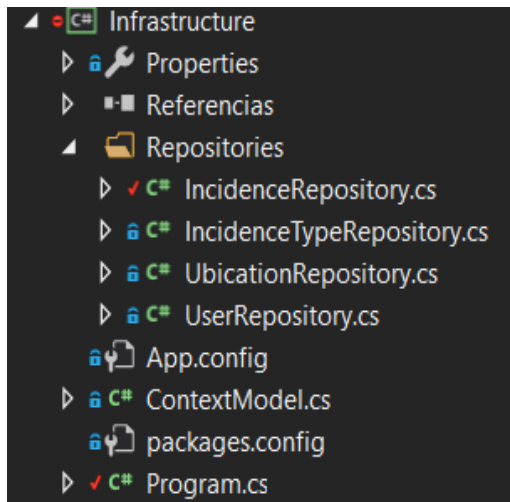


Ilustración 12: Proyecto Infrastructure

En la ilustración 5 vemos el proyecto dedicado a la capa de Infraestructura. Caben destacar dos partes, la carpeta de *Repositories* y la clase *ContextModel*.

La carpeta *Repositories* contiene todas las clases encargadas de leer, escribir y eliminar información en la base de datos, éstas a su vez deben implementar las interfaces definidas por los contratos de las entidades mencionadas en el apartado anterior.

Por otro lado, tenemos la clase *ContextModel* esta clase es la encargada de conectar a la base de datos, en ella reside toda la configuración y la cadena para efectuar la conexión. Hay que destacar que, todas las clases de los repositorios han de hacer uso de esta clase para poder efectuar todas las acciones descritas en lo pertinentes contratos.

### Proyecto *Services*

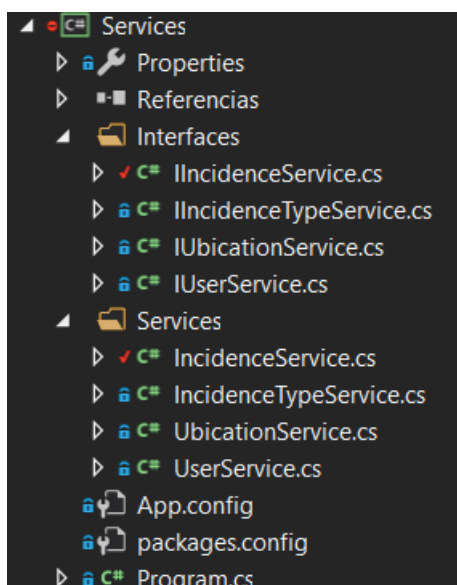


Ilustración 13: Proyecto Services

En la ilustración 6 vemos el proyecto que implementa la capa de *Application* de nuestro modelo. Está formado por los servicios y sus interfaces. Los servicios hacen uso de los repositorios de la capa de infraestructura, por lo tanto, es necesaria la comunicación entre la capa de aplicación y la de infraestructura. Estos servicios son los que dotarán de funcionalidad a la capa de *user-interface* en nuestro caso un web api.

Otro punto para destacar de este proyecto es que será el encargado de mantener el estado de la aplicación, controlando así todas las excepciones

y errores que puedan ocurrir en el funcionamiento de la aplicación.

## Proyecto Web Api

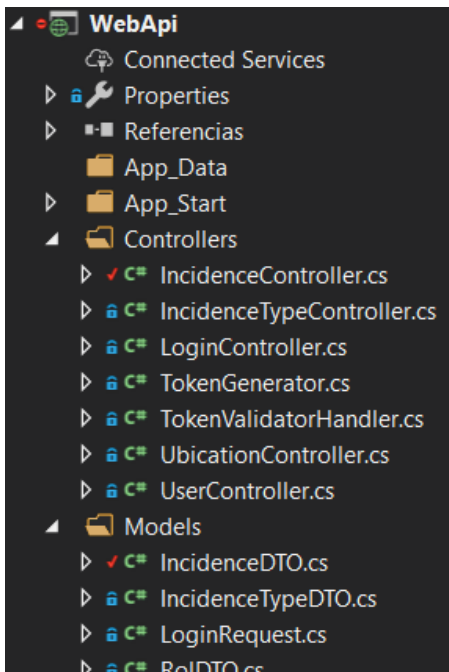


Ilustración 14: Proyecto Web Api

Este proyecto representa la capa de *user-interface*, en nuestro caso es un web api que dotará de los servicios y funcionalidades necesarios a los usuarios. Existen dos partes de este proyecto a destacar, la carpeta *Controllers* y la carpeta *Models*.

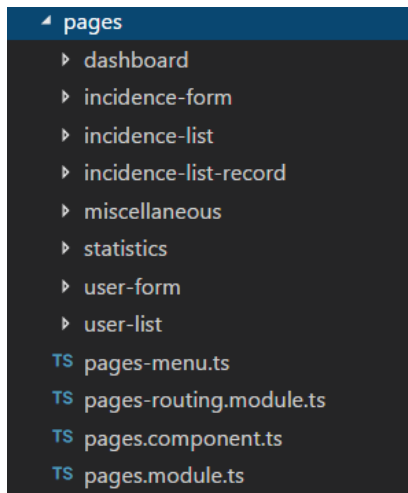
Como se puede apreciar en la ilustración 7 la carpeta *Controllers* contiene las clases públicas las cuales pueden ser accedidas mediante peticiones HTTP, estas clases harán uso de los servicios suministrados por la capa *Application* para poder realizar las acciones pedidas por los usuarios.

En cuanto a la carpeta *Models*, es la carpeta que contiene la forma o el formato de los datos esperados por los usuarios, entonces siempre que se pida un objeto de una entidad este objeto será transformado a su correspondiente modelo.

### 5.2.2 Aplicación Web y Aplicación móvil

Las aplicaciones Angular suelen tener la misma organización de directorios, en concreto vamos a hablar de tres de ellas, la carpeta *pages*, la carpeta *shared* y la carpeta *core*.

## Carpeta *Pages*



En la ilustración 8 podemos ver la carpeta *Pages* en ella están todos los componentes relacionados con las distintas páginas de nuestra aplicación web, serán estos componentes los que formen la interfaz gráfica y por los cuales navegaremos.

## Carpeta *Shared*

Ilustración 15: Carpeta Pages

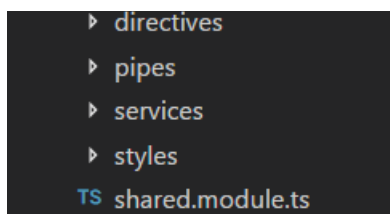


Ilustración 16: Carpeta Shared

En esta carpeta están todas las clases que van a ser usadas de manera global en nuestra aplicación web, en especial la carpeta *services* la cual contiene todos los servicios que suministrarán la información a los componentes de la carpeta *pages*.

## Carpeta *Core*

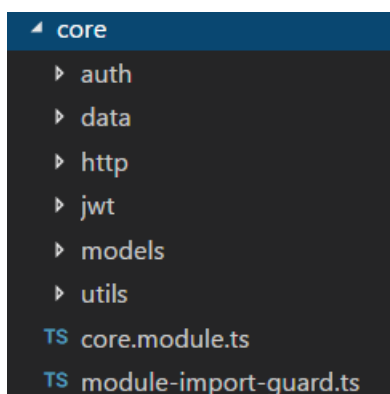


Ilustración 17: Carpeta Core

En esta carpeta residen los servicios utilizados por los propios servicios de la carpeta *shared*, tales como servicios de peticiones *http* o de autenticación con *jwt*, además de los servicios también están los modelos, que son el formato de la información que se espera obtener de las peticiones al servicio web api, estas clases están incluidas en la carpeta *models*.

## 5.3 Tecnologías utilizadas

Para cada subproyecto se han utilizado distintas tecnologías para su implementación y desarrollo.

Para la aplicación web se ha utilizado Angular dado que es uno de los *frameworks* mas utilizados a la hora del desarrollo de aplicaciones web, cuenta con una gran

comunidad y con una amplia información y documentación, también cabe destacar las constantes actualizaciones y el esfuerzo del equipo (Google) por mantener el framework mantenido y estable. El IDE para el desarrollo de la aplicación web ha sido Visual Studio Code ya que tiene compatibilidad tanto con HTML, con TypeScript como con CSS, aparte de esto es un IDE ligero y no requiere de configuración alguna para utilizarlo.

Para la aplicación móvil, se ha decidido utilizar Ionic como framework para el desarrollo, ya que permite desarrollar aplicaciones móviles mediante lenguajes de desarrollo web. Puesto que yo ya tenía conocimientos en desarrollo web, este punto me hizo decantarme por Ionic, aparte de la amplia documentación que ofrece y de la comunidad que tiene. El IDE para el desarrollo de esta aplicación también ha sido el Visual Studio Code dado que se desarrolla como una aplicación web basada en Angular.

El desarrollo del Backend y del servicio web API ha sido implementado en .NET. En específico para el desarrollo del Backend se ha utilizado Entity Framework 6<sup>17</sup> y para el desarrollo del Web Api se ha utilizado ASP.net Web Api 2<sup>18</sup>. Ambas tecnologías están basadas en C#.

La elección de Entity Framework como tecnología para el desarrollo del Backend, fue por la previa experiencia obtenida en la carrera ya que trabajamos con esta tecnología en ISW (Ingeniería del Software) y otro punto a favor de entity framework es la automatización de código.

Lo mismo ocurre con ASP.NET., es una tecnología para aplicaciones web desarrollada por Microsoft. Se usa principalmente para desarrollar sitios web, aplicaciones y servicios web. Apareció en enero de 2002 con su primer versión 1.0. La elección ha sido por haber trabajado anteriormente con ella en un proyecto de una empresa en la que estuve formándome, también por su sencillez y que forma parte de .NET lo que hace menos complejo la comunicación entre el Backend y el servicio Web API.

Para la realización tanto del Backend como del servicio Web Api se ha utilizado como IDE Visual Studio ya que ambas tecnologías pertenecen a .NET y la herramienta ofrece distintas facilidades para la instalación e implementación de ambas tecnologías.

---

<sup>17</sup> <https://docs.microsoft.com/es-es/dotnet/framework/data/adonet/ef/overview>

<sup>18</sup> <https://docs.microsoft.com/es-es/aspnet/#pivot=aspnet>



## 6. Desarrollo de la solución

---

### 6.1 Servicio Web Api

Para el desarrollo del servicio web api se ha utilizado ASP.NET, y quería destacar la sencillez con la que se puede crear un servicio web api. Para ello pondremos el ejemplo del objeto o recurso “Incidencia”.

Primero de todo cada recurso del web api tiene su propio controlador, es decir una clase que ofrece distintos métodos que serán los encargados de ejecutar las acciones sobre el recurso, por ejemplo, obtener la información de un incidencia en particular por su id o una lista completa de todas las incidencias.

```
[EnableCors(origins: "**", headers: "**", methods: "**")]
public class IncidenceController : ApiController
{
    private IncidenceService incidenceService;

    public IncidenceController()
    {
        this.incidenceService = new IncidenceService();
    }

    [Authorize]
    public IHttpActionResult Get()
    {
        List <IncidenceDTO> incidences =
            this.incidenceService.GetAllIncidences()
                .Select(incidence => new IncidenceDTO(incidence)).ToList();
        return Ok(incidences);
    }
}
```

En el código de arriba se puede observar la implementación del recurso *Incidence* y de la acción obtener todas las incidencias mediante el método *Get()*. Primero de todo se crea en el constructor un objeto *IncidenceService*, este objeto comunica con la capa de la aplicación del backend y es el que nos suministrará los datos. Una vez hecho esto



se definen los métodos del *api* para el recurso en cuestión. Si el nombre del método empieza por alguna acción HTTP, ASP.NET automáticamente asignará a ese método la acción por la que es nombrada, en nuestro caso el nombre del método es *Get*, con lo cual para acceder este recurso habrá que hacer una acción *Get* en:

Http://localhost:54906/Incidence

Esto automáticamente devolverá todas las incidencias.

Hay que añadir que el nombre del recurso en nuestro caso es *Incidence* porque hemos nombrado a la clase *IncidenceController*, si hubiéramos nombrado la clase como *UserController* el nombre del recurso habría sido *User*.

## 6.2 Backend

En el apartado del backend también hay que destacar la facilidad para el acceso a los datos mediante *Entity framework*, una vez instalado e implementado correctamente, basta con escribir el siguiente fragmento de código:

```
public List<Incidence> GetAllIncidences()
{
    using (ContextModel context = new ContextModel())
    {
        context.Configuration.LazyLoadingEnabled = false;
        return context.Incidences
            .Include("State")
            .Include("User")
            .Include("Ubication")
            .Include("IncidenceType")
            .ToList();
    }
}
```

Para acceder a los datos solo es necesario obtener un Objeto de la clase *ContextModel*, esta clase ha sido la encargada de leer la base de datos y de mapear toda esta información en objetos, así pues, la lista *Incidences* que hay en el objeto *context* tendrá todas las incidencias de la base de datos. Por último, los *include* son para referenciar los objetos anidados dentro del objeto solicitado, por ejemplo, un objeto de tipo incidencia a su vez tiene un objeto de tipo usuario como atributo.



## 6.3 Aplicaciones basadas en Angular (Aplicación móvil y web)

En el apartado de las aplicaciones basadas en angular me gustaría hacer hincapié en como se obtiene los datos de un servicio web api. Mediante servicios y un tipo de objetos denominados Observables conseguiremos la funcionalidad deseada. Los siguientes fragmentos de código resumen como se hace una petición http a un servicio web desde una aplicación basada en Angular.

```
@Injectable()
export class ContextModelService {

    private incidences;
    private user;
    private types;
    private ubications;

    private dataSource = new Subject();
    private $contextModel Observable<any>;

    constructor(private incidenceService: IncidenceService,
                private incidenceTypeService: IncidenceTypeService,
                private ubicacionService: UbicationService) {
        this.$contextModel = this.dataSource.asObservable();
    }

    public loadContextModel() {
        let incideces$ = this.incidenceService.getIncidences();
        let types$ = this.incidenceTypeService.getIncidencesTypes();
        let ubications$ = this.ubicacionService.getUbications();

        Observable.combineLatest(incideces$, types$, ubications$).subscribe(([incidences, types,
        ubications]) => {
            this.incidences = incidences;
            this.types = types;
            this.ubications = ubications;
            this.dataSource.next();
        });
        return this.$contextModel;
    }
}
```

La clase *ContextModelService* es un *Injectable*, que quiere decir esto, que puede ser inyectada como dependencia en cualquier otra clase, siguiendo el patrón de inyección de dependencias descrito anteriormente en el apartado de la arquitectura de la solución.

Esta clase suministra los datos al resto de páginas de la aplicación, dado que el método *loadContextModel* se ejecutará una vez el usuario se autentique en la aplicación.

El funcionamiento de este método es sencillo, hace uso de un observable público (*\$contextModel*), un observable es un objeto que publica información y al cual otros objetos o clases pueden subscribirse para obtener dicha información. La clase *ContextModelService* hace uso de otros tres servicios, el de incidencias, el de tipos y el de ubicaciones y trata de obtener la información necesaria de estos. Una vez tengo la información de los tres servicios, publico en el observable que la información ya es accesible.

La pregunta siguiente sería ¿Cómo los tres servicios inyectados en éste han obtenido los datos?

Veamos pues le servicio de Incidencias.

```
@Injectable()
export class IncidenceService {

    private endPoint = 'Incidence/';
    private actualIncidence;

    constructor(private httpService: HttpService) { }

    public getIncidences(): Observable<any> {
        return this.httpService.get(this.endPoint );
    }
}

@Injectable()
export class HttpService {
    constructor(private http: HttpClient) { }

    public get(path: string): Observable<any> {
        return this.http.get(`${environment.api_url}${path}`);
    }
}
```



La clase `IncidenceService` es una clase que al igual que la de `ContextModel` es inyectable, y su única función es la de hacer uso del servicio `HttpService` y hacer uso del método `get` pasándole el recurso solicitado en este caso `Incidence`. Vayamos un punto mas lejos y veamos que hace el servicio `HttpService`.

```
@Injectable()
export class HttpService {
  constructor(private http: HttpClient) {}

  public get(resource: string): Observable<any> {
    return this.http.get(`${environment.api_url}${resource}`);
  }
}
```

Esta clase al igual que las otras dos, también es inyectable y a su vez también se inyecta un servicio en esta caso `HttpClient`, perteneciente a las librerías de Angular y del cual utilizamos su método `get` para hacer una petición GET mediante HTTP. De esta manera podremos hacer una petición a nuestro servicio de Web Api con el recurso solicitado.

# 7. Resultados

Para la demostración de los resultados se va a proceder a mostrar distintas imágenes, de la aplicación web y la aplicación móvil en funcionamiento.

## 7.1 Aplicación Web

Como se ha mencionado en anteriores apartados la aplicación web sirve como portal de administración de las distintas incidencias reportadas, aparte de la lectura y actualización de estas incidencias, también es posible ver distintas estadísticas en cuanto a cantidad de incidencias producidas.

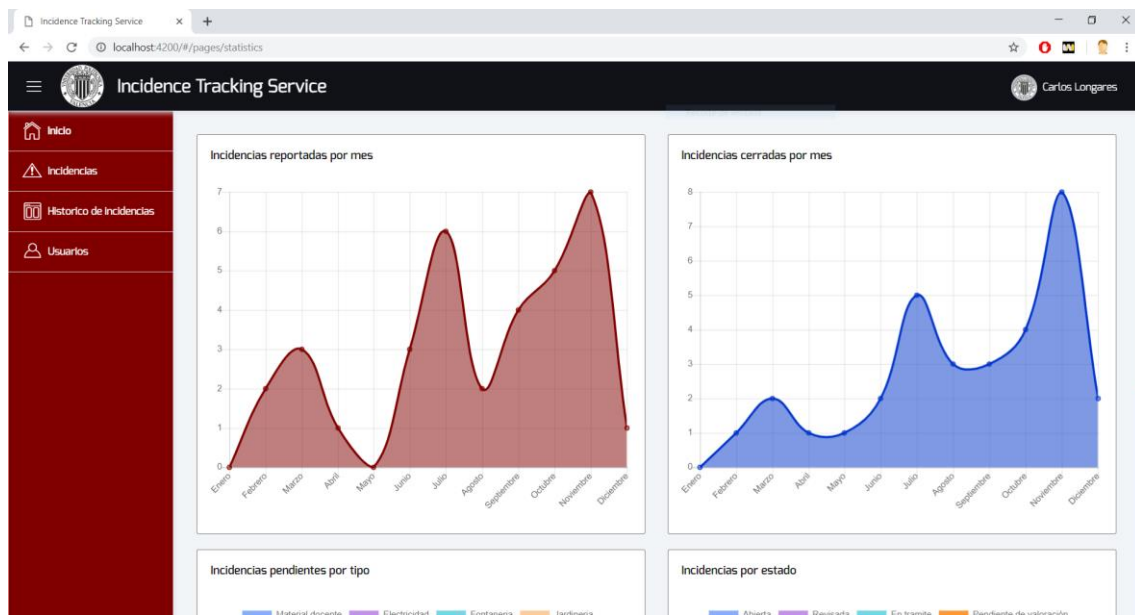
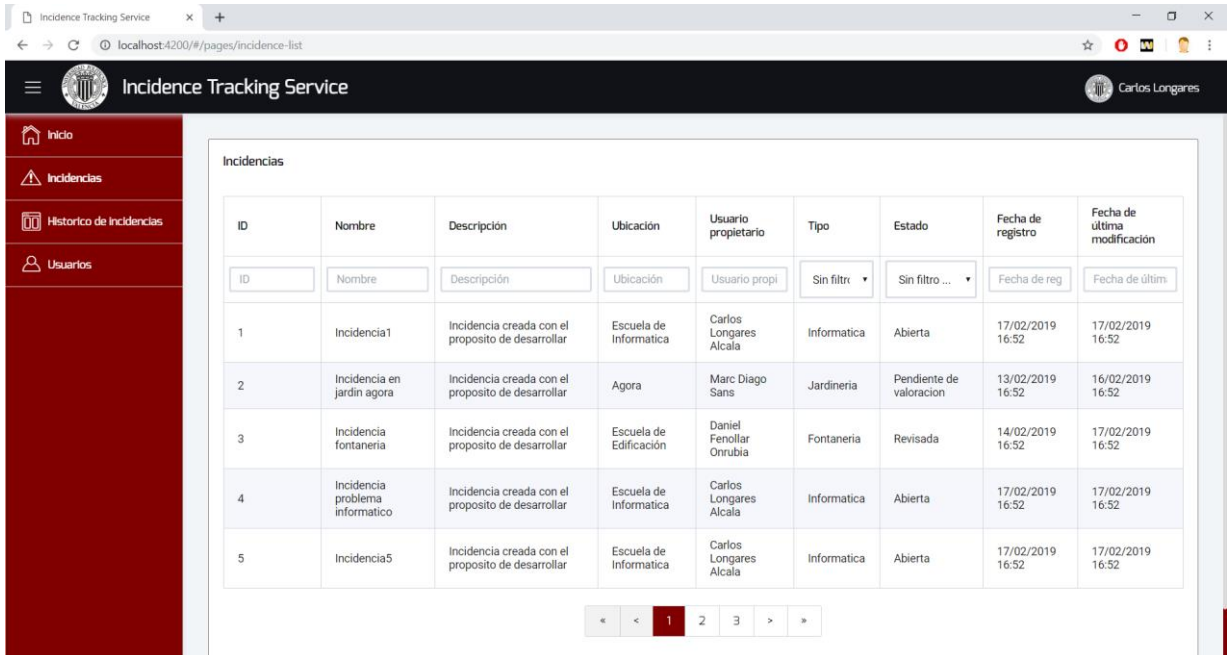


Ilustración 18: Pantalla de estadísticas

Como se muestra en la ilustración 11, se aprecia lo que es la pantalla principal de la aplicación web. En ésta se muestran distintas gráficas sobre estadísticas de las incidencias registradas en el sistema. En esta pantalla hay cuatro tipos de gráficas, dos de ellas son de tipo área y nos muestran las incidencias reportadas y cerradas por mes. Las otras dos gráficas restantes son de tipo tarta y nos muestra cuantas incidencias hay reportadas, filtrándolas por tipo y por el estado en el que se encuentran.

## Aplicación para la notificación y seguimiento de incidencias – Incident Tracker Service (ITS)

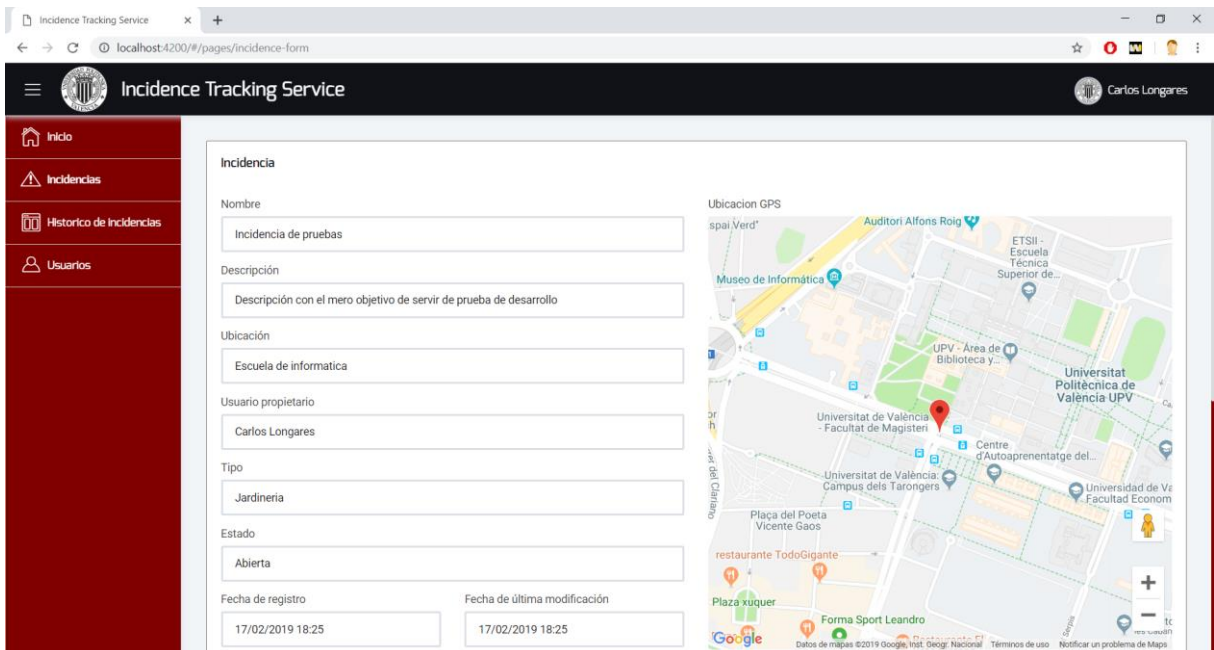


The screenshot shows the Incident Tracker Service (ITS) application interface. On the left is a navigation menu with options: Inicio, Incidencias, Historico de Incidencias, and Usuarios. The main content area displays a table titled 'Incidencias' with the following columns: ID, Nombre, Descripción, Ubicación, Usuario propietario, Tipo, Estado, Fecha de registro, and Fecha de última modificación. Below the table is a pagination control showing page 1 of 3.

ID	Nombre	Descripción	Ubicación	Usuario propietario	Tipo	Estado	Fecha de registro	Fecha de última modificación
1	Incidencia1	Incidencia creada con el proposito de desarrollar	Escuela de Informatica	Carlos Longares Alcalá	Informatica	Abierta	17/02/2019 16:52	17/02/2019 16:52
2	Incidencia en jardin agora	Incidencia creada con el proposito de desarrollar	Agora	Marc Diago Sans	Jardineria	Pendiente de valoracion	13/02/2019 16:52	16/02/2019 16:52
3	Incidencia fontaneria	Incidencia creada con el proposito de desarrollar	Escuela de Edificación	Daniel Fenollar Orrubia	Fontaneria	Revisada	14/02/2019 16:52	17/02/2019 16:52
4	Incidencia problema informatico	Incidencia creada con el proposito de desarrollar	Escuela de Informatica	Carlos Longares Alcalá	Informatica	Abierta	17/02/2019 16:52	17/02/2019 16:52
5	Incidencia5	Incidencia creada con el proposito de desarrollar	Escuela de Informatica	Carlos Longares Alcalá	Informatica	Abierta	17/02/2019 16:52	17/02/2019 16:52

Ilustración 19: Pantalla con el listado de incidencias

En la ilustración 12 tenemos la pantalla con el listado de incidencias, esta pantalla esta compuesta por una tabla en la que cada fila es una incidencia, cada columna representa un atributo de la incidencia, los cuales son: el identificador, el nombre, la descripción, la ubicación, el usuario propietario, el tipo, el estado, la fecha de registro y la fecha de última modificación. También añadir, que es posible filtrar los resultados que muestra la tabla por cualquiera de sus columnas.



The screenshot shows the Incident Tracker Service (ITS) application interface for viewing the details of an incident. The left navigation menu is the same as in the previous screenshot. The main content area is titled 'Incidencia' and contains the following fields:

- Nombre: Incidencia de pruebas
- Descripción: Descripción con el mero objetivo de servir de prueba de desarrollo
- Ubicación: Escuela de informatica
- Usuario propietario: Carlos Longares
- Tipo: Jardineria
- Estado: Abierta
- Fecha de registro: 17/02/2019 18:25
- Fecha de última modificación: 17/02/2019 18:25

On the right side of the form, there is a map titled 'Ubicacion GPS' showing the location of the incident. The map includes labels for 'Auditori Alfons Roig', 'ETSII - Escuela Técnica Superior de...', 'UPV - Área de Biblioteca y...', 'Universitat Politècnica de València UPV', 'Universitat de València - Facultat de Magisteri', 'Universitat de València - Campus dels Tarongers', 'Plaza del Poeta Vicente Gaos', 'restaurant TodoGigante', 'Plaza xuquer', and 'Forma Sport Leandro'.

Ilustración 20: Pantalla con la información de una incidencia

En la ilustración 13 aparece la pantalla con la información específica de una incidencia. Para llegar a esta pantalla es necesario a ver seleccionado una incidencia del listado de incidencias de la pantalla mencionada anteriormente (Ilustración 12). En esta pantalla se muestran los siguientes campos: nombre, descripción, ubicación, usuario propietario, tipo, estado, fecha de registro, fecha de última modificación, posición GPS en Google Maps y una galería de imágenes de la incidencia, si la incidencia tiene imágenes adjuntas. En esta pantalla es posible modificar cualquier dato sobre la incidencia, ya que para acceder a la aplicación web deberemos ser Administradores.

## 7.2 Aplicación móvil

La aplicación móvil tiene como función la de reportar las incidencias, esta aplicación cuenta con distintas pantallas, entre las cuales se encuentran la de login, listado de incidencias, nueva incidencia y nuevo usuario.

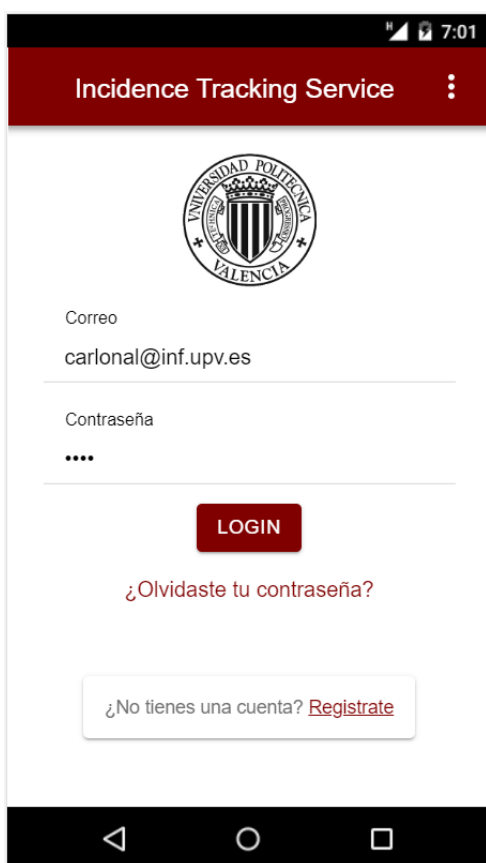


Ilustración 21: Login aplicación móvil



Ilustración 22: Listado de incidencias móvil

Como se puede apreciar en la ilustración 14, tenemos la pantalla de login de la aplicación móvil, su diseño como el resto de las pantallas, sigue las guías de estilo de *Material Design* de Google. Esta pantalla nos ofrece la posibilidad de autenticarnos en la aplicación, de recuperar nuestra contraseña en caso de no recordarla y si no tenemos una cuenta, poder acceder al formulario de nueva cuenta de usuario.

En la siguiente ilustración (ilustración 15) se muestra la pantalla con el listado de incidencias reportadas por el usuario, esta es la pantalla principal de la aplicación, la cual será mostrada nada mas autenticarse en la aplicación. Cada elemento de la lista muestra: el nombre de la incidencia, el tipo, la ubicación, la descripción y el estado en el que se encuentra. Además de la lista tenemos el botón '+' situado en la esquina inferior derecha de la pantalla, el cual nos permitirá registrar nuevas incidencias.

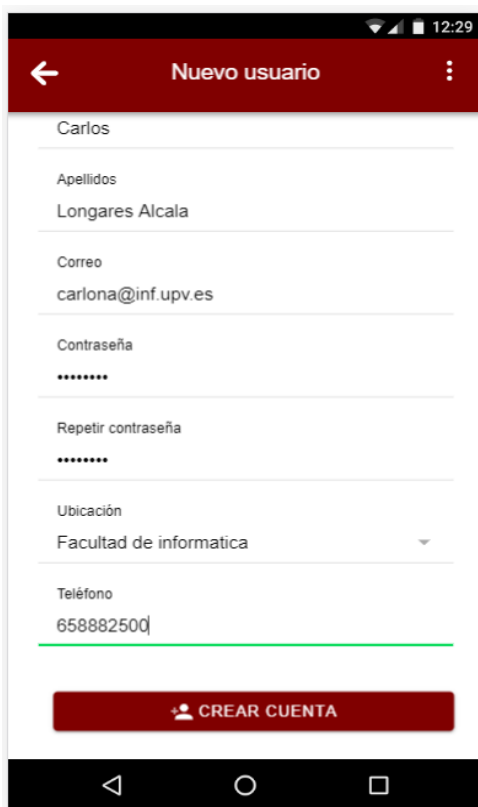


Ilustración 16: Creación de usuario

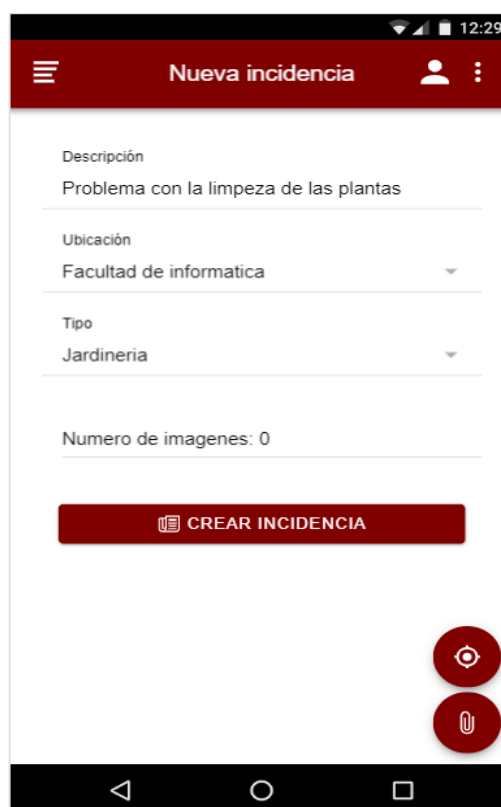


Ilustración 17: Creación de incidencia

La ilustración 16 nos muestra el formulario de creación de usuario, el cual es accesible desde la pantalla de login (ilustración 14). Todos los campos de este formulario son obligatorios, los cuales son: nombre, apellidos, correo, contraseña, repetir contraseña, ubicación y teléfono.

Como ultima pantalla a mostrar (ilustración 17) tenemos el formulario de la creación de incidencias, el cual está compuesto por los siguientes campos: descripción, ubicación,



tipo, posición GPS desde donde se ha reportado la incidencia y archivos multimedia adjuntos. Estos dos últimos campos se rellenan o adquieren mediante los botones situados en la parte inferior derecha.

## 8. Conclusiones

---

Como conclusión, el desarrollo de servicios web y aplicaciones es un método muy útil dada la alta utilización de los dispositivos móviles y las aplicaciones web actualmente.

En este contexto, los servicios web nos proporcionan bastante flexibilidad a la hora de enfrentarnos a cualquier problema de tipo informático. Por ejemplo, que una aplicación sea multiplataforma o que se vea de una forma adecuada según el dispositivo donde se muestre.

Otra conclusión para destacar es la cantidad de lenguajes y tecnologías necesarias para crear una aplicación. En mi caso han sido necesarias 9, desde las tecnologías y lenguajes enfocadas en el backend y el servicio web api (SQL, C#, entity framework y ASP.NET web api 2) hasta las utilizadas en las aplicaciones web y móvil (typescript, HTML, CSS, Angular y Ionic). Esto hace replantearse la cantidad de conocimiento necesario para el desarrollo de una aplicación o herramienta, y del tiempo invertido en esta.

Por lo demás, se han cumplido los siguientes objetivos descritos anteriormente en el apartado de la introducción:

- Ofrecer una herramienta para el reporte de incidencias, ágil e intuitiva a través de una aplicación móvil.
- Ayudar en tareas de logística, dado que será posible reportar faltas de material en las aulas.
- Agilizar la comunicación de incidencias, es decir conseguir que la información de la incidencia llegue lo antes posible al personal de administración o mantenimiento.
- Tener más detalles de cada incidencia mediante fotografías y la ubicación GPS de donde se produjo esta, aportando así mayor calidad de información.
- Llevar un registro de todas las incidencias reportadas, tanto si han sido solucionadas o no.
- Ofrecer distintas estadísticas de incidencias, para saber si se están distribuyendo bien los recursos empleados en la resolución de estas.

Objetivos pendientes para futuros trabajos:

- Permitir la valoración de una incidencia solucionada por parte de los usuarios.

- Trasladar el proyecto de un contexto de desarrollo a un contexto de producción.

Con este proyecto he aprendido a valorar el tiempo que conlleva el desarrollo de una herramienta software, la importancia de la planificación y de la cantidad de errores y contratiempos que pueden surgir a la hora de implementar una aplicación.

Hay que añadir que no se ha podido desarrollar todo lo anteriormente citado en el apartado de análisis, por una mala planificación del tiempo y por una mala gestión las tareas a desarrollar, dado que tareas que pensaba que me llevarían menos tiempo han ocupado mucho mas de lo planeado.

## **8.1 Relación del trabajo desarrollado con los estudios cursados**

Hay varias asignaturas las cuales han sido muy prácticas para la realización de este proyecto, entre ellas cabe destacar: Ingeniería del software y Gestión de proyectos. Sinceramente para un proyecto de este tipo, habría sido más interesante cursar la especificación de Ingeniería del software, pues en el apartado de especificación de requisitos no se ha llegado a la complejidad necesaria para un proyecto de esta envergadura.

Las competencias transversales requeridas para la realización de este proyecto han sido:

- Diseño y proyecto.
- Planificación y gestión del tiempo.
- Aprendizaje permanente.
- Aplicación y pensamiento practico.
- Innovación, creatividad y emprendimiento.

## 9. Referencias

---

- [1] *Las mejores herramientas de ticketing* [en línea] [fecha de consulta: 15 marzo 2019]. Disponible en: <https://integriaims.com/las-mejores-herramientas-ticketing/>
- [2] *Software CRM para service desk y gestión de incidencias* [en línea] [fecha de consulta: 16 marzo 2019]. Disponible en: <https://www.dynamics-crm.es/crm-por-areas-de-gestion/service-desk>
- [3] *Ilustración de la aplicación web de Service desk* [en línea] [fecha de consulta: 15 marzo 2019]. Disponible en: <https://community.dynamics.com/crm/b/microsoftdynamicscrmappstore/archive/2014/06/13/service-desk-solution-for-dynamics-crm-2013>
- [4] *Zendesk, Software de atención al cliente y sistema de tickets de soporte* [en línea] [fecha de consulta: 15 marzo 2019]. Disponible en: <https://www.zendesk.es>
- [5] *Zendesk, Ilustración de la aplicación web de Zendesk* [en línea] [fecha de consulta: 15 marzo 2019]. Disponible en: <https://www.getapp.com/customer-service-support-software/a/zendesk/#gallery-1>
- [6] *JIRA servive desk – Valiantys Atlassian Platinum Partner* [en línea] [fecha de consulta: 15 marzo 2019]. Disponible en: <https://valiantys.com/en/products/jira-service-desk/>
- [7] *Ilustración de la aplicación de escritorio de OTRS* [en línea] [fecha de consulta: 15 marzo 2019]. Disponible en: [https://es.m.wikipedia.org/wiki/Archivo:OTRS\\_5\\_Wikimedia\\_\(en\).png](https://es.m.wikipedia.org/wiki/Archivo:OTRS_5_Wikimedia_(en).png)
- [8] *Software Libre CRM + Marketing Automático para Pymes ACTIVALINK* [en línea] [fecha de consulta: 15 marzo 2019]. Disponible en: <https://www.activalink.com/>
- [9] *Ilustración de la aplicación de ACTIVALINK* [en línea] [fecha de consulta: 15 marzo 2019]. Disponible en: <https://www.activalink.com/>

- [10] *Help Desk Software basado en ticketing* – *ServiceTonic* [en línea] [fecha de consulta: 16 marzo 2019]. Disponible en: <https://www.servicetonic.es/help-desk-software/>
- [11] *FreshService ITSM System ITIL-aligned service desk software* [en línea] [fecha de consulta: 15 marzo 2019]. Disponible en: <https://freshservice.com>
- [12] *Flow Marker online and Diagram Software* [en línea] [fecha de consulta: 16 marzo 2019]. Disponible en: <https://www.draw.io/>
- [13] *Resumen sobre DDD. Domain Driven Desing* [en línea] [fecha de consulta: 10 marzo 2019]. Disponible en: [https://github.com/jatubio/5minutos\\_laravel/wiki/Resumen-sobre-DDD.-Domain-Driven-Design](https://github.com/jatubio/5minutos_laravel/wiki/Resumen-sobre-DDD.-Domain-Driven-Design)
- [14] *Introducción a Domain Driven Desing (DDD): Parte 1* [en línea] [fecha de consulta: 10 marzo 2019]. Disponible en: <https://devexperto.com/domain-driven-design-1/>
- [15] *Inyección de dependencias* [en línea] [fecha de consulta: 5 marzo 2019]. Disponible en: [https://es.wikipedia.org/wiki/Inyecci%C3%B3n\\_de\\_dependencias](https://es.wikipedia.org/wiki/Inyecci%C3%B3n_de_dependencias)
- [16] *Información general sobre Entity Framework – Microsoft Docs* [en línea] [fecha de consulta: 5 marzo 2019]. Disponible en: <https://docs.microsoft.com/es-es/dotnet/framework/data/adonet/ef/overview>
- [17] *Documentación de ASP.NET – Microsoft Docs* [en línea] [fecha de consulta: 6 marzo 2019]. Disponible en: <https://docs.microsoft.com/es-es/aspnet/#pivot=aspnet>